

Gh. Sabău

V. Avram

A. Cojocaru

\*

A. Sotir

V. Orbeanu

C. Baronide

R. Nedelcu

\* \*

N. Fildan

A. Alexandrescu

A. Dobrițoiu

D. V. Norea

\* \*

Coordonatori : Gh. Sabău (coord. gen.) și A. Sotir

## Practica bazelor de date

Totul despre... SOCRATE și SOCRATE-MINI  
pe Felix C, CORAL, Independent

2

- Manuale:
- lansare,
- utilizare,
- operare

- Exemple
- Aplicații complexe.

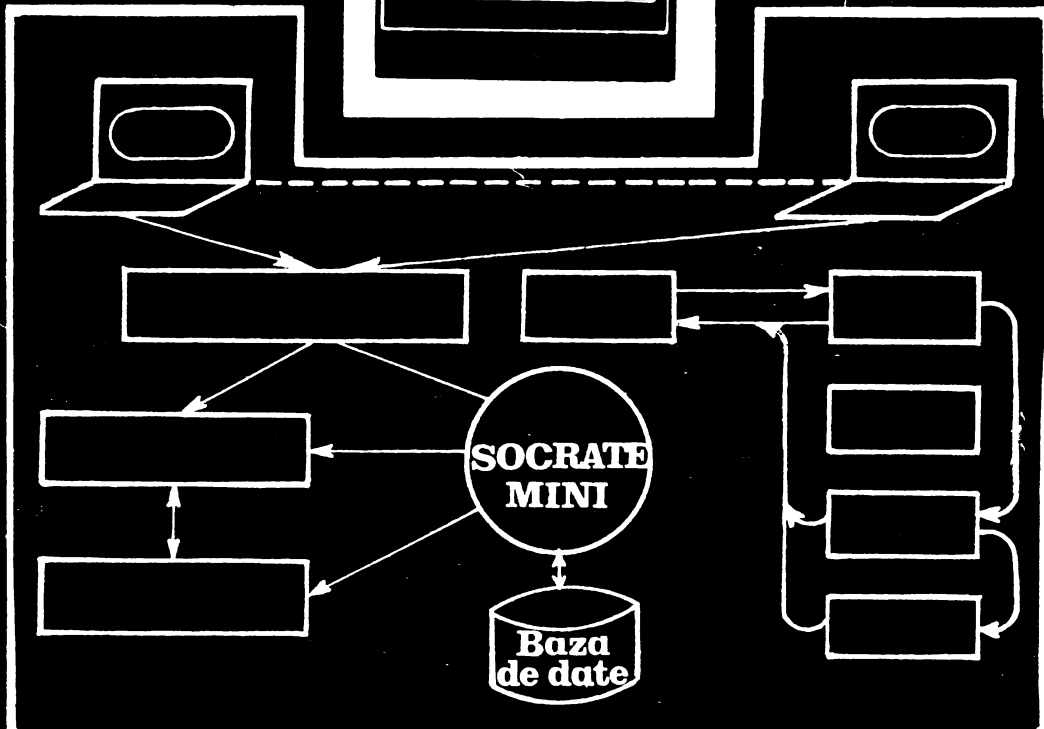
AUTOMATICA

ELECTRONICA

SERIA PRACTICA

INFORMATICA

MANAGEMENT





AUTOMATICA  
E  
L  
E  
C  
T  
R  
O  
N  
I  
C  
A  
  
M  
A  
N  
A  
G  
E  
M  
E  
N  
T

# BIBLIOTECA DE AUTOMATICĂ INFORMATICĂ-ELECTRONICĂ- MANAGEMENT

- S. Radu, D. Filotti : Centrale telefonice automate  
M. Bodea ș.a. : Tranzistoare cu efect de cimp  
D. N. Shapiro : Proiectarea radioreceptoarelor  
V. Antonescu, M. Popovici : Ghid pentru controlul statistic al calității  
V. Baltac ș.a. : Calculatorul FELIX C-256. Structură și programare  
G. Sonea, Sileșchi M. : Creșterea planificată a productivității muncii  
R. L. Moriss : Proiectarea cu circuite integrate TTL  
A. Brilliantov : Calculul și construcția televizoarelor portabile  
Kaoru Ishikawa : Controlul de calitate pentru maștri și șefi de echipe  
Magnus Radke : 222 măsuri pentru reducerea costurilor  
I. Stăncioiu : Eficiența economică a asimilării de utilaje noi  
G. Lajtha : Proiectarea rețelelor de telecomunicații  
Vădășescu, A. ș.a. : Dispozitive semiconductoare. Manual practic  
Ch. Jones : Design. Metode și aplicații  
E. S. Buffa : Conducerea modernă a producției, vol. I și II  
D. W. Davies, ș.a. : Rețele de interconectarea calculatoarelor  
Gh. Baștirea ș.a. : Comanda numerică a mașinilor-unelte  
L. W. Crum : Analiza valorii  
P. Fotaș : Automatica și informatica în procesele editoriale-poliigrafice  
P. Vezeanu, Șt. Pătrașcu : Măsurarea temperaturii în tehnică  
T. Penescu, V. Petrescu : Măsurarea presiunii în tehnică  
P. Popescu, P. Mîhordea : Măsurarea debitului de tehnică  
P. Vezeanu : Măsurarea nivelului în tehnică  
A. Nadoto : Măsurarea volumului și cantității lichidelor în industrie  
C. Hidoș, P. Isac (coordonatori) : Studiul muncii, I-VIII  
C. Hidoș : Analiza și proiectarea circuitelor informaționale  
Gh. I. Pisău : Elaborarea și implementarea sistemelor informatice  
P. Constantinescu, ș.a. : Sisteme informatice, modele ale conducerii  
V. Penescu, ș.a. : Fișiere, baze și bănci de date  
I. Ceaușu ș.a. : S.D.V. Organizarea concepției, fabricației, gestiunii  
S. Brebenel : Practica transferului internațional de tehnologie  
P. Constantinescu ș.a. : Analiză, decizie, control  
A. Vădășescu ș.a. : Circuite integrate liniare. Manual de utilizare, vol. 1, 2, 3  
S. Măican : Sisteme numerice cu circuite integrate  
I. Rîstea ș.a. : Manualul muncitorului electronist  
M. Florescu ș.a. : Cibernetică, informatică, automată în industria chimică  
E. Statnic, M. Gănescu : Televizoare cu circuite integrate  
T. Geber ș.a. : Echipamente periferice  
S. Călin ș.a. : Optimizări în automatizări industriale  
M. Simonescu : Proiectarea unitară a circuitelor electronice  
C. Cruceru : Tehnica măsurărilor în telecomunicații  
P. Nițulescu : Electroalimentarea instalațiilor de telecomunicații  
R. Răpeanu ș.a. : Circuite integrate analogice. Catalog  
T. Rădulescu ș.a. : Centrale telefonice automate  
N. Drăgulănescu : Agenda radioelectronistului  
V. Baltac, A. Davidoviciu, C. Mașec ș.a. : Sisteme interactive și limbaje conversaționale  
Th. Boranțu, R. Dobrescu : Structuri moderne de conducere automată a mașinilor-unelte  
A. Petrescu ș.a. : Microcalculatoarele FELIX M18, M18B, M118, vol. I și II  
D. Stanomir : Sisteme electroacustice  
N. Iosiș ș.a. : Tiristoare și module de putere. Catalog  
S. Călin, I. Dumitrache ș.a. : Reglarea numerică a proceselor tehnologice  
G. Ionescu ș.a. : Traductoare pentru automatizări industriale  
D. Boboc ș.a. : Cartea operatorului de la tablourile de automatizare  
A. Mîlea : Cartea metrologului  
M. Voicu : Tehnica de analiză a stabilității sistemelor automate  
A. Stănescu ș.a. : Sisteme de automatizare pneumatice  
H. M. Moșt, A. Ciocrița-Vasilescu : Debitmetrie industrială  
N. Terțîșco, P. Stoica, Th. Popescu : Identificarea asistată de calculator a sistemelor  
T. Baron ș.a. : Calitate și fiabilitate. Manual practic, vol. 1+vol. 2



Lector univ. dr. ec. **Gheorghe Sabău**  
Ec. **Vasile Avram**  
Ec. **Aurelian Cojocaru**

Ing. **Alexandru Sotir**  
Ing. **Valeriu Orbeanu**  
Ing. **Constanțin Baronide**  
Ing. **Răzvan Nedelcu**

Ing. **Nicolae Fildan**  
Ing. **Adrian Alexandrescu**  
Mat. **Andrian Dobrițoiu**  
Ing. **Dan Viorel Norea**

\*

\*\*

\*\*

# Practica bazelor de date

Totul despre... SOCRATE și SOCRATE-MINI  
pe Felix C, CORAL, INDEPENDENT

Coordonatori : **Gh. Sabău** (coord. gen.) și **Al. Sotir**

## Vol. 2

Manuale de :

- lansare
- utilizare
- operare

- Exemple
- Aplicații complexe



**EDITURA TEHNICA**  
București, 1989

Recenzii : Dr. mat. **MARGARETA DRĂGHICI**  
Dr. ing. **STELIAN GUȚU**

Cuvînt înainte : Dr. mat. **MARGARETA DRĂGHICI**

Redactor : ing. **PAUL ZAMFIRESCU**

Tehnoredactor : **MARIA TRĂZNEA**  
Desene : Arh. **MIOARA HORTOPAN**  
**ADRIAN GUGA**

---

B.T. 27,02,1989. C.T. 17

C.Z. : 621.377

---

**ISBN 973-31-0020-X**

**ISBN 973-31-0022-6**

---

I. P. Oltenia – Craiova  
str. Mihai Viteazul, nr. 4  
Comanda nr. 151/1988



## CUPRINS GENERAL

<b>Volumul 1</b>		3. Instalarea produsului . . . . .	16
Cuvînt înainte . . . . .	5	4. Noutățile versiunii . . . . .	17
Prezentări . . . . .	7	5. Portabilitatea versiunii . . . . .	21
TESTE (1—8) . . . . .	15	6. SOCRATE-MINI în „cifre” . . . . .	22
<b>Secțiunea I. Baze de date și SGBD-ul SOCRATE</b>		7. Anomaliile și restricțiile ridicate în raport cu V 2.1 . . . . .	23
1. Baze de date — definiție și caracteristici . . . . .	17	8. Restricțiile versiunii . . . . .	24
2. SGBD SOCRATE — Prezentare sintetică . . . . .	32	<b>Secțiunea VI. Manual de utilizare SOCRATE-MINI</b>	
<b>Secțiunea II. SGBD SOCRATE — Tratare generală</b>		1. Introducere . . . . .	25
3. Limbajul de descriere a datelor . . . . .	66	2. Limbajul de descriere a structurii . . . . .	27
4. Limbajul de manipulare a datelor . . . . .	115	3. Limbajul de manipulare . . . . .	50
5. Alte componente ale SGBD-SOCRATE . . . . .	193	4. Programul bibliotecar . . . . .	83
6. Utilizarea funcțiilor SGBD-SOCRATE . . . . .	238	5. Interfețe de comunicare cu alte limbaje. Exemple de programe SOCRATE, COBOL, PASCAL, FORTRAN, BASIC . . . . .	92
7. Integritatea bazelor de date . . . . .	274	6. Interfața pentru lucrul într-o rețea de baze de date . . . . .	105
<b>Secțiunea III. Optimizări și dezvoltări de SGBD-uri</b>		7. Interfața SOCRATE-MINI—SOCRATE-Felix . . . . .	123
8. Optimizarea structurilor bazelor de date și a programelor de exploatare . . . . .	286	<b>Secțiunea VII. Manual de operare SOCRATE-MINI</b>	
9. Dezvoltări SOCRATE și alte SGBD-uri . . . . .	348	1. Introducere . . . . .	148
<b>Secțiunea IV. Aplicație complexă cu SGBD SOCRATE</b>		2. DFS — Procesorul de definiție a structurilor . . . . .	151
10. Proiectarea bazelor de date pentru activitatea de personal . . . . .	375	3. SOC — Procesorul limbajului de manipulare . . . . .	158
11. Implementarea bazelor de date PERSONAL (listing complet generarea bazei, întreținerea bazei de date, generarea automată a programelor... testarea coerenței fizice și logice, simularea rularii) . . . . .	395	4. MGS — Programul bibliotecar . . . . .	163
ANEXE . . . . .	455	5. Programe utilitare . . . . .	167
TESTE (9—25) . . . . .	482	6. Interfețe de comunicare cu alte limbaje . . . . .	194
Bibliografie . . . . .	495	7. Interfața pentru regăsirea datelor dintr-o rețea de baze de date . . . . .	195
<b>Volumul 2</b>		8. Interfața SOCRATE MINI—SOCRATE-Felix . . . . .	197
<b>Secțiunea V. Manual de lansare SOCRATE-MINI</b>		<b>Secțiunea VIII. Aplicație complexă cu SGBD SOCRATE-MINI pentru activitatea de exploatare a flotei maritime</b>	
1. Introducere . . . . .	11	1. Analiza activității . . . . .	200
2. Descrierea benzilor de livrare . . . . .	14	2. Proiectarea schemei conceptuale a bazei de date . . . . .	204
		3. Elaborarea programelor și a procedurilor . . . . .	209
		4. Rezolvarea, pe baza de date implementată, a activității de programare a transportului de mărfuri (Studiu de caz) . . . . .	209
		5. Listingul complet al schemei conceptuale, al manipulării datelor, al rezolvării studiului de caz . . . . .	216
		ANEXA. Noutățile versiunii V.4.0 a produsului SOCRATE-MINI . . . . .	256

## CUPRINS DETALIAȚ

### Volumul 2

<p><b>Secțiunea V. Manual de lansare SOCRATE-MINI</b> . . . . . 11</p> <p><b>1. INTRODUCERE</b> . . . . . 11</p> <p style="padding-left: 20px;">Arhitectura generală . . . . . 11</p> <p style="padding-left: 20px;">Structura datelor . . . . . 11</p> <p style="padding-left: 20px;">Schema de memorare . . . . . 12</p> <p style="padding-left: 20px;">Programe de aplicație . . . . . 12</p> <p style="padding-left: 20px;">Funcțiunile SGBD SOCRATE-MINI</p> <p style="padding-left: 40px;">Generarea schemei de memorare a bazei de date . . . . . 12</p> <p style="padding-left: 40px;">Generarea structurii datelor pentru baza de date . . . . . 12</p> <p style="padding-left: 40px;">Accesul la baza de date . . . . . 13</p> <p style="padding-left: 40px;">Utilizatorii bazei de date . . . . . 13</p> <p style="padding-left: 40px;">Întreținerea bazei de date . . . . . 14</p> <p><b>2. DESCRIEREA BENZII DE LIRARE</b> . . . . . 14</p> <p style="padding-left: 20px;">Componentele produsului . . . . . 14</p> <p style="padding-left: 20px;">Programele de test . . . . . 15</p> <p style="padding-left: 20px;">Documentația . . . . . 16</p> <p><b>3. INSTALAREA PRODUSULUI</b> . . . . . 16</p> <p><b>4. NOUȚĂȚILE VERSIUNII V.30</b> . . . . . 17</p> <p style="padding-left: 20px;">Creșterea performanțelor . . . . . 17</p> <p style="padding-left: 20px;">Interfața pentru regăsirea datelor dintr-o rețea de baze de date . . . . . 17</p> <p style="padding-left: 20px;">Interfața SOCRATE-MINI — SOCRATE-Felix . . . . . 18</p> <p style="padding-left: 20px;">Calculul expresiilor aritmetice cu rotunjire . . . . . 18</p> <p style="padding-left: 20px;">Specificarea dinamică a numărului de zecimale la variabilele <math>W_i</math> . . . . . 19</p> <p style="padding-left: 20px;">Extinderea gamei de tipuri de comparație posibile . . . . . 19</p> <p style="padding-left: 20px;">Protejarea implicită prin BLOQUER-LIBERER a cererilor G, S, SE . . . . . 19</p> <p style="padding-left: 20px;">Unificarea tratării erorilor . . . . . 20</p> <p style="padding-left: 20px;">O facilitate suplimentară la preluarea datelor cu EXT . . . . . 20</p> <p style="padding-left: 20px;">Posibilitatea preluării valorii ERREUR într-o variabilă <math>Y_i</math> . . . . . 20</p> <p style="padding-left: 20px;">Creșterea numărului de variabile <math>X_i</math> și <math>W_i</math> . . . . . 20</p>	<p>Posibilitatea catalogării programelor cu fișier de securitate . . . . . 21</p> <p><b>5. PORTABILITATEA VERSIUNII</b> . . . . . 21</p> <p style="padding-left: 20px;">Portabilitatea la nivelul limbajului sursă . . . . . 21</p> <p style="padding-left: 20px;">Portabilitatea la nivelul bazei de date . . . . . 21</p> <p style="padding-left: 20px;">Portabilitatea la nivelul apelului componentelor . . . . . 22</p> <p><b>6. SOCRATE-MINI PREZENTAT ÎN CIFRE</b> . . . . . 22</p> <p><b>7. ANOMALII ȘI RESTRICȚII RIDICATE ÎN RAPORT CU V.2.1</b> . . . . . 23</p> <p style="padding-left: 20px;">Restricții ridicate . . . . . 23</p> <p style="padding-left: 20px;">Anomalii înlăturate . . . . . 23</p> <p><b>8. RESTRICȚIILE VERSIUNII V.30</b> . . . . . 24</p> <p><b>Secțiunea VI. Manual de utilizare SOCRATE MINI</b> . . . . . 25</p> <p><b>1. INTRODUCERE</b> . . . . . 25</p> <p style="padding-left: 20px;">Componentele SOCRATE-MINI . . . . . 25</p> <p style="padding-left: 20px;">Notații și convenții utilizate în manual . . . . . 27</p> <p><b>2. LIMBAJUL DE DESCRIERE A STRUCTURII</b> . . . . . 27</p> <p style="padding-left: 20px;">Generalități . . . . . 27</p> <p style="padding-left: 40px;">Apelul procesorului de descriere . . . . . 27</p> <p style="padding-left: 40px;">Elemente lexicale : setul de caractere, separatorii limbajului de descriere . . . . . 28</p> <p style="padding-left: 40px;">Elemente sintactice : cuvinte cheie, identificatori, etichete virtuale, constante . . . . . 28</p> <p style="padding-left: 40px;">Noțiunea de caracteristică, clasificări . . . . . 29</p> <p style="padding-left: 40px;">Noțiunea de domeniu . . . . . 31</p> <p style="padding-left: 40px;">Noțiunea de formal . . . . . 32</p> <p style="padding-left: 40px;">Forma generală a unui program de descriere . . . . . 32</p>
--	---

Caracteristici ale bazei de date . . .	33	Cerere D . . . . .	64
Caracteristica de tip MOT . . .	33	Cerere DETACHE . . . . .	64
Caracteristica de tip valoare		Cerere ECRIRE . . . . .	65
numerică bornată . . . . .	34	Cerere EXEC . . . . .	66
Caracteristica de tip listă de		Cerere FAIRE . . . . .	66
valori . . . . .	36	Cerere G . . . . .	67
Caracteristica de tip FILLER . . .	36	Cerere I . . . . .	68
Caracteristica de tip entitate . . .	37	Cerere LIBERER . . . . .	71
Caracteristica de tip bloc . . . . .	38	Cerere LIRE . . . . .	71
Caracteristica de tip inel . . . . .	39	Cerere M . . . . .	71
Caracteristica de tip referință		Cerere PAUSE . . . . .	80
cu inel . . . . .	40	Cerere POUR . . . . .	80
Caracteristica de tip referință		Cerere REFAIRE . . . . .	81
simplă . . . . .	41	Cerere S . . . . .	81
Caracteristica de tip INVERSE	41	Cerere SE . . . . .	82
Organizarea datelor . . . . .	42	Cerere SI . . . . .	82
Tipul de organizare . . . . .	42	Cerere SORTIE . . . . .	83
Domenii și etichete virtuale . . . .	44	Cerere SUIVANT . . . . .	83
Metode de acces la informații . . . . .	47	4. PROGRAMUL BIBLIOTECAR . . . . .	83
Acces secvențial . . . . .	47	Generalități . . . . .	83
Acces direct prin număr de ordi-		Apelul programului bibliotecar	84
ne . . . . .	47	Introducerea macroinstrucțiunilor	
Acces direct prin dicționar . . . . .	47	in baza de date . . . . .	84
Formale . . . . .	48	Noțiunea de macroinstrucțiune	
Caracteristica de tip FORMAL . . . .	48	SOCRATE . . . . .	84
Caracteristica de tip MOT		Definirea și inserarea unei ma-	
FORMAL . . . . .	49	croinstrucțiuni . . . . .	85
Valori numerice în FORMAL . . . . .	49	Apelul unei macroinstrucțiuni	86
3. LIMBAJUL DE MANIPULARE . . . . .	50	Observații importante . . . . .	87
Generalități . . . . .	50	Exemple de macroinstrucțiuni	87
Apelul procesorului limbajului		Introducerea programelor pre-	
de manipulare . . . . .	51	compilate în baza de date . . . . .	90
Setul de caractere. Separatorii		Noțiunea de program precom-	
limbajului de manipulare . . . . .	51	pilat . . . . .	90
Cuvinte cheie . . . . .	52	Definirea și inserarea progra-	
Identificatori . . . . .	52	melor precompilate . . . . .	90
Constante . . . . .	52	Observații importante . . . . .	90
Variabile de lucru . . . . .	52	Ștergerea macroinstrucțiunilor . . .	91
Structura unui program scris		Ștergerea programelor precom-	
în limbajul de manipulare . . . . .	53	pilate . . . . .	91
Macrogeneratorul . . . . .	53	Listarea macrodefinițiilor . . . . .	92
Apelul de macroinstrucțiune . . . .	54	Listarea programelor precompilate .	92
Concepte SOCRATE . . . . .	54	Listarea completă . . . . .	92
Tipuri de date . . . . .	54	5. INTERFEȚE DE COMUNICARE	
Citare-complexă. Citare-formal . . .	55	CU ALTE LIMBAJE (cu exemple	
Noțiunea de condiție . . . . .	57	de programe în SOCRATE,	
Condiție simplă . . . . .	58	COBOL, FORTRAN, BASIC,	
Condiție folosită într-un fil-		PASCAL) . . . . .	92
tru rapid . . . . .	59	Necesitate., obiective . . . . .	92
Criteriul de sortare . . . . .	59	Interfața limbaje evaluate — SO-	
Condiția folosită într-un fil-		CRATE-MINI . . . . .	93
tru secvențial (lent) . . . . .	60	Primitivele de legătură cu baza	
Condiția de existență . . . . .	60	de date . . . . .	94
Logica de evaluare a condi-		Deschiderea bazei de date . . . . .	94
țiilor . . . . .	61	Accesul la baza de date . . . . .	95
Cereri SOCRATE . . . . .	61	Închiderea bazei de date . . . . .	96
Cerere APPEL . . . . .	62	Tratarea erorilor . . . . .	96
Cerere ATTACHE . . . . .	62	Primitiva de legătură cu inter-	
Cerere BLOQUER . . . . .	63	preterul liniilor de comandă . . . . .	97
Cerere CKPT . . . . .	63	Exemple de programe care fo-	
		losoesc interfața cu SOCRATE . . . . .	97

Exemplu de program apelant scris în COBOL . . . .	98	ANEXE . . . . .	129
Exemplu de program apelant scris în FORTRAN . . .	99	Anexa A. Mesaje de eroare . . . .	129
Exemplu de program apelant scris în PASCAL . . . .	100	Erori detectate la catalogarea structurilor . . . . .	129
Exemplu de program apelant scris în BASIC . . . .	101	Erori detectate în analiza sintactică . . . . .	129
Interfața SOCRATE-MINI — Limbaje evaluate . . . . .	103	Erori detectate în analiza semantică . . . . .	129
Exemplu de subprogram COBOL . . . . .	103	Erori fatale . . . . .	129
Exemplu de subprogram FORTRAN . . . . .	104	Coduri de eroare . . . . .	130
Programul apelant SOCRATE . . . . .	104	Erori detectate de procesorul limbajului de manipulare . . . . .	131
Interfața SOCRATE-MINI — Interpretarea liniilor de comandă . . . . .	104	Erori detectate în analiza lexicală . . . . .	131
6. INTERFAȚA PENTRU LUCRUL ÎNTR-O REȚEA DE BAZE DE DATE (cu exemple de programe în SOCRATE, COBOL, FORTRAN)	105	Erori detectate în analiza sintactică . . . . .	132
Necesitate, obiective . . . . .	105	Erori detectate în analiza semantică . . . . .	132
Interfața apelată din limbajul de manipulare . . . . .	106	Erori recuperabile detectate în execuție. Variabila ERREUR . . . . .	132
Primitivele de legătură . . . . .	108	Erori fatale detectate în execuție . . . . .	132
Conectarea unei baze de date . . . . .	108	Erori de operare . . . . .	133
Accesul la o bază de date conectată . . . . .	108	Coduri de eroare . . . . .	133
Deconectarea unei baze de date . . . . .	109	Erori detectate de programul bibliotecar . . . . .	137
Tratarea erorilor . . . . .	109	Erori detectate în analiza lexicală . . . . .	137
Exemplu de program scris în limbajul de manipulare . . . . .	110	Erori detectate în execuție . . . . .	137
Interfața apelată din limbaje evaluate . . . . .	114	Anexa B. Cuvinte cheie . . . . .	138
Primitivele de legătură . . . . .	114	Cuvinte cheie ale limbajului de descriere a structurilor . . . . .	138
Conectarea unei baze de date . . . . .	114	Cuvinte cheie ale limbajului de manipulare . . . . .	139
Accesul dintr-un program scris într-un limbaj evoluat la baza de date conectată . . . . .	115	Cuvinte cheie ale programului bibliotecar . . . . .	139
Deconectarea bazei de date . . . . .	117	Anexa C. Sintaxa limbajului de manipulare . . . . .	139
Tratarea erorilor . . . . .	117	Anexa D. Diferențe față de SOCRATE V1.5 Felix C-256 . . . . .	145
Exemple de programe scrise în limbaje evaluate . . . . .	117	Anexa E. Exemplu de structură . . . . .	146
Exemplu de program scris în COBOL . . . . .	117	Secțiunea VII. Manual de operare SOCRATE-MINI . . . . .	148
Exemplu de program scris în FORTRAN . . . . .	121	1. INTRODUCERE . . . . .	148
7. INTERFAȚA SOCRATE-MINI — SOCRATE FELIX . . . . .	123	Componentele SOCRATE-MINI . . . . .	148
Necesitate, obiective . . . . .	123	Domenii. Schema de memorare . . . . .	149
Primitivele de legătură . . . . .	123	Sesiune SOCRATE-MINI . . . . .	149
Primitiva de conectare . . . . .	125	Convenții de apel generale . . . . .	150
Primitiva de acces . . . . .	125	2. DFS — PROCESORUL DE DEFINIRE A STRUCTURII . . . . .	151
Primitiva de deconectare . . . . .	126	Introducere în DFS . . . . .	151
Tratarea erorilor . . . . .	126	Comutatori . . . . .	152
Exemplu de program . . . . .	126	Apel . . . . .	152
		Linia de comandă . . . . .	152
		Convenții de apel . . . . .	152
		Catalogarea inițială a unei structuri . . . . .	153

Adăugarea de caracteristici la structură . . . . .	153	Mesaaje de eroare . . . . .	171
Alinierea caracteristicilor la subpagină. Comutatorul /SA . . . . .	154	LGI — Program de deschidere a unei sesiuni SOCRATE-MINI . . . . .	174
Geografia datelor. Comutatorul /MAP . . . . .	154	Introducere în LGI . . . . .	174
Mesaaje de eroare apărute la terminal . . . . .	156	Comutatori LGI . . . . .	175
3. SOC — PROCESORUL LIMBAGIULUI DE MANIPULARE . . . . .	158	Linia de comandă inițială . . . . .	175
Introducere în SOC . . . . .	158	Activarea . . . . .	175
Comutatori . . . . .	158	Activarea totală . . . . .	175
Apel . . . . .	159	Activarea parțială . . . . .	175
Linia de comandă . . . . .	159	Activarea cu fișier jurnal rapid . . . . .	176
Convenții de apel . . . . .	159	Restricții de activare . . . . .	176
Dimensionarea zonei de memorie necesară. Comutatorii /DSB și /DSR . . . . .	159	Deschiderea sesiunii SOCRATE-MINI . . . . .	176
Formatul fișierului de intrare. Comutatorul /CVF . . . . .	160	Mesaaje de eroare . . . . .	177
Opțiuni de listare. Comutatorii /SL, /ME și /SP . . . . .	160	LGO — Program de închidere a unei sesiuni SOCRATE-MINI . . . . .	179
Mesaaje de pe eroare apărute la terminal . . . . .	161	Introducere în LGO . . . . .	179
MGS — PROGRAMUL BIBLIOTECAR . . . . .	163	Comutatori LGO . . . . .	179
Introducere în MGS . . . . .	163	Linia de comandă inițială . . . . .	180
Comutatori . . . . .	164	Terminarea normală . . . . .	180
Apel . . . . .	164	Abortarea unei sesiuni SOCRATE-MINI . . . . .	180
Linia de comandă . . . . .	164	Mesaaje de eroare . . . . .	180
Convenții de apel . . . . .	164	ABS — Program de abortarea a procesoarelor SOCRATE-MINI . . . . .	182
Mesaaje de eroare apărute la terminal . . . . .	165	Introducere în ABS . . . . .	181
5. PROGRAME UTILITARE . . . . .	167	Linia de comandă . . . . .	182
GFM — Program de generare și formatare a bazei de date . . . . .	167	Mesaaje de eroare . . . . .	182
Introducere în GFM . . . . .	167	RSS — Program de restaurare a bazei de date după producerea unui incident . . . . .	183
Comutatori GFM . . . . .	167	Introducere în RSS . . . . .	183
Linia de comandă inițială . . . . .	168	Linia de comandă inițială . . . . .	183
Generarea . . . . .	168	Restaurarea într-un punct de reluare . . . . .	184
Generarea totală . . . . .	168	Mesaaje de eroare . . . . .	184
Linile de descriere a domeniilor . . . . .	168	STS — Program de editarea statisticii bazei de date și componenței ei . . . . .	186
Generarea parțială . . . . .	169	Introducere în STS . . . . .	186
Condiții restrictive la generare . . . . .	169	Comutatori STS . . . . .	186
Terminarea fazei de generare . . . . .	170	Linia de comandă inițială . . . . .	187
Comutatorii /-GN sau NOGN . . . . .	170	Editarea statisticii . . . . .	184
Formatarea . . . . .	170	Editarea statisticii întregii baze de date . . . . .	187
Formatarea generală . . . . .	170	Editarea statisticii parțiale a bazei de date . . . . .	187
Formatarea parțială . . . . .	170	Editarea componenței . . . . .	187
Terminarea fazelor de formatare . . . . .	170	Mesaaje de eroare . . . . .	188
Comutatorii /-FM sau NOFM . . . . .	171	SAR — Program de salvare-restaurare a bazei de date . . . . .	189
Drepturi de acces la bază. Comutatorii /RW și /RO . . . . .	171	Introducere în SAR . . . . .	189
		Comutatori SAR . . . . .	190
		Apel . . . . .	190
		Linia de comandă inițială . . . . .	190
		Linile de comandă pentru selectarea domeniilor . . . . .	190
		Schimbarea spațiului real . . . . .	191
		Schimbarea dimensiunii subpaginii . . . . .	191
		Compactarea domeniilor DIC și PRG . . . . .	192
		Mesaaje de eroare . . . . .	192

6. INTERFEȚE DE COMUNICARE CU ALTE LIMBAJE . . . . .	194	5. LISTING COMPLET AL SCHE- MEI CONCEPTUALE, AL MANI- PULĂRII DATELOR, AL REZOL- VĂRII STUDIULUI DE CAZ . . . . .	216
Interfața Limbaje evaluate — SO- CRATE-MINI . . . . .	194	Anexă. Noutățile versiunii V.4.0 SOCRATE-MINI . . . . .	256
Interfața SOCRATE-MINI — Lim- baje evaluate . . . . .	194	1. VERIFICAREA ȘI REFACEREA COERENȚEI FIZICE ȘI LOGICE A BAZEI DE DATE . . . . .	256
7. INTERFAȚA PENTRU LUCRUL INTR-O REȚEA DE BAZE DE DATE . . . . .	195	Extensii ale utilitarului STS . . . . .	256
Considerații generale . . . . .	195	Utilitarul VRU . . . . .	257
Interfața apelată din limbajele ev- luate . . . . .	196	Introducere în VRU . . . . .	257
Interfața apelată din limbajul de manipulare . . . . .	197	Funcțiuni de verificare a coerenței logice a bazei de date . . . . .	259
8. INTERFAȚA SOCRATE-MINI — SOCRATE FELIX . . . . .	197	Funcțiuni de refacere a coerenței logice a bazei de date . . . . .	260
Resurse necesare . . . . .	197	Funcțiuni de refacere a coerenței fizice a bazei de date . . . . .	262
Generarea procesorului de inter- față de la Felix . . . . .	198	Funcțiuni de ștergere . . . . .	262
Indicații de operare . . . . .	199	Afișarea informațiilor dintr-un do- meniu . . . . .	263
<i>Secțiunea VIII. Aplicație complexă cu SGBD SOCRATE-MINI pen- tru activitatea de exploatare a flotei maritime . . . . .</i>	200	2. GESTIUNEA FORMATELOR DE ECRAN . . . . .	263
1. ANALIZA ACTIVITĂȚII . . . . .	200	DSF — utilitar pentru defnirea formatelor de ecran . . . . .	263
2. PROIECTAREA SCHEMEI CON- CEPTUALE A BAZEI DE DATE . . . . .	204	DRW — program de interfață cu o bază de date prin intermediul for- matelor de ecran . . . . .	266
3. ELABORAREA PROGRAMELOR ȘI A PROCEDURILOR . . . . .	209	3. ALTE EXTENSII ALE PRODU- SULUI . . . . .	268
4. REZOLVAREA PE BAZA DE DATE IMPLEMENTATĂ, A AC- TIVITĂȚII DE PROGRAMARE A TRANSPORTULUI DE MĂRFURI (STUDIU DE CAZ) . . . . .	213	Noi facilități ale procesorului DSF . . . . .	268
		Noi funcțiuni ale procesorului SOC . . . . .	269
		Noi funcțiuni ale procesorului MGS . . . . .	269
		Sporirea confidențialității: drep- tuzi de compilare . . . . .	269
		Extinderea setului de caractere pentru procesoare . . . . .	270



## 1. INTRODUCERE

SOCRATE-MINI este un sistem de gestiune a bazelor de date pentru minicalculatoare românești din gama Independent, Coral sau pentru minicalculatoare compatibile cu PDP 11/34. El implementează modelul rețea SOCRATE, fiind compatibil cu versiunea 1.5 a produsului SOCRATE funcțională pe calculatoare FELIX C 256/512/1024. Față de versiunea V1.5 au fost prevăzute importante facilități suplimentare, unele dintre ele fiind inspirate de versiunile V 1.6 și V 1.7 ale produsului SOCRATE.

### Arhitectura generală

Arhitectura generală a sistemului SOCRATE-MINI este prezentată în fig. V.1.1

#### Structura datelor

La nivel conceptual trecerea de la realitate la schema conceptuală presupune un proces de modelare prin care obiectele lumii reale sînt grup în clase. SGBD. SOCRATE-MINI dispune de un Limbaj de Descriere a Datelor (LDD) care permite specificarea schemei conceptuale.

LDD este inevitabil legat de modelul de date implementat de SGBD SOCRATE. LDD permite definirea unor structuri arborescente și a unor structuri de tip rețea. Prin LDD se descriu și se denumesc cîmpurile elementare și, pe de altă parte, relațiile între acestea.

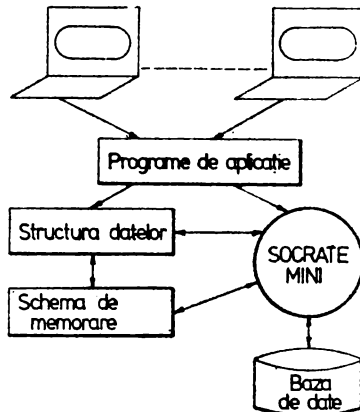


Fig. V.1.1

## Schema de memorare

Schema de memorare definește modul de împărțire în domenii de structură fizică a bazei de date prin intermediul Limbajului de Descriere a Schemei de Memorare.

## Programe de aplicație

Programele de aplicație sînt scrise în Limbajul de Manipulare a Datelor. LMD are toate atributele unui limbaj de programare și în plus permite accesul la datele aflate în baza de date. Cu un program conceput în LMD se pot efectua deci operații asupra datelor din bază : creare, actualizare, interogare. Anumite modificări în structura datelor pot fi efectuate fără a afecta programele de aplicație scrise deja. De asemenea, programele de aplicații sînt complet independente de împărțirea bazei în domenii.

## Funcțiunile SGBD SOCRATE-MINI

Produsul SOCRATE-MINI asigură următoarele funcțiuni :

- generarea Schemei de Memorare a Bazei de Date ;
- generarea structurii datelor ;
- accesul la baza de date ;
- consultarea bazei de date ;
- întreținerea bazei de date.

### Generarea schemei de memorare a bazei de date

Generarea schemei de memorare a bazei de date constă în memorarea informațiilor legate de împărțirea bazei de date în domenii de structură fizică a bazei de date pe disc magnetic. La nivel logic un domeniu reprezintă un spațiu virtual de date. La nivel fizic unui domeniu îi corespund unul sau mai multe fișiere în sensul sistemului de operare pe minicalculator.

Informațiile sînt furnizate utilizînd limbajul de descriere a schemei de memorare (LDSM). Ulterior schema de memorare poate fi completată cu elemente noi în legătură cu alte domenii ce se pot adăuga bazei de date. Descrierea cîmpurilor acestor domenii se face prin furnizarea atributelor corespunzătoare în cadrul structurii datelor.

### Generarea structurii datelor pentru baza de date

Structura datelor conține informații despre denumirea și natura cîmpurilor elementare și despre relațiile dintre ele. Aceste informații sînt furnizate cu ajutorul limbajului de descriere a datelor. Structura datelor este reținută în baza de date de către procesorul de structură și va fi consultată ulterior de celelalte componente ale produsului SOCRATE-MINI.

Structura datelor poate fi oricînd completată cu informații despre noi cîmpuri de date ce pot face parte din domenii descrise anterior sau din domenii noi adăugate la schema de memorare.

O componentă specializată a produsului SOCRATE-MINI, procesorul limbajului de descriere a structurii, are ca sarcină analiza unei structuri descrisă cu ajutorul LDD și memorarea în bază a informațiilor rezultate în urma acestei analize.

### Accesul la baza de date

Accesul la baza de date se realizează cu ajutorul unor programe de cereri scrise în limbajul de manipulare a datelor. Aceste programe au ca scop :

- încărcarea bazei de date
- actualizarea bazei de date
- regăsirea unei informații în baza de date.

Programele pot fi analizate de către procesorul limbajului de manipulare și apoi executate imediat sau pot fi catalogate în baza de date de către o componentă a produsului SOCRATE-MINI, bibliotecarul, și apoi apelate simplu și executate de către interpretorul de cereri.

Procesorul limbajului de manipulare se află prezent permanent în memorie sub forma unei biblioteci rezidente.

Suplimentar, este oferită posibilitatea accesului la baza de date din programe scrise în alte limbaje (COBOL, FORTRAN, MAC, etc.) prin apelarea unor subprograme SOCRATE, catalogate în bază. În același timp se permite apelarea din programe de cereri SOCRATE a unor subprograme scrise în alte limbaje, cît și posibilitatea comunicării cu interpretorul liniilor de comandă al sistemului de operare.

### Utilizatorii bazei de date

În raport cu modul de acces la baza de date se definesc trei categorii de utilizatori :

- administratorul bazei de date
- programatorul de aplicații
- utilizatori nespecialiști

Administratorul bazei de date are un rol complex și în principal el realizează :

- definirea schemei de memorare
- definirea structurii datelor
- asigurarea confidențialității bazei de date prin stabilirea de parole și drepturi de acces
- supravegherea funcționării curente a bazei de date
- reorganizarea bazei de date.

Programatorii de aplicații au în sarcină principală elaborarea programelor de aplicații. În plus aceștia mai au următoarele atribuții :

- asigurarea încărcării inițiale a bazei de date
- crearea pentru utilizatorii nespecialiști a unui limbaj cît mai apropiat de cel uman, pentru a facilita accesul la baza de date.

Pentru utilizatorii nespecialiști, SOCRATE-MINI permite construirea de macroinstrucțiuni și catalogarea acestora în baza de date cu ajutorul bibliotecarului.

Folosind macroinstrucțiunile, programatorul de aplicații poate defini pentru utilizatorii nespecialiști un limbaj mai apropiat de cel uman. Macroinstrucțiunile pot avea și parametri și sînt expandate de către analizorul programelor de cereri.

### Regăsirea datelor din baze de date SOCRATE distribuite în rețea

Regăsirea datelor din baze de date distribuite în rețea se poate realiza cu ajutorul următoarelor componente ale produsului SOCRATE-MINI :

- interfață pentru regăsirea datelor din baza de date SOCRATE-MINI distribuite într-o rețea de minicalculatoare. Această interfață permite apelarea unor subprograme SOCRATE catalogate într-o bază de date aflată într-un nod de rețea, de către un program de cereri SOCRATE sau alt limbaj de programare aflat în alt nod al rețelei. Interfața este prevăzută cu primitive care să permită executarea în paralel a mai multor programe aflate în noduri diferite.

- interfața pentru legătura în ambele sensuri dintre o bază de date SOCRATE-MINI și o bază de date SOCRATE aflată pe FELIX C-256. Interfața permite apelul unui subprogram catalogat într-o bază de date SOCRATE-MINI dintr-un program de cereri de pe FELIX C-256 și invers. Apelurile de subprograme și revenirea din subprograme sînt însoțite de transmitere de date.

### Întreținerea bazei de date

Pentru întreținerea bazei de date administratorul acesteia dispune de un număr de programe utilitare care au ca scop :

- formatarea domeniilor bazei de date ;
- restaurarea domeniilor bazei de date în caz de incident, utilizînd fișierul jurnal rapid ;
- schimbarea spațiului fizic al unuia sau mai multor domenii ;
- statistici efectuate asupra bazei de date.

## 2. DESCRIEREA BENZII DE LIVRARE

Banda ce conține produsul este obținută cu utilitarul BRU și are BACKUP SET-urile SOCMINI și SOCMINI-A (copie). Ea conține, în UFD-ul [1,55] componentele SOCRATE-MINI, în UFD-ul [1,100] un set de programe de test, iar în UFD-ul [1,200] fișierele ce conțin documentația.

### Componentele produsului

Componentele produsului, în UFD-ul [1,55], ocupă o dimensiune totală de circa 900 blocuri.

Acestea sînt :

- CADRE. TSK — bloc de date comune ce trebuie instalat înainte de deschiderea fiecărei sesiuni într-o partiție de tip COM cu numele „CADRE“ ;

-- *SOCLIB. TSK* — biblioteca partajabilă cu modulele reentrante ale procesorului limbajului de manipulare ; trebuie instalată într-o partiție de tip COM cu numele „*SOCLIB*“ ;

-- *DFS, SOC, MGS, GFM, LGI, LGO, ABS, MARKTM, RSS, STS, SAR, SOLE, LESO, LES, RES, LERESO, SOMIFE* — procesoare, module obiect sau programe utilitare.

Observație : Componentele de pe Felix C-256/1024, necesare pentru interfața SOCRATE-MINI — SOCRATE Felix, sînt livrate pe o bandă separată, obținută cu utilitarul MAINT, cu FN : *SOCMINI* și sînt grupate în biblioteci astfel :

-- *SOCMINI SOU, GN : 1, VN : 0*, ce conține procedura de generare a procesorului de interfață de la Felix, cu numele *GENESOMI* ;

-- *SOCMINI RBN, GN : 1, VN : 0*, ce conține modulele obiect necesare generării ;

*SOCMINI IMT, GN : 1, VN : 0*, ce conține două procesoare de interfață particulare :

– *SOMIFE15*, pentru SOCRATE V1.5 CII ;

– *SOMIFE16*, pentru SOCRATE V1.6.R ITC, cu observația că aceste două

versiuni particulare nu corespund cu certitudine ultimelor versiuni existente în țară în momentul actual.

## Programele de test

În UFD [1, 100] se găsește un set de programe de test, utile atît ca exemple de variante de sintaxă posibile, cît și pentru a demonstra buna funcționare a produsului în momentul livrării.

Aceste programe de test ocupă circa 1 100 blocuri și sînt organizate astfel :

o bază de date generată, dar neformatată, cu numele *FISDIR.DAT*, în care sînt definite 14 domenii, cu denumirile : *STR, PRG, DIC, FIS, D01, D02, ..., D10*. Aceste domenii, la execuția programelor de test, vor necesita un spațiu total de circa 2 200 blocuri.

-- o structură formată din 6 fișiere : *STR1.DEF—STR6.DEF* ;

-- 235 programe de test, cu denumirile *P001.SOC, P002.SOC, ...* și un program de încărcare a bazei de date cu numele *ÎNCARC.SOC*.

Programele de test se împart în următoarele categorii :

– *P107*, program ce se rulează separat, într-o sesiune cu fișier jurnal, pentru testul cererilor *PAUSE* și *CKPT* ;

– *P225—P227*, programe care se execută doar dacă softul de rețea este instalat și nodul a fost activat ;

– *P235*, program care se execută doar dacă este asigurată legătura cu un calculator Felix, unde este instalată o bază de date SOCRATE ;

– programe ce se execută cu baza de date goală : *P001—P055, P203* și *P219—P222* ;

– programe ce se execută cu baza de date încărcată : restul.

– cinci task-uri utilizate în exemplele de comunicare *COBOL-SOCRATE (P219), FORTRAN-SOCRATE (P220), SOCRATE-COBOL (EXTRAC), SOCRATE-FORTRAN (EXTRAF)* și *SOCRATE-MACRO-SOCRATE (AFLUIC)*. Aceste task-uri pot fi create și de utilizatori, fiind furnizate în acest scop fișierele sursă : *P219.CBL, P220.FTN, EXTRAD.CBL, EXTRAD.FTN, AFLUIC.MAC* și fișierele de comenzi indirecte pentru editarea de legături : *TKBC.CMD, TKBF.CMD* și *TKBM.CMD*.

- task-urile P226 și P227 pentru testarea apelării interfeței de rețea din COBOL, respectiv FORTRAN, pot fi create de utilizatori, fiind furnizate în acest scop fișierele sursa P226.CBL și P227.FTN și fișierul de comenzi indirecte TKBR.CMD ;
- fișiere de test pentru programul INCARC ;
- un fișier de comenzi indirecte, cu numele TEST.CMD, ce apelează alte două fișiere PROGOL.CMD și PROINC.CMD și care, în funcție de răspunsul la întrebarea „CE PROGRAME DORIȚI” poate realiza următoarele :
  - SEC0 : execuția tuturor programelor de test ;
  - SEC1 : execuția programelor ce necesită baza goală ;
  - SEC2 : execuția programelor ce necesită baza încărcată ;
  - Pxxx : execuția numai a programului Pxxx.

## Documentația

În UFD [1,200] se găsește un set de fișiere cu documentația produsului. Ele ocupă circa 1 000 blocuri și conțin următoarele :

Manualul de utilizare :

DEBUT.DOC  
INTUT.DOC  
STRUT.DOC  
CERUT.DOC  
MAGUT.DOC  
INLUT.DOC  
NETUT.DOC  
FELUT.DOC  
ERRUT.DOC  
SINUT.DOC  
KEYUT.DOC  
MODUT.DOC  
EXEUT.DOC

Manualul de operare :

DEBOP.DOC  
INTOP.DOC  
STROP.DOC  
CEROP.DOC  
MAGOP.DOC  
UTIOP.DOC  
INLOP.DOC  
NETOP.DOC  
FELOP.DOC  
ADMOP.DOC

Manualul de lansare versiune :

MANLA.DOC

## 3. INSTALAREA PRODUSULUI

Pentru începerea lucrului cu SOCRATE-MINI, trebuie să parcurse următoarele etape :

- crearea cu VMR a două partiții de tip COM :
  - „CADRE” de dimensiune 40 000 în octal (16 ko) ;
  - „SOCLIB” de dimensiune 311 000 în octal (101 ko) ;
- instalarea componentelor, în ordinea :
  - CADRE.TSK ;
  - SOCLIB.TSK ;
  - celelalte task-uri, cu excepția task-ului LES, a cărui instalare e opțională.

**Observații**

-- Dacă se dorește lucrul cu fișier jurnal rapid avînd punctele de securitate declanșate la intervale de timp, este de dorit ca programul MARKTM, apelat automat de LGI, program neevacuabil, să fie instalat într-o partiție diferită de partiția utilizator sau să fie fixat la începutul/sfîrșitul partiției utilizator, pentru a evita fragmentarea spațiului memoriei.

— Privitor la instalarea opțională a task-ului LES, citiți detaliile în Manualul de utilizare, cap. 5 și în Manualul de operare, cap. 6.

**4. NOUȚĂȚILE VERSIUNII V3.0\*****Creșterea performanțelor**

Față de versiunea precedentă, V2.1, prin rescrierea interpretorului limbajului de manipulare, adaptînd mai bine algoritmi folosiți de către acesta la caracteristicile minicalcutoarelor, s-a obținut o creștere a vitezei de execuție de 400—600%. Ținînd cont că la această creștere de viteză se poate adăuga o creștere obținută prin utilizarea facilităților versiunii 3.0 (comparații între mai multe tipuri de date, sporirea numărului de variabile Xi și Wi, posibilitatea de rotunjire la calculul expresiilor), considerăm cifrele menționate mai sus ca minimale.

**Interfața pentru regăsirea datelor dintr-o rețea de bază de date**

Această interfață permite apelul dintr-un program scris într-un limbaj oarecare (SOCRATE-MINI, COBOL, FORTRAN, PASCAL, BASIC, MACRO, etc.) a unuia sau mai multor programe precompilate catalogate în domeniile PRG ale unor baze de date SOCRATE-MINI aflate într-o rețea de minicalcutoare, rețea compatibilă soft cu DECNET V3.0.

La nivel logic, sînt furnizate cinci primitive de interfață cu rețeaua, primitive ce pot fi apelate cu APPEL din limbajul de manipulare SOCRATE-MINI sau cu CALL din celelalte limbaje :

— NT-CNT (NETCNT) — deschide accesul la baza de date din nodul specificat (verifică drepturile de acces la nod și la bază, apelează procesorul de interfață, îi furnizează acestuia valorile DSB, DSR, etc.) ;

— NT-SOC (NETSOC) — permite apelul unui precompilat fără a se sincroniza cu terminarea prelucrării din nodul destinație ;

— NT-SYN (NETSYN) — permite sincronizarea execuției task-ului apelant cu terminarea execuției programului precompilat din nodul de destinație ; cu ajutorul primitivelor NT-SOC și NT-SYN se permite posibilitatea multiprocesării într-o rețea de minicalcutoare ;

\* Menționăm că noutățile V4.0 — apărută în timpul cit lucrarea s-a aflat în tipografie — se tratează între paginile 257—264.

- NT-SCS (NETSCS) — permite apelul cu sincronizare pe terminarea prelucrării, a unui precompilat din nodul destinație ;
- NT-DSC (NETDSC) — închide accesul la baza de date din nodul de destinație și dacă aceasta era ultima bază deschisă din rețea, deconectează procesorul de la rețea.

În catalogul [1,55] sînt furnizate, pentru a asigura această interfață, componentele RES. TSK, LERESO.OBJ și LERESO.SKL.

În bateria de test, funcționarea interfeței este ilustrată de fișierele STR5.DEF, P225.SOC, P226.CBL și P227.FTN.

## Interfața SOCRATE-MINI – SOCRATE Felix

Această interfață permite apelul dintr-un program scris în limbajul de cereri SOCRATE-MINI a unuia sau mai multor programe precompilate catalogate într-o bază de date SOCRATE Felix. SGBD-ul de la Felix poate fi SOCRATE V1.5 CII, SOCRATE V1.6R ITC sau orice altă versiune care respectă regulile de apel ale interfeței COBOL-SOCRATE.

Din punct de vedere hardware, resursele necesare sînt :

- la Felix : CTQM ;
- la mini : interfața sincronă de tip DUP-11.

Legătura este de tip half-duplex, utilizînd protocolul TMM-UC.

Din punct de vedere software, resursele necesare sînt :

- la Felix : S.G.T. ;
- la mini : driverul pentru interfața sincronă, pentru alinierea la caracteristicile CTQM-ului, numit XWDRU, aparținînd pachetului TELEMINI.

La nivel logic, în limbajul de manipulare SOCRATE-MINI sînt furnizate trei primitive de legătură, primitive ce pot fi apelate cu instrucțiunea APPEL :

- FX-CNT — lansează task-ul de interfață, deschide linia, conectează procesorul de interfață de la Felix, îi transmite parametrul de apel, verifică drepturile de acces, deschide baza de date de la Felix (LOGIN) ;
- FX-EXC — transmite parametrul de apel, lansează în execuție programul precompilat dorit, catalogat în baza de date de la Felix, recepționează rezultatele ;
- FX-DSC — închide baza de date de la Felix (LOGOUT), închide linia, termină cele două task-uri de interfață.

Pentru a asigura această interfață, sînt furnizate :

- în catalogul [1,55] componenta SOCRATE-MINI, numită SOMIFE. TSK ;
- pe o bandă separată procedura de generare a procesorului de interfață de la Felix.

În bateria de test, funcționarea interfeței este ilustrată de fișierele STR6.DEF și P235.SOC.

## Calculul expresiilor aritmetice cu rotunjire

În V3.0, la calculul expresiilor aritmetice a fost introdusă opțiunea de rotunjire a rezultatului la numărul de zecimale al receptorului (echivalentă cu opțiunea ROUNDED de la instrucțiunea COMPUTE din COBOL).



Ca sintaxă, opțiunea este specificată prin operatorul de atribuire " = = " în locul operatorului " = ".

În bateria de test, opțiunea este ilustrată de programul P232.SOC.

### **Specificarea dinamică a numărului de zecimale la variabilele $W_1$**

Datorită introducerii opțiunii de rotunjire a rezultatului, cât și din motive de creștere a vitezei de evaluare a unei expresii aritmetice, a fost prefixat numărul de zecimale implicit al variabilelor  $W_1$  la 3. Însă prin aceasta, precizia obținută în calcule, cât și forma de editare a valorilor din variabilele  $W_1$  poate fi alta decât cea dorită de utilizatori. Pentru a evita acest lucru a fost introdusă o nouă instrucțiune, care are scopul de a modifica numărul de zecimale al unei variabile  $W_1$  în funcție de necesitățile utilizatorului.

Sintaxa instrucțiunii este

$$D W_1 = n \quad (0 \leq N \leq 18)$$

În bateria de test, opțiunea este ilustrată de programul P231.SOC.

### **Extinderea gamei de tipuri de comparații posibile**

Pînă în V2.1, în condiții, unul din membri trebuia să fie o variabilă  $Y_1$ ,  $W_1$ ,  $Z_1$ ,  $X_1$  sau constantă. În actuala versiune este permisă utilizarea în ambii membri a întregii game de tipuri de date, în măsura compatibilității lor.

Cele de mai sus se referă atît la condițiile din cadrul unei cereri SI cît și la cele din filtrul tip AYANT. Pentru filtrul tip AVEC nu a fost extinsă gama de comparații posibile.

În bateria de test, modificarea este ilustrată de programele P229.SOC și P230.SOC.

### **Protejarea implicită prin BLOQUER-LIBERER a cererilor G, S, SE**

Pînă în V2.1, pentru a evita incoerențele logice în cazul lucrului multi-utilizator, era necesară protejarea cererilor G, S, SE și a cererii M pentru referințe pe inel prin cererile BLOQUER-LIBERRE inserate explicit în program. În V3.0, protejarea acestor cereri se realizează în mod implicit de către procesorul limbajului de manipulare.

## Unificarea tratării erorilor

Pînă în V2.1, tratarea erorilor nefatale nu era unitară. Începînd cu V3.0, tratarea erorilor nefatale se realizează în mod unitar și anume : cîmpului receptor  $i$  se atribuie valoarea nedefinit, în cazul apariției unei erori pe parcursul execuției cererii respective.

Exemplu : în V2.1, atribuirea unei valori deja existente pentru o caracteristică cheie unică ducea la eventuala ștergere a valorii anterioare a cîmpului receptor, atît din bază cît și din dicționar și la inserarea noii valori doar în bază. În V3.0, în cazul unei chei unice deja existente, cîmpului receptor  $i$  se atribuie valoarea nedefinit și în bază. Observație : Singura excepție de la această regulă este pentru cererea

M referința = ...

care, în cazul unei erori nefatale, nu atribuie cîmpului receptor valoarea nedefinit, realizarea respectivă rămînînd atașată la același inel.

## O facilitate suplimentară la preluarea datelor cu EXT

Pentru a permite o mai facilă utilizare a preluării datelor cu ajutorul videoformatelor, a fost introdusă posibilitatea păstrării poziției cursorului la o cerere de forma

M  $Z_1$  = EXT ('text')

în cazul în care primul caracter din text este caracterul \$. Comportarea este similară și dacă 'text' a fost introdus în prealabil într-o variabilă  $Z_1$ .

Orice alt tip de variabilă sau caracteristică pentru cîmpul receptor, în afara unei variabile  $Z_1$  nu permite utilizarea acestei facilități.

În bateria de test, modificarea este ilustrată de programul P233.SOC.

## Posibilitatea preluării valorii ERREUR într-o variabilă $Y_1$

Pînă în V2.1, nu era posibilă preluarea valorii cîmpului ERREUR într-o variabilă  $Y_1$ , deoarece înainte de cererea M respectivă, variabila ERREUR era deja inițializată cu nedefinit. În V3.0 se poate prelua valoarea lui ERREUR într-o variabilă  $Y_1$  prin cererea

M  $Y_1$  = ERREUR

De asemenea, ea poate fi testată și printr-o cerere SI, cu obligația ca variabila ERREUR să apară în membrul stîng al primei condiții dintr-un eventual șir de condiții compuse.

## Creșterea numărului de variabile $X_1$ și $W_1$

Pentru sporirea posibilităților de lucru cu limbajul de manipulare, a fost crescut numărul variabilelor  $X_1$  la 15 și al variabilelor  $W_1$  la 20.

## Posibilitatea catalogării programelor cu fișier de securitate

Pînă în V2.1, nu era posibilă, la catalogarea programelor precompilate și a macroinstrucțiunilor cu MGS, reluarea catalogării dintr-un punct de securitate, în cazul unui incident. În V3.0, înainte și după catalogarea unui program precompilat sau a unei macroinstrucțiuni, în mod implicit bibliotecarul inițiază o cerere de creare a unui punct de securitate. Cînd condițiile cerute la deschiderea sesiunii cu fișier de securitate sînt îndeplinite, se crează punctul de securitate.

În cazul unui incident, domeniul PRG poate fi restaurat într-un punct de securitate cu ajutorul utilitarului RSS, rămînd ca, utilizînd funcțiile de listare ale bibliotecarului, utilizatorul să afle care a fost ultimul program catalogat.

## 5. PORTABILITATEA VERSIUNII V3.0

### Portabilitatea la nivelul limbajului sursă

... Variabilele  $W_1$  au numărul de zecimale predefinit (3). Precizia obținută în calcule, cît și forma de editare a valorilor din variabilele  $W_1$  a suferit modificări. Pentru a obține precizia sau forma de editare dorită, utilizatorul are la dispoziție cererea  $DW_1$ .

— Variabilele  $Z_1$  nu pot conține valoarea spațiu. Cererea  $MZ_1 = \text{spațiu}$  devine echivalentă cu cererea  $MZ_1 = U$ .

Codurile de eroare posibile de testat în variabila ERREUR : 8, 13, 19 nu mai există ; 8 a devenit echivalent cu 7, iar 13 cu 14.

— În tratarea cererii M, la detectarea unei erori nefatale se poziționează totdeauna receptorul pe nedefinit (excepție face M pentru referință).

### Portabilitatea la nivelul bazei de date

Bazele generate și încărcate sub V2.1 pot fi exploatate sub V3.0 numai cu următoarele restricții :

- structura trebuie recatalogată ;
- programele precompilate din domeniul PRG trebuiesc recatalogate ;
- pentru bazele de date instalate pe minicalculatoarele CORAL, datorită funcționării diferite a instrucțiunii cod mașină DIV față de cele din gama Independent-100, trebuiesc salvate domeniile de date cu utilitarul SAR V2.1 și restaurate cu utilitarul SAR V3.0.

## Portabilitatea la nivelul apelului componentelor

În V2.1, la terminarea cu LGO a unei sesiuni cu fișier de securitate, nu se crea punct de securitate, indiferent de opțiunea utilizată (/TR sau /AB).

În V3.0, la terminarea sesiunii cu opțiunea /TR se crează punct de securitate, în timp ce opțiunea /AB funcționează ca în V2.1.

## 6. SOCRATE-MINI PREZENTAT IN CIFRE

- Capacitatea maximă a bazei de date : 500 000 blocuri, adică peste 5 discuri de masă.
- Timp de acces punctual la bază : 1,5 ms.
- Memorie necesară :
  - partiție „CADRE” : 16 ko ;
  - partiție „SOCLIB” : 101 ko ;
  - zone de reentrănță pentru apelul procesorului limbajului de manipulare : 12—16 ko.
- Număr maxim de domenii posibile : 15 (inclusiv domeniile STR, DIC și PRG). Oricare din aceste domenii poate avea spațiul real definit la generare pe unul sau mai multe fișiere, dar numărul total de fișiere disc ce compun baza de date nu poate fi mai mare de 15.
- Dimensiunea maximă a spațiului virtual al unui domeniu :  $2^{31}$  octeți.
- Numărul de realizări maxim al unei entități :  $2^{24} - 1$ .
- Dimensiunea maximă a unei realizări :  $2^{24} - 1$  octeți.
- Număr maxim pliuri (raport între spațiul virtual și spațiul real) : 128.
- Număr de cadre : 30 în timpul unei sesiuni cu fișier de securitate și 31 în absența fișierului de securitate.
- Suma bufferelor și a zonelor de tip FORMAL utilizate simultan într-un program de cereri nu poate depăși 4,5 ko.
- Indicatori de eveniment globali utilizați : 33—37.
- Număr maxim de domenii în care pot fi declarate caracteristicile unei entități : 4.
- Număr maxim de imbricări posibile pentru caracteristici (entități-blocuri sau formale) : 4.
- Număr maxim de imbricări posibile în structură pentru etichete virtuale : 4
- Număr variabile de lucru :
  - 15 variabile  $X_1$  (exclusiv  $X_0$ ) ;
  - 30 variabile  $Y_1$  ;
  - 20 variabile  $W_1$  ;
  - 10 variabile  $Z_1$ .
- Număr maxim de apeluri imbricate ale instrucțiunii EXEC fără salvare de context : 30.
- Număr maxim de secvențe POUR imbricate : 10.
- Număr maxim de secvențe POUR cu filtru prin dicționar, imbricate : 4.

## 7. ANOMALII ȘI RESTRICȚII RIDICATE ÎN RAPORT CU V2.1

### Restricții ridicate

- Dacă lanțul de biți al unei entități era saturat și sosea o cerere  
G UN nume — entitate  
nu se semnală eroare. În V3.0, în acest caz este poziționată valoarea 20 în variabila ERREUR.
  - La generarea unui element pentru o caracteristică INVERSE, nu se făcea verificarea ca entitatea din membrul drept să fie chiar entitatea inversată conform structurii.
- De asemenea, la inserarea unei realizări într-un inel, prin cererea  
M referință DE... = opd<sub>2</sub>  
nu se făcea verificarea ca entitatea desemnată de opd<sub>2</sub> să fie chiar entitatea care conține inelul asociat referinței în structură.  
În V3.0, într-unul din aceste cazuri este furnizat codul de eroare 258.

### Anomalii înlăturate

- DFS : nu se execută controlul ca două referințe să nu puncteze un același ANNEAU. În V3.0 este furnizată în acest caz eroarea cu codul 39.
- DFS : dacă, dintr-o eroare de sintaxă, lipsea FIN de la opțiunea AVEC CLE, procesorul bucla.
- DFS : dacă la catalogarea unei structuri cu opțiunea /MAP, în timpul editării geografiei datelor era abortat procesorul DFS dintr-o cauză oarecare, baza rămânea blocată.
- SOC : nu putea fi utilizată cererea ATTACHE de mai mult de 12 ori într-un program.
- SOC : pe minicalculatoarele CORAL, datorită funcționării diferite a instrucțiunii cod mașină DIV față de minicalculatoarele I-100, se calcula în anumite cazuri eronat adresa paginii în cadrul bazei (anomalie introdusă în V2.1).
- SOC : la programele mari (cca 4 000 de linii, obținute în urma expandării macroinstrucțiunilor), apărea în timpul analizei mesajul de eroare

#### EROARE SGF PE FIȘIER DE MANEVRĂ TEMPORAR

- SOC : nu se făcea control asupra unui calificator X<sub>1</sub> decât dacă realizarea punctată de el cuprindea domeniul obiectului calificat.
- MGS : pînă în V2.1 era posibil ca o macroinstrucțiune și un program precompilat să aibă același nume. Aceasta avea ca efect expandarea nedorită a macroinstrucțiunii la o cerere EXEC nume. În V3.0, acest lucru nu mai este posibil.

- MGS : în anumite conjuncturi, la catalogarea unei macroinstrucțiuni se pierde un caracter, ceea ce avea ca efect o expandare eronată în momentul apelului macroinstrucțiunii.
- GFM : la formatarea unui domeniu care avea 20 000 de blocuri, utilitarul se aborta cu mesajul ODD ADRESS.
- GFM : la formatarea unui domeniu nu se semnală corect faptul că nu mai există suficient spațiu contiguu pe disc.
- LGI : nu se verifică decât primul domeniu dacă este contiguu.
- LGO : la închiderea unei sesiuni cu fișier de securitate funcție de timp, programul MARKTM rămîne activ încă cîteva minute.
- ABS : pentru utilizatorii neprivilegiați, ABS nu permitea abortarea procesoarelor SOC, DFS, MGS.
- SAR : la restaurarea unui domeniu, dacă acesta avea mai mult de 4 000 de blocuri, utilitarul se oprea cu un mesaj de eroare pe acces la domeniu.

## 8. RESTRICȚIILE VERSIUNII V3.0

- Nu funcționează cererea  $S_i X_i$  decât pentru cazul cînd  $X_i$  conține adresa unui FORMAL.
- Cîmpurile FORMAL de tip PACHE nu sînt implementate.
- Nu funcționează cererea  $M X_i = \text{SUIVANT DE} \dots \text{AVEC sau PAR} \dots$ .
- Nu funcționează opțiunea RAD50 pentru caracteristici de tip MOT.
- Nu se poate declara o caracteristică ANNEAU decât dacă i s-a prevăzut o caracteristică REFERE asociată.
- Nu se face verificarea existenței unei cereri POUR sau FAIRE pe un nivel superior unei cereri REFAIRE, SORTIE sau SUIVANT.
- La FORMAL de tip repetitiv indexat prin  $Y_i$  nu se face verificarea de zero, nedefinit și marja superioară.
- Nu se verifică dacă fiecărei cereri PAUSE îi corespunde o cerere CKPT.
- Nu se verifică dacă fiecărei cereri BLOQUER îi corespunde o cerere LIBERER.
- La o cerere ATTACHE, nu se verifică dacă BANDE n este deja utilizat de un ATTACHE anterior.
- La catalogarea unei macroinstrucțiuni cu MGS, o constantă de tipul 'text' nu poate avea pe prima sau pe ultima poziție caracterul „:”. Pentru evitarea erorii lexicale, este suficientă adăugarea unui blank la șirul de caractere drept prim, respectiv ultim caracter.
- La o cerere de forma

$M X_i = \text{SUIVANT DE} \langle \text{citare} \rangle$

dacă citarea referă o caracteristică imbricată, aceasta trebuie calificată fie direct printr-o variabilă  $X_i$ , fie în mod explicit pînă la ultimul nivel.

## 1. INTRODUCERE

*SOCRATE-MINI* este un sistem de gestiune a bazelor de date pentru minicalculatoarele din gama INDEPENDENT-100, CORAL și, în general, pentru orice minicalculator compatibil cu PDP 11/34. Produsul asigură organizarea și gestiunea unei colecții de date astfel încât să fie respectate toate dezideratele unei baze de date :

- necesitatea ca mai mulți utilizatori să aibă acces simultan la aceeași colecție de date ;
- organizarea colecției de date astfel încât să fie satisfăcute în mod optim cerințele de prelucrare specifice fiecărui utilizator ;
- redondanță minimă în colecția de date ;
- independența programelor de aplicație față de modul de organizare a datelor, astfel încât dezvoltarea aplicațiilor să nu afecteze programele deja realizate ;
- mai mult, posibilitatea dezvoltării aplicațiilor și a completării colecției de date fără a reorganiza informațiile deja încărcate în baza de date ;
- înregistrarea nu numai a informațiilor, ci și a relațiilor dintre ele (de tip arborescent, de tip rețea) ;
- securitatea colecției de date ;
- discreția, relativă la accesul la date.

### Componentele SOCRATE-MINI

Prezentăm pe scurt componentele SGBD SOCRATE-MINI și funcțiile fiecăreia dintre ele.

— *Generatorul schemei de memorare* — *GFM* — reține într-un fișier director informații despre domeniile componente ale bazei de date și despre modul de implantare a acestor domenii pe discul magnetic. Partajarea în domenii permite, printre altele, montarea și activarea parțială a bazei de date pe durata unei sesiuni.

În plus, *GFM* realizează și scrierea inițială a domeniilor (formatare).

— *Procesorul de structură* — *DFS* — permite memorarea structurii datelor în baza de date, într-un domeniu distinct.

Structura datelor conține informații despre :

- denumirea și natura câmpurilor elementare din baza de date ;
- apartenența acestor câmpuri la diversele domenii ;
- relațiile dintre câmpuri (liniare, arborescente, de tip rețea) ;
- bufferele de memorie utilizate în programele de cereri.

— *Procesorul limbajului de manipulare* — *SOC* — permite accesul la baza de date prin intermediul unor programe scrise în limbajul de manipulare a datelor. În particular, această componentă mai realizează :

— validări automate la nivelul bazei și nu al tratamentului ;

— definirea și utilizarea unor macroinstrucțiuni, prin intermediul cărora sînt posibile, pe de o parte scurtarea programelor (evitarea secvențelor repetitive) pe de altă parte crearea unui limbaj cît mai apropiat de limbajul uman, pentru utilizatorii nespecialiști ;

— sincronizări la nivelul utilizatorilor pentru crearea punctelor de reluare și asigurarea securității bazei de date.

— *Programul bibliotecar* — *MGS* — permite catalogarea în baza de date, într-un domeniu distinct, a macroinstrucțiunilor și a programelor precompilate. Produsul realizează gestiunea completă a domeniului respectiv : ștergeri de programe, actualizări, rezumate.

— *Programe și subprograme de interfață* :

— *LES, LESO, SOLE* — asigură interfața limbaje evolute — *SOCRATE-MINI* ;

— *RES, LERESO* — asigură interfața pentru regăsirea datelor dintr-o rețea de baze de date ;

— *SOMIFE* — asigură interfața *SOCRATE-MINI* — *SOCRATE Felix*.

— *Programe utilitare pentru controlul unei sesiuni SOCRATE-MINI*. Acestea realizează :

— *LGI* — deschiderea unei sesiuni ;

— *LGO* — închiderea unei sesiuni ;

— *ABS* — abortarea programelor *SOCRATE-MINI*.

— *Programe utilitare pentru întreținerea unei baze de date*. Acestea realizează :

— *RSS* — restaurarea bazei de date după producerea unui incident, utilizînd fișierul de securitate rapid ;

— *STS* — editarea statisticii bazei de date și, la cerere, a schemei de memorare ;

— *SAR* — salvarea — restaurarea bazei de date în vederea schimbării spațiului real.

Componentele prezentate mai sus se pot împărți în două categorii :

— *produse la dispoziția programatorilor de aplicații* : procesorul de structură, procesorul limbajului de manipulare, bibliotecarul și programele de interfață (cu observația că programatorii nu vor face catalogări în structură decît sub controlul administratorului bazei de date). Aceste produse primesc în intrare fișiere sursă cu programe care descriu funcția dorită. Ele nu pot fi apelate decît pe durata unei sesiuni *SOCRATE-MINI*.

— *produse la dispoziția administratorului bazei de date* : generatorul schemei de memorare și programele utilitare. Acestea realizează funcțiile dorite în urma unui dialog la terminal. Majoritatea se apelează în afara sesiunii *SOCRATE-MINI*.

Produsele din prima categorie vor purta uneori în manual denumirea globală de *procesoare SOCRATE-MINI*, iar cele din a doua categorie denumirea de *programe utilitare*.

În Manualul de utilizare nu sînt prezentate decît funcțiile și posibilitățile procesoarelor, singurele care primesc în intrare programe sursă. Funcțiile programelor utilitare vor fi prezentate, împreună cu modul de apel și dialogul la terminal, în Manualul de operare.



## Notății și convenții utilizate în manual

În descrierea limbajelor din manual, atât în prezentarea sintaxei instrucțiunilor, cât și în exemple, vor fi utilizate următoarele notații și convenții:

- MAJUSCULE — sînt utilizate pentru desemnarea cuvintelor cheie din limbaj ; acestea, ca și caracterele speciale, trebuie scrise în programe fără nici o modificare față de forma prevăzută în sintaxă ;
- Minuscul — desemnează în general identificatori sau constante ; ele indică în sintaxă natura elementului respectiv, fără a preciza valoarea sa ;
- [ ] — parantezele drepte indică prezența opțională a parametrului din interior ;
- { } — acoladele indică faptul că elementele din interior dispuse pe verticală, pot fi prezente alternativ.
- -- — elementul subliniat indică valoarea implicită în cazul cînd un parametru este opțional și poate lua alternativ mai multe forme ; absența completă a unui element echivalează cu prezența cuvîntului cheie subliniat ;
- punctele de suspensie indică repetarea elementului sintactic precedent de un număr de ori nedeterminat ;
- <RET> — reprezintă tasta CARRIAGE-RETURN.

## 2. LIMBAJUL DE DESCRIERE A STRUCTURII

### Generalități

#### Apelul procesorului limbajului de descriere

După deschiderea sesiunii SOCRATE cu LGI, DFS — procesorul limbajului de descriere a structurii — poate fi apelat din orice terminal cuplat la sistem (care a deschis o sesiune „HELLO” de lucru sub sistemul de operare).

*Linia de comandă pentru apel are următoarea formă :*

>DFS fișier-ieșire/SW<sub>1</sub> = fișier-intrare<sub>1</sub>/SW<sub>2</sub>, fișier-intrare<sub>2</sub>/SW<sub>3</sub>...

fișier-intrare — este specificatorul unui fișier sursă ce conține caracteristicile de catalogat în structură ;

fișier-ieșire — este specificatorul fișierului ce va conține :

— listarea fișierului sursă

— mesaje de eroare

-- un eventual tabel cu „geografia datelor” solicitat prin comutatorul MAP.

- SW<sub>1</sub>** — sînt comutatori ataşaţi fişierului de ieşire :
- MAP** solicită editarea tabelului cu „geografia datelor“, ce ilustrează modul de implantare a caracteristicilor în spaţiul virtual ;
  - SP** solicită listarea fişierului de ieşire la LP ;
- SW<sub>2</sub>** — sînt comutatori ataşaţi unui fişier de intrare :
- SA** (Subpage Alignment) — solicită ca implantarea caracteristicilor în spaţiul virtual să fie astfel realizată încît o caracteristică simplă să fie cuprinsă într-o aceeaşi subpagină ;
  - CVF** (Convenţional Format) — determină luarea în consideraţie doar a primelor 60 de caractere dintr-un articol al fişierului de intrare.

**Elemente lexicale : setul de caractere, separatorii limbajului de descriere**

**SETUL DE CARACTERE** utilizat în limbajul de descriere este format din :

- litere majuscule A—Z
- cifre zecimale 0—9
- caractere speciale :
  - + (plus)
  - (minus sau linie)
  - .
  - \$ (dolar)
  - (asterisc)
  - ( (paranteză stîngă)
  - ) (paranteză dreaptă)
  - ? (semnul întrebării)
  - (blanc)
  - (tab)
  - / (slash)

Observaţie : În afară de acestea, valorile definite în cadrul unei liste de valori pot conţine orice alt caracter aparţinînd setului ASCII.

Din caracterele de mai sus, singurele (simple sau compuse) acceptate drept **SEPARATORI** sînt :

- (blanc)
- (tab)
- ( (paranteză stîngă)
- ) (paranteză dreaptă)
- ? (semnul întrebării)

**Elemente sintactice : cuvinte cheie, identificatori, etichete virtuale, constante**

Un **CUVÎNT CHEIE** este un element al limbajului de descriere, care poate fi utilizat acolo şi numai acolo unde apare explicit în sintaxă. În manual, cuvintele cheie, atît în descrierea sintaxei, cît şi în exemple, vor fi scrise cu majuscule. Cuvintele cheie ale limbajului de descriere sînt prezentate în anexa A.

Un **IDENTIFICATOR** este o succesiune de maximum 30 de caractere alfanumerice (litere, cifre, liniuţă), dintre care primul este obligatoriu literă. Nu pot fi utilizate drept identificatori :

- cuvinte cheie ale limbajului de descriere ;

- cuvinte cheie ale limbajului de manipulare ;
- variabile  $X_1, Y_1, W_1, Z_1$  utilizate în limbajul de manipulare ;
- denumiri de programe sau de macro-uri.

ETICHETELE VIRTUALE sînt de două tipuri :

- etichete virtuale de început, de forma \$xxx și
- etichete virtuale de sfîrșit, de forma \*xxx,

unde xxx sînt trei caractere alfanumerice (litere sau cifre) dintre care primul literă.

CONSTANTELE limbajului de descriere pot avea una din următoarele forme :

- constante numerice întregi :

$$\left\{ \begin{array}{c} + \\ - \end{array} \right\} nn \dots n$$

unde  $nn \dots n$  este o succesiune de maximum 18 cifre zecimale.

- constante numerice fracționare :

$$\left\{ \begin{array}{c} + \\ - \end{array} \right\} \text{parte-întreagă, parte-zecimală}$$

unde parte-întreagă și parte-zecimală sînt succesiuni de cifre zecimale, una din ele putînd fi eventual vidă.

Numărul de cifre dintr-o constantă numerică nu poate depăși 18 ;

-- valori, ce fac parte dintr-o listă de valori : sînt succesiuni de maximum 30 caractere ASCII, mai puțin separatorii limbajului de descriere.

### Noțiunea de caracteristică, clasificări

Limbajul de descriere permite :

- pe de o parte, de a defini, pentru o informație simplă, numele și natura sa ;
- pe de altă parte, de a descrie, pentru un ansamblu de informații simple, structuri arborescente și structuri de tip rețea.

O informație complexă se poate considera că este alcătuită din date elementare. De exemplu, o Persoană poate fi definită prin datele următoare :

- nume,
- vîrstă,
- stare socială,
- număr de copii,
- întreprinderea unde lucrează etc.

Se zice că toate aceste date elementare descriu CARACTERISTICILE unei Persoane.

Se constată că aceste caracteristici simple pot fi de mai multe tipuri :

de tip *cuvînt* (sau *MOT*) ; de ex. : nume, întreprindere ;

de tip *NUMERIC* ; de ex. : vîrstă, număr de copii, retribuție ;

de tip *LISTĂ DE VALORI* ; de ex. : „stare socială“ poate avea un număr finit de valori posibile : căsătorit, necăsătorit, văduv.

Un număr oarecare de caracteristici se poate reuni sub un același identificator. De exemplu :

```
Data-nașterii
  An
  Lună
  Zi
```

„Data-nașterii” se zice că este o caracteristică de tip *BLOC*.

„Persoana” descrisă mai sus, fiind o reuniune de caracteristici, se poate considera că este și ea o caracteristică grupată. Diferența față de o caracteristică de tip bloc este că în baza de date nu se vor găsi informații despre o singură persoană, ci despre mai multe, deci trebuie rezervat loc sub forma de vector sau de matrice. O astfel de caracteristică grupată repetitivă se numește caracteristică de tip *ENTITATE*, iar fiecare element al vectorului astfel obținut se numește *REALIZARE* a entității respective.

O parte din realizările unei entități pot avea o proprietate comună. De exemplu, dintre persoanele descrise, o parte sînt din același județ, Constanța. Atașăm deci entității „Persoana” o caracteristică „Constănțean” care să ne selecteze automat dintre toate realizările entității, numai realizările ce conțin constănțeni, fără a fi obligați să analizăm persoană după persoană și să vedem unde s-a născut. O astfel de caracteristică se cheamă că e de tip *INVERSE*.

O entitate poate fi și ea la rîndul ei o caracteristică în cadrul unei alte entități (de asemenea și blocurile). Se definesc astfel *ENTITĂȚILE INCLUDE*. În spațiul virtual al unei baze de date, care este liniar, să ne închipuim o entitate „B” cu 3 realizări posibile, inclusă într-o entitate „A” cu 4 realizări posibile (fig. VI.2.1).

Prin intermediul entităților incluse se realizează astfel structuri arborescente. Ca să realizăm structuri de tip rețea se definesc alte două tipuri de caracteristici: *INEL* și *REFERINȚĂ*. Inelul, conținut într-o realizare a unei entități, este un punctator către un lanț de realizări al altei entități. Referința este o caracteristică definită în a doua entitate ce permite atît înlănțuirea realizărilor din lanț între ele, cît și punctarea, din fiecare realizare a lanțului, către realizarea primei entități care are inelul. Legăturile realizate se pot descrie schematic astfel, pentru cazul cînd, la realizarea a doua a entității „A” care conține inelul sînt legate realizările 3, 5 și 6 ale entității „B” care conține referința (fig. VI.2.2):

În felul acesta presupunînd că entitatea „A” conține date despre secții, iar entitatea B date despre persoane, se poate ca, pornind de la o secție (a doua) să se regăsească informații despre toate persoanele care lucrează în acea secție și, invers, pornind de la o persoană, să se afle imediat în ce secție lucrează.

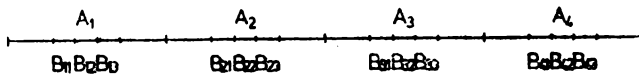


Fig.VI.2.1

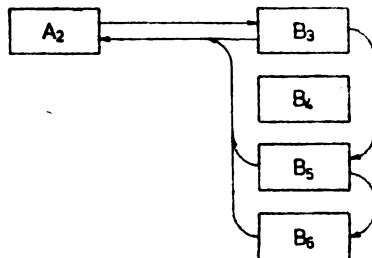


Fig. VI.2.2.

În concluzie, din punct de vedere al limbajului de descriere, caracteristicile se pot clasifica astfel:

- Caracteristici simple ce definesc natura informațiilor :
  - Cuvînt (MOT)
  - Valori numerice
  - Listă de valori
- Caracteristici grupate :
  - Bloc
  - Entitate
- Caracteristici simple ce definesc și realizează legături între entități :
  - Inversă
  - Inel
  - Referință

În schimb, vom vedea că limbajul de manipulare privește caracteristicile ca și cum ar fi astfel clasificate :

- Caracteristici simple :
  - Cuvînt (MOT)
  - Valori numerice
  - Listă de valori
- Caracteristici compuse :
  - Bloc
  - Realizare de entitate
  - Referință (deoarece punctează către o realizare de entitate și anume, cea care conține inelul).
- Mulțimi de caracteristici compuse :
  - Entitate (este o mulțime de realizări)
  - Inversa (deoarece permite accesul la mai multe realizări ale unei entități, realizări care au o proprietate comună)
  - Inel (deoarece punctează către un lanț de realizări ale entității care conține referința).

### Noțiunea de domeniu

Utilizată la nivelul limbajului de descriere, o „*etichetă virtuală*” precizează un spațiu virtual în care figurează valorile caracteristicilor care o urmează. Un astfel de spațiu virtual poartă denumirea de *DOMENIU*.

Etichetele virtuale se utilizează astfel :

Setic <listă de caracteristici> \*etic

unde :

- „etic” = 3 caractere reprezentînd un nume logic al domeniului ;
- <listă de caracteristici> cuprinde caracteristicile ce se doresc a fi definite în spațiul virtual respectiv.

Pe plan fizic, unui domeniu îi corespunde :

- fie un singur fișier ;
- fie mai multe fișiere în sensul de exploatare al sistemului.

Avantajele partajării bazei de date în domenii sînt :

- montaj parțial al bazei de date ;
- fuzionarea mai multor baze ;
- ameliorarea performanțelor (viteză, ocupare spațiu disc) printr-o repartiție mai bine controlată a diverselor valori conținute într-o bază ;
- extensie facilă a bazei de date : adăugare de noi caracteristici la o entitate veche, într-un domeniu nou ;
- securitate : datele stabile pot fi separate de datele variabile, în domenii diferite ;
- confidențialitate.

### Noțiunea de formal

O caracteristică de tip *FORMAL* este o caracteristică fictivă, ceea ce înseamnă că în momentul definerii nu îi este asociată nici o rezervare nici în baza de date, nici în memorie. Ea permite descrierea unei structuri arborescente pentru o înregistrare de tip COBOL.

Într-o astfel de înregistrare pot fi descrise cîmpuri numerice (în diverse formate), cîmpuri alfanumerice, cîmpuri grupate, simple sau repetitive și redefiniri de zone.

La nivelul limbajului de manipulare, acestui format logic de înregistrare (notat pe scurt formal) i se va asocia un buffer în memoria centrală. Cîmpurile descrise în formal vor putea fi în acest fel utilizate ca zone de lucru în memorie, suplimentare față de cele oferite prin variabilele  $Y_1$ ,  $Z_1$ ,  $W_1$ .

În particular, formalele permit utilizarea operațiilor de intrare-ieșire la nivel de fișiere. În limbajul de manipulare, asociind un formal unui buffer, se realizează descrierea unei zone-articol pentru fișierul ce se dorește a fi prelucrat.

### Forma generală a unui program de descriere a structurii

Limbajul de descriere permite atât catalogarea inițială a unei structuri, cât și adăugarea ulterioară de noi caracteristici.

Forma generală a unui program de descriere pentru *catalogarea inițială* a unei structuri este :

```

[$FIS]
DEBUT
.
.   caracteristici bază de date
.
[ .   caracteristici formale ]
.
FIN
[*FIS]
?
```

Etichetele virtuale \$FIS și \*FIS sînt opționale și au rol de comentariu. Și în mod implicit și în mod explicit, domeniul de bază al caracteristicilor bazei de date

la o catalogare inițială este domeniul FIS. În interior se poate solicita ca anumite caracteristici ale bazei de date să aparțină la alte domenii, utilizând etichete virtuale.

Forma generală a unui program pentru *adăugarea de caracteristici* la structură, este :

```
[ $dom ]
D
.
[ . caracteristici bază de date ]
.
[ . caracteristici formale ]
.
FIN
[ *dom ]
?
```

Cuvîntul „D” arată că e vorba de o adăugare de structură. Etichetele virtuale „\$dom” și „\*dom” definesc domeniul de bază al caracteristicilor. În mod implicit, acesta este domeniul FIS. Însă indiferent care este domeniul de bază (FIS sau altul, definit explicit) în interior se poate solicita ca anumite caracteristici ale bazei de date să aparțină la alte domenii, utilizînd etichete virtuale.

*LINIILE SURSĂ* au un format de scriere complet liber. Trebuie totuși ținut cont că ele nu sînt concatenate, deci un element sintactic (cuvînt cheie, identificator, constantă etc.) trebuie să fie conținut în întregime într-o aceeași linie sursă.

*IDENTIFICATORII* limbajului de descriere sînt utilizați de limbajul de manipulare pentru lucrul cu caracteristicile bazei de date. Din acest motiv, pentru a elimina orice ambiguitate la nivelul formulării cererii, se interzice prezența la același nivel (într-un același bloc, entitate sau formal) a două caracteristici cu identificatori identici.

De asemenea, reamintim că nu pot fi utilizate drept identificatori :

- cuvinte cheie ale limbajului de descriere ;
- cuvinte cheie ale limbajului de manipulare ;
- variabile  $X_1, Y_1, W_1, Z_1$  ;
- denumiri de programe sau macro-uri.

*COMENTARIILE* pot fi inserate în programul de descriere în două moduri :

- fie prin linii comentariu, care încep cu caracterele /# ;
- fie pe marginea programului, începînd cu coloana 61, caz în care la apelul procesorului va fi utilizat comutatorul /CVF.

## Caracteristici ale bazei de date

### Caracteristica de tip MOT

Definește un lanț de caractere oarecare în cod ASCII.

*Sintaxa*

Identificator MOT  $\left[ \left\{ \left\{ \begin{array}{l} \text{întreg} \\ (\text{întreg}) \end{array} \right\} \right\} \right]$  [RAD50]

unde :

— „întreg“ este un cuvînt cuprins între 1 și 30 ; acest număr determină lungimea lanțului de caractere. Parantezele sînt facultative. În cazul absenței parametru-lui, valoarea implicită este 30 ;

— RAD50 este un cuvînt cheie utilizat cînd lanțul conține numai caractere de tip RADIX-50 : litere, cifre, \$ (dolar), . (punct), B (blanc) și — (liniuță). În acest caz, caracteristica va fi memorată în bază în cod RADIX-50 și va ocupa, deci, numai două treimi din lungimea necesară în mod normal.

### Exemple

```
nume-persoană MOT 20 RAD50
întreprindere MOT
unitate-de-măsură MOT (10)
```

### Comentarii

— În limbajul de manipulare, caracteristica de tip MOT poate face obiectul unor con-părări de tip lanț de caractere. Compararea se efectuează de la stînga la dreapta ; dacă două lanțuri sînt de lungimi inegale, cel mai scurt este completat cu blankuri la dreapta.

— Deși orice caracter ASCII poate fi acceptat într-o valoare de tip MOT, trebuie totuși utilizat cu prudență caracterul ' (apostrof), datorită utilizării lui în limbajul de manipulare pentru definirea constantelor de tip Z.

— Caracteristicile de tip MOT cu opțiunea RAD50 nu pot fi declarate chei.

### Imagine în spațiul virtual

Sînt rezervați atîția octeți cîți sînt necesari pentru memorarea valorii maxime a caracteristicii în cod ASCII sau RADIX-50, plus un octet pentru lungimea reală a valorii în bază, care îi precede.

Din exemplele de mai sus, „nume-persoană“ ocupă 15 octeți (două treimi din 20, plus 1 octet pentru lungime), „întreprindere“ ocupă 31 octeți, „unitate-de-măsură“ 11 octeți.

### Caracteristica de tip valoare numerică bornată

Definește o valoare numerică cuprinsă între două limite, borna inferioară și borna superioară. Deși sintaxa are două forme distincte, una pentru numere întregi, alta pentru numere zecimale, ele sînt prezentate împreună, deoarece țatît memoria în structură, cît și în baza de date este comună.

Sintaxa pentru valori numerice întregi

```
Identificator [BINAIRE] DE  $\begin{bmatrix} + \\ - \end{bmatrix} nr_1$  A  $\begin{bmatrix} + \\ - \end{bmatrix} nr_2$   $\left\{ \begin{matrix} nr_3 \\ (nr_3) \end{matrix} \right\}$ 
```

unde :

—  $nr_1$ ,  $nr_2$  și  $nr_3$  sînt numere întregi, cu max. 18 cifre, mai precis cu valori :

$$-(2^{64} - 1) \leq nr_1 \leq (2^{64} - 1)$$

—  $nr_1$  și  $nr_2$  sînt borna inferioară, respectiv superioară. Ele vor fi utilizate în limbajul de manipulare la validarea automată a valorilor numerice înscrise în baza



de date. În cazul absenței parametrului  $nr_3$ , diferența  $(nr_2 - nr_1)$  este utilizată la rezervarea de spațiu virtual. Ea trebuie să fie :

$$(nr_2 - nr_1) \leq 2^{64} - 1$$

—  $nr_3$  este maximum prevăzut pentru borna superioară. Când parametru<sup>1</sup> este prezent, diferența  $(nr_3 - nr_1)$  este utilizată la rezervarea de spațiu virtual. Trebuie ca :

$$nr_3 > nr_2 \text{ și } (nr_3 - nr_1) \leq 2^{64} - 1$$

*Sintaxa pentru valori numerice zecimale*

Identificator **DECIMAL**  $\left\{ \begin{array}{l} n_1 [V n_2] \\ (n_1 [V n_2]) \end{array} \right\}$  DE  $\left[ \begin{array}{c} + \\ - \end{array} \right] nr_1$  A  $\left[ \begin{array}{c} + \\ - \end{array} \right] nr_2$

unde :

—  $n_1$  = număr maxim de cifre pentru partea întregă :

$$n_1 \leq 18$$

—  $n_2$  = număr maxim de cifre pentru partea zecimală :

$$n_2 \leq 18$$

În cazul absenței opțiunii „V  $n_2$ “, valoarea implicită pentru  $n_2$  este 0 și caracteristica este tratată ca o valoare numerică întregă.

—  $n_1$  și  $n_2$  trebuie să îndeplinească condiția

$$1 \leq (n_1 + n_2) \leq 18$$

—  $nr_1$  și  $nr_2$  sînt borna inferioară, respectiv superioară. Ele sînt numere zecimale cu max.  $n_1$  cifre întregi și max.  $n_2$  cifre zecimale. Diferența  $(nr_2 - nr_1)$  este utilizată la rezervarea de spațiu virtual. De asemenea  $nr_1$  și  $nr_2$  vor fi utilizate de către procesorul limbajului de manipulare la validarea automată a valorii numerice înscrise în bază.

### Exemple

Cod material	DE 1 A 200 (300)
Temperatură	BINAIRE DE -70 A +150}
Preț	DECIMAL (5 V 2) DE 0 A 20000
Diametru	DECIMAL 0 V 7 DE 0.0001 A 0.5
Gradație	DECIMAL (3) DE -70 A 150

### Comentarii

— În programele de cereri, caracteristicile de tip valoare numerică pot face obiectul unor comparări și operații aritmetice.

— În primul exemplu, „Cod-material“ nu va putea lua valori mai mari de 200. Dar mai târziu, dacă va fi nevoie, structura va putea fi schimbată încît borna superioară să poată crește pînă la 300, fără nici o schimbare a domeniului respectiv din baza de date.

— Deși diferite ca sintaxă „Temperatură“ și „Gradație“ sînt de fapt identice din punct de vedere al memorării în structură și al rezervării de spațiu virtual.

— În opțiunea „V  $n_2$ “, V este cuvînt-cheie, nu separator al limbajului. Din acest motiv el va fi totdeauna încadrat de alți doi separatori (blancuri).

*Imagine în spațiul virtual*

Bornele și numărul de poziții întregi și zecimale sînt reținute în structură. În bază se vor rezerva atîți octeți (între 1 și 8) cîți sînt necesari pentru a memora diferența între borne ( $nr_2 - nr_1$ ) sau, în cazul prezenței parametrului  $nr_3$ , diferența ( $nr_3 - nr_1$ ).

De exemplu, pentru „Cod-material“ se rezervă 2 octeți, necesari pentru memorarea valorii maxime 300, deși deocamdată nu se pot înscrie în bază valori mai mari de 200, pentru care ar fi necesar un singur octet.

În bază se va înscrie efectiv nu valoarea numerică reală, ci diferența între ea și borna inferioară.

**Caracteristica de tip listă de valori**

Permite ca dintr-o listă finită de valori punctuale, în baza de date să fie înscris un element oarecare al acestei liste, prin numărul său de ordine.

*Sintaxa*

Identificator ( $n_1 [n_2]$ ) (valoare<sub>1</sub> valoare<sub>2</sub> ... valoare<sub>n</sub>)

unde :

- $n_1$  = numărul maxim de valori din listă

$$1 \leq n_1 \leq 254$$

- $n_2$  = lungimea maximă a unei valori

$$1 \leq n_2 \leq 30$$

În absența parametrului  $n_2$ , este considerată valoarea implicită 30 ;

– valoare<sub>1</sub> = un lanț de caractere ASCII (cu excepția separatorilor de limbaj) de lungime maximă 30.

**Exemple**

Stare-civilă (5 12) (căsătorit necăsătorit văduv)

Diametru (6 5) ( 10 11.25 13 16.5 0.7 )

Medicament (100) (Penicilină Acid-acetilsalicilic...)

*Comentarii*

– În limbajul de manipulare, caracteristica de tip listă de valori poate face obiectul unor comparații de tip lanț de caractere. Compararea se face de la stînga la dreapta; dacă cele două lanțuri sînt inegale, cel mai scurt este completat cu blăncuri la dreapta.

– În exemplul al doilea, întimplător toate elementele listei sînt valori numerice. Dar ele nu sînt tratate decît tot ca lanțuri de caractere, deci valorile „+10” sau „10.0” nu vor fi considerate ca făcînd parte din lista de valori, deși valoarea „10” există.

*Imagine în spațiul virtual*

Lista de valori este reținută în structură. În spațiul virtual se rezervă totdeauna un octet. Înscrierea unei valori în baza de date constă de fapt în memorarea, în acest octet, a numărului de ordine al valorii în listă.

**Caracteristica de tip FILLER**

Permite rezervarea de spațiu virtual, fie pentru o utilizare ulterioară, fie din motive de aliniere.

*Sintaxa*

Identificator **FILLER**  $\left\{ \begin{array}{l} n_1 \\ (n_1) \end{array} \right\}$

unde :

—  $n_1$  este dimensiunea în octeți a zonei ce se dorește rezervată în spațiul virtual

$$1 \leq n_1 \leq 65535$$

**Exemplu**

**Rezervat FILLER 70**

*Comentarii*

— FILLER este cuvânt cheie, deci nu poate fi utilizat drept identificator.  
 — Caracteristica de tip FILLER este singura care nu este reținută în structură. Singurul ei efect este rezervarea de spațiu virtual, fără a-i da acestui spațiu un nume. Drept urmare, evitându-se denumirile caracteristicilor obișnuite, toate caracteristicile FILLER din baza de date pot purta același identificator, fie că sint sau nu la același nivel.

*Imagine în spațiul virtual*

Se rezervă „ $n_1$ ” octeți care nu pot fi utilizați în limbajul de manipulare. Pot fi în schimb ulterior utilizați, prin suprapunerea într-o nouă structură a unor caracteristici obișnuite, fără a schimba domeniul corespunzător din baza de date.

**Caracteristica de tip entitate**

Este o caracteristică compusă repetitivă, care permite regruparea mai multor caracteristici sub un același identificator, și repetarea acestei grupări de un număr de ori.

*Sintaxa*

**ENTITE**  $\left\{ \left\{ \begin{array}{l} \text{întreg} \\ ((\text{întreg})) \end{array} \right\} \right\}$  Identificator **DEBUT** <listă-caracteristici> **FIN**

unde :

— *întreg* definește numărul maxim de realizări posibile. În lipsa parametruului, valoarea implicită este 1.

$$1 \leq \text{întreg} \leq 2^{24} - 1$$

**Exemplu**

**ENTITE (100) întreprindere**

**DEBUT**

denumire MOT 25

localitate MOT 20

·

·

**ENTITE 20 secție**

**DEBUT**

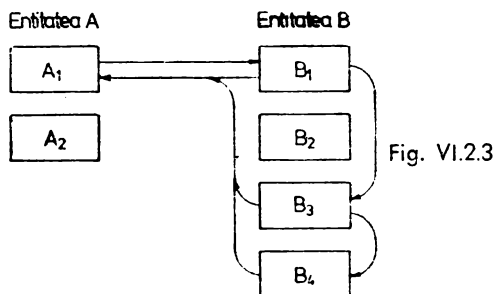
denumire MOT 15

·



### Caracteristica de tip inel

Caracteristica de tip inel este un cap de lanț. Acest cap de lanț, definit ca o caracteristică în entitatea A, leagă pentru oricare din realizările lui A (A1 de exemplu) mai multe realizări ale unei entități B (B1, B3 și B4, de exemplu) care referă, toate, această realizare a lui A (fig. VI.2.3).



Din schemă reies posibilitățile la nivel de cereri :

- prin intermediul inelului din realizarea A1, se poate explora secvențial întreg lanțul de realizări B1, B3 și B4 ;
- prin intermediul referinței din oricare din realizările B1, B3 sau B4, se poate interoga realizarea A1.

*Sintaxa*

Identificator **ANNEAU** [AVEC CHAINE [ {SIMPLE} ] [ {AVANT} ] [ {ARRIERE} ] FIN ]

unde :

- Parametrul CHAINE definește tipul înlănțuirii :
  - opțiunea SIMPLE corespunde unei înlănțuirii similare cu cea prezentată în figura de mai sus ;
  - opțiunea DOUBLE permite în plus și o înlănțuire inversă : A1 punctează către B4, B4 către B3 și B3 către B1. Deci capul de lanț punctează atât prima cât și ultima realizare din lanț, iar în cadrul lanțului, realizările se punctează una pe alta atât înainte cât și înapoi ;
  - opțiunea AVANT permite înlănțuirea realizărilor astfel încât, în limbajul de manipulare, la o parcurgere secvențială a inelului, prima realizare furnizată să fie cea mai veche introdusă ;
  - opțiunea ARRIERE permite înlănțuirea realizărilor astfel încât, la o parcurgere secvențială a inelului, prima realizare furnizată să fie cea mai recent introdusă.
- În absența parametrului CHAINE, valorile implicite sînt SIMPLE și ARRIERE.

**Exemplu**

```

ENTITE 100 întreprindere
  DEBUT
    nume MOT 20
    om-al-muncii ANNEAU
  FIN
ENTITE 100000 persoană
  DEBUT
    nume MOT 30
    loc-de-muncă REFERE om-al-muncii DE UNE întreprindere
  FIN
  
```

### Comentarii

— Prin intermediul cuplului de caracteristici ANNEAU-REFERE din exemplul de mai sus, în limbajul de manipulare se pot obține, pe de o parte, pentru o întreprindere dată, informații despre toate persoanele ce lucrează în acea întreprindere ; pe de altă parte, pentru o persoană dată, informații despre întreprinderea unde lucrează.

— Pentru a utiliza un inel, îi trebuie asociată o referință pe inel și una singură. Totuși, această asociere nu este obligatorie. Câtă vreme referința nu a fost încă definită în structură, capul de lanț există potențial dar nu poate fi utilizat.

— Tipul de înlănțuire (SIMPLE/ DOUBLE sau AVANT/ ARRIERE) nu poate fi modificat în programele de cereri. Dacă se declară CHAINE DOUBLE în structură, sint create înlănțuirii în ambele sensuri, dar sensul principal (cel care furnizează realizările în ordinea „suiwant”) este cel definit de opțiunea AVANT/ ARRIERE.

— Referința asociată la un inel trebuie să fie obligatoriu de același tip cu inelul (CHAINE SIMPLE sau DOUBLE).

— Un inel trebuie totdeauna declarat la primul nivel în entitatea referită (fără a avea, de exemplu, un bloc intercalat între ele).

— Pot fi definite mai multe inele într-o aceeași entitate.

— Pot fi definite un inel și referința sa asociată într-o aceeași entitate.

### Imagine în spațiul virtual

Sînt rezervați 4 octeți dacă inelul este declarat CHAINE SIMPLE și 8 octeți dacă inelul este declarat CHAINE DOUBLE.

### Caracteristica de tip referință cu inel

Este un punctator dublu : atît către realizarea entității referite care conține capul de lanț, cît și către o realizare din lanțul din care face parte. Referința cu inel trebuie obligatoriu asociată la o caracteristică de tip inel definită în entitatea referită.

### Sintaxa

identificator REFERE nume-inel DE  $\left\{ \begin{array}{l} \text{UN} \\ \text{UNE} \end{array} \right\}$  nume-entitate

$\left[ \text{DE } \left\{ \begin{array}{l} \text{UN} \\ \text{UNE} \end{array} \right\} \text{ nume-entitate} \right] \dots$

$\left[ \text{AVEC CHAINE } \left\{ \begin{array}{l} \text{SIMPLE} \\ \text{DOUBLE} \end{array} \right\} \text{ FIN} \right]$

unde :

— Parametrul CHAINE are același sens ca la caracteristicile de tip inel ;

— În absența parametrului CHAINE, implicit este CHAINE SIMPLE.

### Comentarii

— Referința cu inel trebuie să fie de același tip (CHAINE SIMPLE sau DOUBLE) cu inelul asociat.

— Referința trebuie declarată la primul nivel în entitatea pentru care se doresc înlănțuite realizările pe inel (fără a avea, de exemplu, un bloc intercalat).

— Asocierea referinței la un inel asigură gestiunea automată de către procesorul limbajului de manipulare a actualizării inelului, dacă are loc o actualizare sau ștergere a referinței, și reciproc.

*Imagine în spațiul virtual*

Se rezervă 8 octeți pentru referința cu inel CHAINE SIMPLE și 12 octeți pentru referința cu inel CHAINE DOUBLE.

**Caracteristica de tip referință simplă**

Este un punctator simplu, către o realizare a entității referite.

**Sintaxa**

Identificator REFERE  $\left\{ \begin{array}{l} \text{UN} \\ \text{UNE} \end{array} \right\}$  nume-entitate  
 $\left[ \text{DE } \left\{ \begin{array}{l} \text{UN} \\ \text{UNE} \end{array} \right\} \text{ nume-entitate} \right] \dots$

**Exemplu**

```

ENTITE 100 institut
DEBUT
  denumire MOT 30
  localitate MOT 15
FIN
ENTITE 10000 persoană
DEBUT
  nume          MOT 25
  studii        REFERE UN institut
FIN

```

**Comentarii**

— În exemplul de mai sus, caracteristica „studii” permite, pentru o anumită „persoană”, de a afla informații despre institutul pe care l-a terminat. Nu oferă însă nici o posibilitate automată de a afla, pentru un anumit „institut”, informații despre persoanele care l-au urmat.

— Referința simplă trebuie declarată la primul nivel într-o entitate, și nu într-un bloc.

— Procesorul limbajului de manipulare nu oferă o gestiune automată a actualizării referinței simple. De exemplu, dacă o anumită realizare pentru entitatea „institut” este ștearsă, caracteristica „studii” din diversele realizări ale entității „persoană” punctează mai departe acea realizare ștearsă.

*Imagine în spațiul virtual*

Sînt rezervați 4 octeți.

**Caracteristica de tip INVERSE**

Definește un lanț de biți asemănător cu cel de existență al entității inversate.

**Sintaxa**

Identificator INVERSE  $\left\{ \begin{array}{l} \text{TOUT} \\ \text{TOUTE} \end{array} \right\}$  nume-entitate  
 $\left[ \text{DE } \left\{ \begin{array}{l} \text{UN} \\ \text{UNE} \end{array} \right\} \text{ nume-entitate} \right] \dots$

## Exemplu

```

ENTITE 1000 Persoană
DEBUT
  Nume      MOT 20
  An-naștere DE 1900 A 2000
  Profesie  MOT 15
FIN
Inginer INVERSE TOUTE Persoană

```

## Comentarii

- În exemplul de mai sus, caracteristica „Inginer” va fi constituită dintr-un lanț de 1 000 de biți, identic cu al entității „Persoană”. Dacă în programul de cereri, o dată cu înscrierea valorii „inginer” în caracteristica „Profesie” pentru o anumită realizare, se va genera și un bit corespunzător acelei realizări în caracteristica de tip INVERSE, aceasta va permite ulterior o selectare automată numai a persoanelor respective.
- Pot fi declarate oricâte caracteristici de tip INVERSE pentru o aceeași entitate.
- Din sintaxă reiese că se pot inversa și entități incluse. Dar acest lucru este permis numai dacă inversa și entitatea inversată sînt definite la același nivel.
- O caracteristică de tip INVERSE se poate defini în structură numai după entitatea inversată.

## Imagine în spațiul virtual

Se rezervă un contor de realizări existente și un lanț de biți de prezență identice cu cele ale entității inversate.

## Organizarea datelor

### Tipul de organizare

În sistemul SOCRATE, organizarea datelor este o *organizare dispersată*.

### Exemplu

Fie structura :

```

DEBUT
ENTITE 100 uzină
DEBUT
  denumire  MOT 20
  adresă    MOT 30
ENTITE 5000 angajat
DEBUT
  nume     MOT 20
  sex      (2 8) (masculin feminin)
FIN
cod DE 1 A 100000
FIN
FIN

```

Presupunem că avem la dispoziție un spațiu de memorie foarte mare, de 2 la puterea 31 octeți, și că în acest spațiu dorim să rezervăm loc pentru informațiile descrise în structura de mai sus. Îi vom numi *SPAȚIU VIRTUAL*.



Organizarea dispersată se realizează în maniera următoare :

— la începutul memoriei se va găsi prima din cele 100 „uzine“ cu denumirea sa, urmată de adresa sa, urmată de cei 5 000 angajați ai săi, adică de numele și sexul primului angajat, apoi de numele și sexul celui de-al doilea ș.a.m.d. pînă la al 5 000-lea, în sfîrșit urmat de codul uzinei. Pentru toate acestea se vor presupune caracteristicile cu valoarea lor maximală ;

— în același mod urmează rezervarea pentru toate caracteristicile posibile pentru uzina a doua, apoi pentru a treia ș.a.m.d. pînă la a 100-a uzină:

Reprezentarea acestei organizări se poate face schematic astfel :

uzina 1	}	denumire	}	
		adresă		
		angajat 1		{ nume
				{ sex
		angajat 2		{ nume
				{ sex
		angajat 5 000	{ nume	
			{ sex	
		cod		

uzina 2	}	denumire	}	
		adresă		
		angajat 1		{ nume
				{ sex
		angajat-2		{ nume
				{ sex
		angajat 5 000	{ nume	
			{ sex	
		cod		

uzina 100	}	denumire	}	
		adresă		
		angajat 1		{ nume
				{ sex
		angajat 2		{ nume
				{ sex
		angajat 5 000	{ nume	
			{ sex	
		cod		

Este evident că cele 100 uzine pot să nu existe toate în același timp, și că pe de altă parte, ele nu vor avea toate 5 000 angajați. Rezultă că acestui spațiu virtual rezervat foarte mare nu trebuie să-i corespundă un spațiu real de memorare la fel de mare.

Sistemul SOCRATE nu va conserva în spațiul real decât zonele din spațiul virtual care conțin efectiv informații la un moment dat:

### Domenii și etichete virtuale

Din motivele prezentate mai înainte a reieșit că este deosebit de util ca în baza de date, informațiile să fie implantate nu într-unul, ci în mai multe spații virtuale, pe care le-am denumit *domenii*.

Caracteristicile pot fi plasate în unul sau altul dintre domenii prin intermediul *etichetelor virtuale*, astfel :

**\$etic** <listă de caracteristici> **\*etic**

unde „etic“ are 3 caractere și reprezintă numele logic al domeniului.

Reguli de utilizare a etichetelor virtuale :

- \$etic și \*etic trebuie să apară la același nivel în structură ;
- o etichetă virtuală poate fi definită fie la nivelul cel mai înalt în structură, fie imediat în interiorul unei entități de cel mai înalt nivel. Ea nu poate fi definită într-o entitate ierarhic inclusă în altă entitate ;
- caracteristicile unei entități pot fi definite în cel mult 4 domenii diferite : acestea pot fi toate diferite de domeniul ce conține lanțul de biți ;
- orice apariție a unei etichete virtuale în structură antrenează dezimbricarea caracteristicilor asupra cărora se referă în raport cu caracteristicile care o înconjoară (nu se pot totuși depăși 4 nivele de imbricare). În special dacă eticheta virtuală se referă la o singură caracteristică, valorile asociate la această caracteristică sînt aranjate vertical în spațiul virtual.

### Exemple

1. Separarea datelor reale și a datelor de lucru, permițînd creșterea securității de tratament (mai ales dacă datele reale sînt numai în interogare).

```

$FIS
  DEBUT
    ENTITE 100 persoană
    DEBUT
      .
      .
      .
    FIN
  $MAN
    ENTITE 10 lucru
    DEBUT
      .
      .
      .
    FIN
  *MAN
    FIN
  *FIS

```

FIS
persoană 1
persoană 2
.
persoană 100

MAN
lucru 1
lucru 2
.
lucru 10

2. Separarea lanțurilor de biți de prezență, regrupându-le pe un volum separat, pentru a permite optimizarea accesului la ele.

```

$FIS
  DEBUT
    ENTITE 100 persoane
    DEBUT
      $INF
      .
      .
      *INF
    FIN
    ENTITE 10 secție
    DEBUT
      $INF
      .
      .
      *INF
    FIN
  FIN
*FIS

```

FIS
lanț-biți-persoana
lanț-biți secție

INF
persoană 1
.
persoană 100
secție 1
.
secție 10

3. Adăugarea de noi caracteristici într-un nou domeniu, la o entitate definită anterior, fără a schimba informațiile deja încărcate în bază.

```

$FIS
  DEBUT
    ENTITE 100 persoană
    DEBUT
      nume MOT 25
      sex (21) (M F)
    $NOU
      oraș MOT 12

```

```

stradă MOT 20
număr DE 1 A 200
*NOU
. profesie MOT 15
FIN

```

```

FIN
*FIS

```

FIS

lanț-biți persoană	
nume	1
sex	1
profesie	1
.	.
.	.
nume	100
sex	100
profesie	100

NOU

oraș	1
stradă	1
număr	1
.	.
.	.
.	.
oraș	100
stradă	100
număr	100

4. Separarea unei entități de nivelul II pentru a permite o creștere a lucrului secvențial pe această entitate.

```

$FIS
DEBUT
  ENTITE 100 uzină
  DEBUT
    denumire MOT 20
  $SEP
    ENTITE 10 secție
    DEBUT
      cod MOT 8
    FIN
  *SEP
  FIN
FIN
*FIS

```

FIS

lanț-biți uzină	
denumire	1
denumire	2
.	.
.	.
.	.
denumire	100

SEP

lanț-biți secție 1	
cod 1	1
cod 1	2
.	.
.	.
lanț-biți secție 100	
cod 100	1
cod 100	2
.	.
.	.
.	.
cod 100	10

## Metode de acces la informații

### Acces secvențial

Acest acces permite :

- un baleiaj al unei entități în ordinea realizărilor marcate ca prezente în lanțul de biți ;
- un baleiaj al unei entități urmînd lanțul de biți al unei caracteristici de tip INVERSE ;
- o parcurgere a unei entități conform unei înlănțuiri de tip INEL.

### Acces direct prin număr de ordine

Acesta este accesul cel mai rapid. El se realizează prin indicarea numărului de ordine al realizării căutate.

Putem solicita, de exemplu, informații despre al 500-lea angajat de la uzina a 5-a.

### Acces direct prin dicționar

Se realizează declarînd o caracteristică simplă definită în cadrul unei entități drept CHEIE. În acest caz, introducerea unei valori în baza de date este însoțită automat de înscrierea în dicționar a unui cuplu Valoare-cheie/Nr. realizare-entitate. Dicționarul este organizat după metoda arborilor de tip B.

#### Sintaxa

$$\left\{ \begin{array}{l} \text{Caracteristică MOT} \\ \text{Caracteristică numerică} \\ \text{Caracteristică lista de valori} \end{array} \right\} \quad \text{AVEC CLE} \quad \left\{ \left\{ \begin{array}{l} \text{UNIQUE} \\ \text{DOUBLE} \end{array} \right\} \right\}$$

$$\left\{ \left\{ \begin{array}{l} \text{AVANT} \\ \text{ARRIERE} \end{array} \right\} \right\} \left[ \text{CHAINE} \left\{ \left\{ \begin{array}{l} \text{SIMPLE} \\ \text{DOUBLE} \end{array} \right\} \right\} \right] \left\{ \left\{ \begin{array}{l} \text{ORDONNE} \\ \text{NON ORDONNE} \end{array} \right\} \right\} \text{FIN}$$

unde :

- Parametrul UNIQUE/DOUBLE definește natura cheii :
  - UNIQUE definește o cheie de acces „discriminantă”, în care nu pot exista două realizări de entitate cu o aceeași valoare a caracteristicii.
  - DOUBLE definește o cheie de acces „rapidă” la care valorile identice sînt permise.

În absența parametrului, DOUBLE este implicit.

- Parametrul AVANT/ARRIERE definește ordinea de introducere a cheilor în dicționar, în funcție de valoarea lor :
  - AVANT solicită înregistrarea cheilor în dicționar în ordine crescătoare ;
  - ARRIERE solicită înregistrarea cheilor în dicționar în ordine descrescătoare.

În absența parametrului, AVANT este implicit.

- Parametrii CHAINE și ORDONNE sînt comentarii. Au fost păstrați în sintaxă pentru compatibilitatea cu versiunea SOCRATE FELIX C-256.

## Exemple

Cod DE 1 A 17000 AVEC CLE UNIQUE FIN  
 Denumire MOT 10 AVEC CLE ARRIERE FIN  
 Nr-inmatriculare MOT 12 AVEC CLE UNIQUE ARRIERE FIN  
 An-naștere DE 1900 A 2000 AVEC CLE DOUBLE AVANT FIN

### Comentarii

— În exemplele de mai sus, caracteristicile Cod și Nr-inmatriculare sînt declarate chei discriminante. În aceste cazuri, la introducerea unei valori în bază, se realizează verificarea automată ca aceeași valoare să nu mai existe înscrisă pentru o altă realizare.

— În limbajul de manipulare, avînd valoarea unei chei, se pot obține în acces direct informații despre realizarea ce o conține.

— Caracteristicile Denumire și An-naștere din exemple sînt declarate chei rapide. Pentru ele este permisă repetarea valorilor în bază.

— În limbajul de manipulare, avînd valoarea unei chei, se pot obține informații în acces direct, pe rînd, despre toate realizările care conțin această valoare. Realizările cu valori de cheie identice sînt furnizate în ordinea de înregistrare a cheilor în dicționar.

— Atît pentru cheile discriminante cit și pentru cele rapide, în limbajul de manipulare există posibilitatea baleierii tuturor realizărilor entității, în ordinea valorilor cheii. Ordinea de baleiere (crescătoare sau descrescătoare) este dată de parametrul AVANT/ARRIERE din structură și nu poate fi modificată în programul de cereri.

— În exemplele de mai sus, cheile Cod și An-naștere permit acces secvențial în ordine crescătoare, iar cheile Denumire și Nr-inmatriculare în ordine descrescătoare.

— Caracteristicile declarate chei nu sînt autorizate decît imediat în interiorul entităților de nivel 1.

— Caracteristicile de tip MOT cu opțiunea RAD50 nu pot fi declarate chei.

## Formale

### Caracteristica — de tip FORMAL

Permite descrierea structurii arborescente a unei înregistrări de tip COECL.

#### Sintaxa

FORMAL  $\left\{ \left\{ \begin{array}{l} n_i \\ (n_i) \end{array} \right\} \right\}$  Identificator DEBUT <caract-formal> FIN

unde :

—  $n_i$  este factor de repetiție

$$1 \leq n_i \leq 2^{16} - 1$$

Valoarea implicită este 1.

### Exemplu

FORMAL articol  
 DEBUT  
 :  
 :  
 :  
 FORMAL 10 cimp-repetitiv

```

DEBUT
  .
  .
  .
FIN
FIN

```

### Comentarii

- Formalul de nivel cel mai înalt nu poate fi repetitiv.
- Dacă formalul conține direct sau la al n-lea nivel o caracteristică de tip BINAIRE, originea lui va fi cadrată la multiplu de 2. În același caz, dacă formalul este repetitiv ( $n_1 \neq 1$ ), și lungimea lui este rotunjită superior la multiplu de 2.

### Caracteristica de tip MOT formal

Permite definirea unor câmpuri alfanumerice.

#### Sintaxa

Identificator MOT  $\left\{ \begin{array}{l} [n_1] \ n_2 \\ ([n_1] \ n_2) \end{array} \right\}$

unde :

- $n_1$  definește poziția în formal ; folosește la redefiniri de zone (cadraj înapoi) sau la salturi de zone (cadraj înainte).

$$0 \leq n_1 \leq 65\ 535$$

Dacă  $n_1$  nu este indicat, valoarea sa este dată de poziția +1 a ultimului caracter definit anterior.

- $n_2$  definește lungimea câmpului, în octeți.

$$1 \leq n_2 \leq 80$$

#### Exemple

```

nume MOT 20
adresă MOT (15 25)

```

### Valori numerice în formal

Permite definirea unor câmpuri numerice.

#### Sintaxa

Identificator  $\left\{ \begin{array}{l} DILATE \\ PACKE \\ BINAIRE \end{array} \right\} \left\{ \begin{array}{l} [n_1] \ n_2 \ [V \ n_3] \\ ([n_1] \ n_2 \ [V \ n_3]) \end{array} \right\}$

unde :

- *DILATE* — definește câmpuri în format zecimal dilatat.
- *PACKE* — definește câmpuri în format zecimal condensat.
- *BINAIRE* — definește câmpuri în format binar.
- $n_1$  — vezi  $n_1$  de la MOT formal.

–  $n_2$  – este numărul de poziții întregi.

$$0 \leq n_2 \leq 18$$

–  $n_3$  – este numărul de poziții zecimale.

$$0 \leq n_3 \leq 18$$

Valoarea implicită a lui  $n_3$  este 0.

--  $n_2$  și  $n_3$  trebuie să fie :

$$1 \leq (n_2 + n_3) \leq 18$$

#### Lungime ocupată

-- DILATE :  $(n_2 + n_3)$  octeți

-- PACKE :  $\left\lfloor \frac{n_2 + n_3}{2} \right\rfloor + 1$  octeți

-- BINAIRE :  $n_2 + n_3 \leq 4$  : 2 octeți

4 <  $n_2 + n_3 \leq 9$  : 4 octeți

9 <  $n_2 + n_3 \leq 18$  : 8 octeți

#### Exemple

An-naștere	DILATE 4
Preș-unitar.	PACKE (10 5 V 2)
Diametru	BINAIRE 2 V 5

#### Comentarii

- La caracteristicile BINAIRE, originea este cadrată la multiplu de 2.
- La caracteristicile DILATE pentru citirea (de ex. de pe o cartelă) în formal a unui număr zecimal ( $n_3 \neq 0$ ), valoarea respectivă poate avea oricare din următoarele două forme :
  - forma cu virgulă virtuală, utilizată în COBOL ;
  - forma cu punct zecimal explicit, utilizată în FORTRAN. În acest al doilea caz, în declararea dimensiunii câmpului prin „ $n_2$  V  $n_3$ ” trebuie rezervat în formal un octet pentru punctul zecimal și eventual un octet pentru semn.
- Ca și la caracteristica de tip DECIMAL, în opțiunea „V  $n_3$ ”, V este cuvânt-cheie, nu separator al limbajului. În consecință va fi totdeauna încadrat de doi separatori de limbaj (blancuri).

## 3. LIMBAJUL DE MANIPULARE

### Generalități

Limbajul de manipulare permite efectuarea de prelucrări asupra datelor conținute în bază. Principalele tipuri de cereri permise asupra datelor sînt :

- interogare
- actualizare
- ștergere
- generare

Vom numi program de cereri sau mai simplu program, o mulțime structurată de cereri terminată printr-un „?”.



## Apelul procesorului limbajului de manipulare

După lansarea sesiunii SOCRATE, se pot executa programe de cereri de la orice terminal cuplat în sistem (care a deschis o sesiune „HELLO” de lucru sub sistemul de operare).

Un program de cereri se lansează astfel :

>SOC fișier-de-ieșire [opțiuni] = fișier-de-intrare [/CVF]

fișier-de-intrare este specificatorul fișierului sursă ce conține programul de cereri.

fișier-de-ieșire este specificatorul fișierului ce va conține

– listarea programului sursă

– mesaje de eroare

– datele tipărite prin cererile de listare din program.

opțiuni sînt de forma :

/com<sub>1</sub> : val<sub>1</sub>/com<sub>2</sub> : val<sub>2</sub>/...

lista comutatorilor admiși :

DSB (Define Storage Buffer) definește, prin valoarea sa, maximul sumei dimensiunilor bufferelor FCS necesare la un moment dat pentru lucrul cu fișiere externe bazei de date.

DSR (Define Storage Record) definește maximul sumei lungimilor înregistrărilor logice (FORMALE) folosite la un moment dat ca zone articol pentru fișiere externe bazei sau ca zone de memorie de manevră.

SL (Source Listing) permite listarea textului sursă pe fișierul de ieșire. Opțiunea este interesantă pentru valoarea sa negată /NOSL (NO Source Listing).

ME (Macro Expansion) listează textul expandat pentru apelurile de macroinstrucțiuni.

SP (Spool) solicită listarea fișierului de ieșire la LP:, utilizînd *spoolerul*.

CVF (Conventional Format) determină luarea în considerație doar a primelor 60 de caractere dintr-un articol al fișierului de intrare.

Programul sursă este trecut succesiv prin mai multe faze de analiză, în urma cărora se generează un fișier temporar conținînd forma interpretabilă a programului. Ultima fază este constituită de interpretarea structurilor generate.

## Setul de caractere. Separatorii limbajului de manipulare

Setul de caractere este format din :

Litere majuscule : A – Z

Cifre zecimale : 0 – 9

Caractere speciale :

.	(punct)	=	(egal)
,	(virgulă)	<	(mai mic)
:	(punct și virgulă)	>	(mai mare)
:	(două puncte)	\$	(dolar)
(	(paranteză deschisă)	.	(apostrof)
)	(paranteză închisă)	%	(procent)
+	(plus)	?	(semnul întrebării)
-	(linie sau minus)		(bară oblică)

*	(asterisc)		(spațiu)
/	(slash)	<TAB>	(tab)
"	(ghilimele)	↑	(săgeată-verticală)
—	(subliniere)		

Separatorii limbajului de manipulare

(spațiu)	)
=	(
↑=	+
>	*
<	/

### Cuvinte cheie

Un *cuvânt cheie* este un element dintr-o cerere, care poate fi folosit numai după cum arată sintaxa cererii. În acest manual cuvintele cheie apar scrise cu majuscule, în descrierile sintactice, ca și în exemplele de programe. Este interzisă folosirea cuvintelor cheie ca identificatori. Anexa B conține lista cuvintelor cheie.

### Identificatori

Un *identificator* este o succesiune de maximum 30 de caractere (literă, cifra, —) primul caracter fiind obligatoriu literă. Printr-un identificator ne vor referi doar la o caracteristică descrisă în prealabil în structura bazei de date, prin limbajul de descriere a structurii SOCRATE.

### Constante

- constantă Z (alfanumerică) are forma : 'șir de-caractere'.
- constantă Y (numerică întregă) are forma :

$$\left\{ \begin{array}{l} + \\ - \end{array} \right\} c_1 c_2 \dots c_n \quad \text{unde } c_1 \text{ este o cifră zecimală și } c_1 \neq 0.$$

- constantă W (numerică fracționară) are forma :

$$\left\{ \begin{array}{l} + \\ - \end{array} \right\} \text{ parte-întreagă } \bullet \text{ parte-zecimală} \quad \text{unde :}$$

parte-întreagă este o constantă Y, iar

parte-zecimală este o succesiune de cifre zecimale.

Numărul de cifre dintr-o constantă numerică nu poate depăși valoarea 18.

Numărul de caractere dintr-o constantă Z nu poate depăși valoarea 30.

### Variabile de lucru

Utilizatorul are la dispoziție un număr de *variabile de lucru*, valabile pe durata unui apel al procesorului limbajului de manipulare. Acestea sînt de patru tipuri :

— *Variabile alfanumerice* Z<sub>1</sub> (Z<sub>1</sub> la Z<sub>10</sub>). Pot conține șiruri de maximum 30 caractere alfanumerice.

Suportă următoarele tipuri de utilizări :

- modificare ;
- interogare ;
- test.
- Variabile numerice întregi  $Y_i$  ( $Y_1$  la  $Y_{30}$ ). Pot conține valori numerice întregi cu maximum 18 cifre.
- Variabile numerice zecimale  $W_i$  ( $W_1$  la  $W_{20}$ ). Pot conține valori numerice fracționare în care suma numărului de cifre întregi și zecimale nu poate depăși 18. Numărul implicit de zecimale al variabilelor  $W_i$  este 3 ; acest număr poate fi modificat utilizând cererea D.

Variabilele  $Y_i$  și  $W_i$  suportă următoarele tipuri de utilizări :

- modificare ;
- interogare ;
- test ;
- calcul numeric.
- Variabile de adresă  $X_i$  ( $X_1$  la  $X_{15}$ ). Pot fi utilizate pentru a puncta adresa virtuală a unei realizări de entitate, fie direct, fie utilizând mecanismul *inel-referință* sau *INVERSE*.

Suportă următoarele tipuri de utilizări :

- definire ;
- modificare ;
- interogare ;
- test.

Observație : Există și o variabilă  $X_0$ , prin care sistemul SOCRATE punctează totdeauna adresa de debut a bazei de date, pe nivel 0, superior tuturor caracteristicilor. Variabila  $X_0$  nu suportă nici unul din tipurile de utilizări prezentate mai sus.

### Structura unui program scris în limbajul de manipulare

Un program în limbajul de manipulare este o succesiune de cereri simple sau structurate, terminate prin „?”.

Disponerea cererilor în articolele fișierului sursă este liberă. O cerere poate ocupa mai multe linii de program ; putem de asemenea scrie mai multe cereri pe o linie. Nu putem despărți însă un atom lexical (identificator, cuvînt cheie, constanta) pe două linii de program.

Se pot insera în program linii de comentarii. O linie de comentariu începe prin gruparea /\* și va conține la sfîrșit delimitatorul simetric \*/. Cei doi delimitatori /\* și \*/ trebuie să încadreze fiecare linie comentariu în întregime. Comentariile nu se pot insera într-o linie de program ce conține și cereri.

### [Macrogeneratorul

Macrogeneratorul este un instrument care permite dezvoltarea unui limbaj simplu, specific unei anumite aplicații pornind de la limbajul de manipulare. Folosirea noului limbaj introdus este simplă și comodă și nu presupune cunoașterea sintaxei limbajului de manipulare SOCRATE și nici a structurilor de date folosite.

Utilizatorul își poate defini un șir de caractere ca nume al macroinstrucțiunii și îi asociază acestuia o secvență de cereri SOCRATE, numită corpul macroinstrucțiunii.

În definirea unei macroinstrucțiuni este permisă folosirea parametrilor ca și a separatorilor de apel.

### Apelul de macroinstrucțiune

Apelul de macroinstrucțiune se referă la modul de folosire al acesteia și se face prin specificarea numelui ei, urmat eventual de parametrii actuali de apel.

Un apel de macroinstrucțiune poate apare într-o frază a limbajului de manipulare sau într-un alt apel de macroinstrucțiune.

Principiile de tratare ale apelului de macroinstrucțiune, adică ale expansiunii ei sînt :

- numele macroinstrucțiunii este înlocuit prin corpul ei ;
- parametrii formali sînt înlocuiți prin parametrii actuali ;
- dacă parametrul actual nu este urmat de separator de apel, el este considerat ca fiind șirul de caractere pînă la primul separator al limbajului de manipulare ;
- dacă parametrul actual este urmat de separator de apel, atunci el este considerat ca fiind șirul de caractere pînă la separatorul de apel, acesta din urmă fiind încadrat între doi separatori ai limbajului de manipulare ;
- în cazul în care un parametru actual este la rîndul său un apel de macroinstrucțiune, eventualii parametri și separatori ce urmează sînt considerați ca aparținînd noii macroinstrucțiuni ;
- în momentul expansiunii unei macroinstrucțiuni are loc verificarea separatorilor de apel cu cei definiți.

## Concepte SOCRATE

### Tipuri de date

Limbajul de manipulare SOCRATE lucrează cu următoarele tipuri de date :  
*caracteristică simplă* din baza de date :

- *mot*
- *listă-de-valori*
- *caracteristică-numerică* (întreagă sau zecimală)

*caracteristici-compuse*

- *realizare de entitate*
- *referință*
- *bloc*

*mulțimi de caracteristici compuse* :

- *entitate*
- *inel*
- *invers*

*caracteristici-formal :*

- *caracteristica-formal simplă*
- *caracteristica-formal compusă*

*variabile simple :*

- *alfanumerice*  $Z_i$ ,  $i = 1, 2, \dots, 10$
- *numerice întregi*  $Y_i$ ,  $i = 1, 2, \dots, 30$
- *numerice zecimale*  $W_i$ ,  $i = 1, 2, \dots, 20$
- *adresa*  $X_i$ ,  $i = 1, 2, \dots, 15$

*zone articol pentru fișiere, BUFFER*  $n$ ,  $n = 1, 2, \dots, 10$

*fișiere logice BANDE*  $n$ ,  $n = 1, 2, \dots, 10$

### Citare-complexă. Citare-formal

Vom folosi noțiunea de *citare-complexă* pentru a ne referi la :

- *caracteristici simple*
- *caracteristici compuse*
- *mulțimi de caracteristici compuse,*

iar noțiunea de *citare-formal* pentru

- *caracteristici formal*

atunci când utilizăm aceste tipuri de date în cadrul cererilor (instrucțiunilor) SOCRATE.

Prin *citare*, utilizatorul specifică, pe lângă identificatorul unui obiect dintr-un anumit tip de date, și alte informații :

- *calificarea* obiectului (ca în COBOL) pentru ridicarea ambiguității relative la structură ;
- *cuantificarea* obiectului, dacă este de tip mulțime, prin UN(E) sau TOUT(E) pentru a preciza dacă interesează unul sau mai mulți reprezentanți ai acestei mulțimi ;
- *selecția*
  - prin *filtre* — permițind alegerea unor elemente dintr-o mulțime eventual ordonată ;
  - prin *nr. realizare* — permițind accesul la un singur element dintr-o mulțime de tip entitate ;
- *asocierea* unei variabile  $X_i$ , ca *adjectiv*, pe lângă obiectul citat (entitate, inel, invers, referință) pentru a putea defini ulterior, prin calificare cu  $X_i$ , contextul formal al acelui obiect.

*Sintaxa citarii-complexe*

[cuantificator] identificator [adjectiv X] [nr-de-realizare]  
[filtru-rapid] [filtru-lent] [calificare]

în care :

<i>cuantificator</i>	poate fi UN, UNE, TOUT, TOUTE
<i>adjectiv X</i>	este o variabilă $X_i$
<i>nr-de-realizare</i>	poate fi o constantă numerică întreagă sau o variabilă $Y_i$ ,
<i>filtru-rapid</i>	poate avea una din formele : AVEC condiție ; PAR identificator ;
<i>filtru-lent</i>	are forma : AYANT condiție ;
<i>calificare</i>	are forma : DE citare-complexă... [DE $X_i$ ]

### Comentarii

O variabilă  $X_1$  nu poate juca rol de *adjectiv* decât pentru o caracteristică de tip entitate, *inel*, *invers* sau *referință*.

Cuantificatorii sînt obligatorii pentru caracteristici de tip mulțime (*entitate*, *inel*, *invers*) și interziși pentru celelalte tipuri.

Nu se pot folosi simultan selecția prin filtru cu selecția prin număr de realizare.

Selecția prin număr de realizare se poate face doar pentru o caracteristică de tip entitate.

Calificarea se poate face doar cu :

— o caracteristică de tip *entitate*, *inel*, *invers*, *referință*, *bloc* ;

— o variabilă  $X_1$  ;

— cuvîntul cheie  $X_0$  ; calificarea prin  $X_0$  fixează contextul la *nivel fișier*, adică primul nivel din descrierea bazei de date.

Calificarea prin variabila  $X_1$  poate avea două forme :

— calificare *postfixată* : identificator DE  $X_1$

— calificare *prefixată* :  $X_1$ . identificator

Calificarea prefixată este valabilă numai pentru caracteristicile simple.

Trebuie specificăți toți calificatorii, în ordine ierarhică, chiar dacă identificatorul calificat este unic.

### Exemple de citare complexă

- |  |   |
|--|---|
| 1. UN angajat  | prima realizare generată a entității angajat  |
| 2. UN angajat 1  | realizarea numărul 1 a entității angajat.   |
| 3. UN angajat X1 Y1  | realizarea entității angajat, cu numărul de realizare egal cu valoarea din Y1 și poziționarea variabilei X1 pe această realizare.   |
| 4. UN angajat AVEC nume = 'POPESCU' ;  | realizarea entității angajat ce conține în caracteristica-simplă nume, valoarea 'POPESCU'.  |
| 5. UN angajat AYANT nume = 'POPESCU' ;   | identifică același obiect din bază ca și 4 dar prin metode de acces diferite.   |
| 6. TOUT copil DE X1  | toate realizările generate ale entității copil aparținînd realizării din entitatea angajat, pe care punctează X1.   |
| 7. TOUT angajat PAR nume ;   | toate realizările generate ale entității angajat în ordinea crescătoare a valorilor din caracteristica-simplă nume.   |
| 8. TOUT angajat PAR nume ;<br>AYANT funcție = 'inginer' ;                                  | din realizările de la 7 sînt selectate doar cele care conțin în caracteristica-simplă funcție, valoarea 'inginer'.  |
| 9. UN atelier X1 5 DE UNE<br>secție 2  | realizarea 5 a entității atelier din realizarea 2 a entității secție și poziționarea variabilei X1.   |
| 10. nume DE TOUT personal-atelier<br>DE X1   | caracteristica nume din toate realizările entității angajat ce fac parte din inelul constituit în realizarea entității atelier punctată de X1.  |
| 11. TOUT copil DE șef-secție<br>DE UNE secție  | toate realizările generate ale entității copil din cadrul unei realizări din entitatea angajat pe care punctează caracteristica referință șef-secție din prima realizare generată a entității secție  |
| 12. nume DE șef-atelier DE<br>apartenența-atelier DE<br>UN angajat AVEC nume = 'IONESCU' ; | caracteristica-simplă nume din realizarea entității angajat spre care punctează caracteristica șef-atelier din realizarea entității atelier spre care punctează referința apartenența-atelier din cadrul primei realizări a entității angajat ce conține valoarea 'IONESCU' în caracteristica nume. |
| 13. funcție DE TOUT tesa<br>AYANT stare-politică = 'UTC'<br>ET sex = 'f' ;                 | caracteristica-simplă funcție din toate realizările entității angajat ce fac parte din inversul tesa și care conțin valorile 'UTC' și respectiv 'F' în caracteristicile stare-politică și sex.  |

14. TOUT angajat AYANT  
an-de-naștere DE UN copil  
>= 1983 ;

toate realizările entității angajat ce conțin realizări ale entității copil având în caracteristica simplă an-de-naștere o valoare mai mare sau egală cu 1983.

*Citare-formal*

*Sintaxa citării-formal :*

FORMAL identificator  $\left\{ \begin{array}{l} \text{DANS BUFFER } n_1 \\ [n_2] \text{ DE } X_1 \end{array} \right\}$

identificator DE  $X_1$

—  $n_1$  este numărul unui bufer ( $0 \leq n_1 \leq 10$ )

Obs. : BUFFER 0 specifică zona articol a fișierului de ieșire.

—  $n_2$  este numărul formalului repetitiv care se suprapune peste zona de memorie adresată prin variabila  $X_1$ .

*Comentarii*

Deosebim în descrierea sintactică 3 forme posibile :

(1) identificator **DANS BUFFER**  $n_1$

(2) identificator  $n_1$  **DE**  $X_1$

(3) identificator **DE**  $X_1$ .

(1) Rezervă în memoria centrală o zonă BUFFER  $n_1$  (dacă nu a fost deja rezervată și aplică peste această zonă, *formalul* de nivel 1 descris în structura bazei de date.

(2) Permite suprapunerea unui *formal repetitiv* de nivel mai mare sau egal ca 2 peste o zonă de memorie adresabilă deja prin  $X_1$ .

Formele (1) și (2) se pot folosi doar ca membru drept într-o atribuire de tip D sau M pentru o variabilă  $X_1$ .

Forma (1) aplică un *formal* inițial peste zona bufer iar forma (2) detaliază prin *indexare* această descriere, pe mai multe nivele.

Pentru cimpurile ultimului nivel se folosește forma (3), aceasta putînd apare în cadrul cererilor ca și *citarea-complexă* pentru caracteristicile simple (*mot*, *listă-de-valori*, *numeric*). Ca și la *citarea-complexă*, forma (3) poate avea și varianta de calificare prefixată :

$X_1$ . identificator

**Exemple**

1' **FORMAL f-angajat DANS BUFFER 3**

Identifică o înregistrare logică de nivel 1 f-angajat, peste o zonă de memorie BUFFER 3 de lungime mai mare sau egală cu lungimea înregistrării logice.

2 **FORMAL f-copil 3 DE X1**

Identifică a treia înregistrare logică repetitivă de nivel > 1 peste o zonă de memorie a cărei adresă de început se află în  $X_1$ .

3 **an-de-naștere DE X1**

Identifică un cîmp dintr-o înregistrare logică suprapusă peste o zonă de memorie adresabilă prin  $X_1$ .

**Noțiunea de condiție**

O *condiție* (sau *condiție-compusă*) se întâlnește într-un filtru sau într-o cerere condițională (SI). Ea este formată dintr-o succesiune de *condiții simple sau de existență*, independente unele de altele, legate doar prin operatorii logici ET sau OU. Unei condiții i se asociază valoarea logică adevărat, fals sau nedefinit.

*Sintaxa unei condiții*

condiție [ET condiție...] [[OU condiție [ET condiție...]]...]

în care : condiție este condiție-de-existență sau condiție-simplă.

**Condiție simplă**

Sintaxa unei condiții-simple :

$$\begin{array}{ccc}
 \left. \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ Z_1 \end{array} \right\} \text{op-relațional} & & \left. \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ Z_1 \\ \text{constantă} - Z \\ U \end{array} \right\} \\
 \\
 \left. \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ Y_1 \\ W_1 \end{array} \right\} \text{op-relațional} & & \left. \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ Y_1 \\ W_1 \\ \text{constantă-numerică} \\ U \end{array} \right\} \\
 \\
 X_i & \left\{ \begin{array}{l} = \\ \uparrow = \end{array} \right\} & \left\{ \begin{array}{l} X_i \\ U \end{array} \right\}
 \end{array}$$

în care op-relațional poate fi unul din următorii operatori :

$$\begin{array}{llll}
 = & \text{(egal)} & >= & \text{(mai mare sau egal)} \\
 \uparrow = & \text{(diferit)} & < & \text{(mai mic)} \\
 > & \text{(mai mare)} & <= & \text{(mai mic sau egal)} \\
 == & \text{(egal parțial)} & & 
 \end{array}$$

Operatorul == se poate utiliza doar pentru operanzi alfanumerici, avînd lungimea operandului 2 mai mică sau egală cu cea a operandului 1. Condiția  $\text{opd}_1 == \text{opd}_2$ , în care  $\text{opd}_1$  desemnează un șir alfanumeric de lungimea  $l_1$  iar  $\text{opd}_2$  un șir de lungime  $l_2$  ia valoarea adevărat numai dacă  $\text{opd}_1$  conține în primele  $l_2$  poziții  $\text{opd}_2$ .

*Citare-complexă* și *citare-formal* desemnează caracteristici simple (*mot*, *listă de valori*, *numeric*) și trebuie să fie compatibile că tip cu celălalt operand.

Caracteristicile de tip mulțime (*entitate*, *inel*, *invers*) nu pot face obiectul unei condiții simple.

La comparația cu U (nedefinit) singurii operatori relaționali admiși sînt egal și diferit.

Dacă vreunul din operanzi are valoarea nedefinit (exceptînd cazul cînd operandul 2 este simbolul U), condiția ia valoarea nedefinit.

Dacă unul din operanzi poziționează, la evaluarea sa, *variabila ERREUR*, condiția ia valoarea nedefinit.

**Exemple**

1     Z1 = funcție DE UN angajat



la valoarea adevărat sau fals după cum conținutul lui  $Z_1$  este sau nu egal cu cel din caracteristica-simplă funcție din prima realizare a entității angajat și valoarea nedefinită dacă unul din operanzi este nedefinit.

2    **marca DE X1 ↑ = U**

la valoarea adevărat sau fals după cum caracteristica marcă are sau nu o valoare diferită de nedefinit.

### Condiția folosită într-un filtru rapid

$$\text{AVEC identificator} \left\{ \begin{array}{l} = : \\ > \\ > = \\ < \\ < = \\ = = \end{array} \right\} \left\{ \begin{array}{l} \text{constanta} \\ Z_1 \\ Y_1 \\ W_1 \end{array} \right\} ;$$

în care :

— identificator desemnează numele unei *caracteristici simple*, declarate *cheie*.

Operandul drept trebuie să fie de un tip corespunzător cu cel al caracteristicii rapide.....

Caracteristica rapidă va fi numită *discriminantă* dacă *cheia* este declarată unică, caz în care caracteristica va lua o valoare diferită pentru fiecare realizare a entității din care face parte.

Ansamblul de valori luate de o astfel de caracteristică este reținut într-un *dicționar* care va asocia la o valoare a *cheii* un număr de realizare a *entității*.

Caracterul „;” nu este separator al limbajului de manipulare. În consecință, va fi încadrat de doi separatori (blăncuri).

### Exemple

1    **UN angajat AVEC marca = Y1 ;**

Se caută în dicționar o cheie corespunzătoare caracteristicii marca cu valoarea din Y1 și dacă este găsită se furnizează numărul de realizare pentru entitatea angajat.

2;    **TOUT angajat AVEC nume == 'IONESCU' ;**

Sînt selectate din dicționar toate realizările entității angajat avînd primele caractere din cheia nume identice cu șirul 'IONESCU'.

### Criteriul de sortare

**PAR** identificator ;

Cînd se folosește un *filtru lent*, se baleiază toate realizările entității și se evaluează filtrul pentru fiecare din ele. Ordinea în care se face baleierea este aceea a numerelor de realizare crescătoare.

Se poate baleia toată mulțimea sau o parte a sa după o ordine impusă de valorile anumitor caracteristici numite caracteristici de sortare.

O astfel de caracteristică trebuie să fi fost declarată *rapidă* sau *discriminantă* (CLE sau CLE UNIQUE) în descrierea de structură.

Observația de la AVEC referitoare la „;” rămîne valabilă.

### Condiția folosită într-un filtru secvențial (lent)

Prin *filtrul lent* se efectuează o selecție a realizărilor unei entități, *inel* sau *invers* și anume acelea în contextul cărora condiția ia valoarea adevărat.

**AYANT** condiție ;

Caracteristica de tip mulțime ce se dorește filtrată definește un context de calificare implicit pentru :

- operanzii din partea stângă a condițiilor simple
- caracteristicile din condițiile de existență.

Ruperea contextului se poate face prin calificare cu variabila  $X_1$  sau  $X_0$ .

Observația de la AVEC referitoare la „;” rămâne valabilă.

#### Exemplu

TOUT angajat AYANT stare-politică = 'UTC' ET stare-civilă = 'căsătorit'  
ET EXISTE UN copil OU stare-politică = 'PCR' ET stare-socială = 'căsătorit' ;

Va selecta toți angajații membrii UTC căsătoriți și cu copii sau membrii PCR și căsătoriți.

Cuantificatorul UN ce apare în exteriorul unui *filtru* permite desemnarea primei realizări generate dintr-o mulțime, ce răspunde filtrului

Apariția însă a cuantificatorului UN în interiorul unui filtru va conduce la selecția primei realizări din mulțime pentru care condiția din filtru ia valoarea adevărat.

#### Exemplu

I UN copil DE UN angajat

Va tipări prima realizare a entității copil din prima realizare a entității angajat.

I UN angajat AYANT an-de-naștere DE UN copil = 1984 ;

Cuantificatorul UN al entității copil va desemna primul copil din mulțime pentru care an-de-naștere are valoarea 1984, dacă există cumva o astfel de realizare.

### Condiția de existență

Sintaxa *condiției de existență* :

{ EXISTE }  
{ PAS } citare-complexă

în care *citare-complexă* poate desemna o caracteristică de tip :

*entitate, inel, invers* — caz în care *condiția de existență* introdusă prin

EXISTE va avea valoarea adevărat dacă există ce puțin o realizare a mulțimii desemnate prin citare și fals în caz contrar ;

PAS va avea valoarea fals dacă există cel puțin o realizare a mulțimii desemnate prin citare și adevărat în caz contrar.

*mot, listă-de-valori, numeric, referință*

caz în care *condiția de existență* introdusă prin

EXISTE ia valoarea adevărat dacă caracteristica citată are în bază o valoare diferită de nedefinit și va avea valoarea fals în caz contrar ;

PAS ia valoarea fals dacă caracteristica citată are o valoare diferită de nedefinit și valoarea adevărat în caz contrar.

Condiția de existență pentru o caracteristică de tip *mot*, *listă-de-valori*, *numeric* poate fi înlocuită printr-o condiție simplă având în membrul drept constanta U (nedefinit). Tot așa se va testa de existență sau nu o variabilă  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $X_1$ , prin comparare cu U.

### Exemple

#### 1 EXISTE UN angajat Y1

la valoarea adevărat dacă a fost generată realizarea cu numărul din Y1 a entității angajat și fals în caz contrar.

#### 2 EXISTE apartenența-atelier DE UN angajat

la valoarea adevărat dacă a fost creată referința apartenența-atelier din prima realizare angajat și fals în caz contrar.

#### 3 PAS UN pers-atelier AYANT nume = 'IONESCU' ; DE UN atelier 2 DE UNE secție 3:

la valoarea adevărat dacă nu există nici un angajat cu numele IONESCU în atelierul 2 din secția 3 și fals în caz contrar.

### Logica de evaluare a condițiilor

În logica clasică (booleană), rezultatul unei condiții poate fi: adevărat sau fals.

Existența valorilor *nedefinit* și a mulțimilor vide, într-o bază de date, ne impune o a treia posibilitate: valoarea unei condiții poate fi *nedefinit*. Vom utiliza deci o logică cu trei valori:

adevărat (A) fals (F) nedefinit (U).

Logica operației „ET” (SI)

ET	U	A	F
U	U	U	F
A	U	A	F
F	F	F	F

Logica operației „OU” (SAU)

OU	U	A	F
U	U	A	U
A	A	A	A
F	U	A	F

Se observă *comutativitatea* fiecărei operații logice în raport cu cei doi operanzi. Logica booleană este inclusă în logica cu trei valori.

O condiție va fi evaluată de la stînga la dreapta, ținînd cont de prioritatea lui ET în raport cu OU.

## Cereri SOCRATE

Acest capitol descrie toate cererile SOCRATE.

### Cerere APPEL

*Funcțiune* Cererea APPEL lansează în execuție un subprogram (task-fiu) scris în alt limbaj, sau interpreterul liniilor de comandă.

*Sintaxa*

$$\text{APPEL nume-subprogram} \left[ \text{AVEC} \left\{ \begin{array}{l} \text{BUFFER } n \\ \text{constanta } -Z \\ Z_i \end{array} \right\} \right]$$

— *nume-subprogram* este numele subprogramului în alt limbaj atribuit la editarea de legături, avînd max. 6 caractere RADIX-50.

— BUFFER *n*, constantă-*Z* sau *Z<sub>i</sub>* sînt parametrii de apel.

### Cerere ATTACHE

*Funcțiune* Cererea ATTACHE afectează un fișier secvențial fizic la un fișier-logic BANDE *i*.

*Sintaxa*

$$\text{ATTACHE BANDE } n \text{ A } \left\{ \begin{array}{l} \text{'specificator-fișier'} \\ Z_i \end{array} \right\} \text{ AVEC } \left\{ \begin{array}{l} \text{'prm, bfs, rcs, rcf'} \\ Z_j \end{array} \right\}$$

— *n* este numărul fișierului-logic ( $1 \leq n \leq 10$ )

— *specificator-fișier* este specificatorul fișierului fizic

— *bfs* definește dimensiunea buferului (înregistrare fizică)

— *rcs* definește lungimea articolului (înregistrare logică)

— *prm* definește modul de prelucrare (I pt. input, O pt. output)

— *rcf* definește formatul articolului (F pt. fix, V pt. variabil).

*Comentarii*

— Cererea ATTACHE echivalează cu descrierea de fișier și instrucțiunea OPEN din COBOL.

— Alocarea spațiului pentru buffer și zona articol se face cu cererea ATTACHE iar dealocarea lui, cu cererea DETACHE.

— După cererea ATTACHE se pot lansa cereri I/E (LIRE, ECRIRE) pe fișierul logic BANDE *n*.

*Exemple*

1 ATTACHE BANDE 1 A 'CR:' AVEC 'I,80,80,F'

Afectează lectorul de cartele perifericului logic BANDE 1.

2 M Z1 = 'F.LST'  
M Z2 = 'O,512,132,V'  
ATTACHE BANDE 3 A Z1 AVEC Z2

Afectează fișierul SY :F.LST în scriere cu format variabil perifericului logic BANDE 3.

## Cerere BLOQUER

*Funcțiune* Cererea BLOQUER interzice accesul altor utilizatori la baza de date.

*Sintaxa*

### BLOQUER

*Comentarii*

— Cererea este utilă atunci când se dorește ca, pe durata actualizării bazei de date dintr-un program de cereri, alți utilizatori să nu aibă acces la informații.

În intervalul de timp în care baza este *blocată*, toate programele care solicită accesul la bază sînt puse în așteptare.

— Efectul cererii BLOQUER este anulat printr-o cerere LIBERER.

— În intervalul de timp în care baza este blocată, fiind eliminat mecanismul de sincronizare multiacces, crește viteza de acces la baza de date.

— Nu se recomandă utilizarea instrucțiunilor

LIRE și  
M citare = EXT

între instrucțiunile BLOQUER și LIBERER, atunci când se lucrează în mod concurrent (mai multe procesoare SOC active simultan).

— Pentru instrucțiunile G, S, SE și M referință nu este necesară protejarea lor explicită prin mecanismul BLOQUER-LIBERER, deoarece ele sînt protejate în mod implicit de către procesorul limbajului de manipulare.

## Cerere CKPT

*Funcțiune* Cererea CKPT declanșează crearea unui *punct de reluare* dacă au fost îndeplinite condițiile stabilite la deschiderea sesiunii cu privire la punctele de reluare.

*Sintaxa*

### CKPT [Y<sub>i</sub>]

— variabila Y<sub>i</sub> va primi după execuția cererii CKPT valoarea numărului punctului de reluare dacă programul se execută într-o sesiune cu fișier de securitate și va rămîne nemodificată în caz contrar.

*Comentarii*

— Cererea CKPT trebuie să succedă setului de cereri care actualizează baza de date. Ea declanșează procesul de creare a punctului de reluare dacă condițiile erau îndeplinite în momentul atingerii de către program a cererii PAUSE ce a precedat-o.

— Se recomandă folosirea mai multor perechi de cereri PAUSE CKPT care să includă între ele diversele seturi de cereri de modificare a bazei de date, în locul folosirii unui singur set PAUSE CKPT pentru tot programul (mai ales pentru programele mari).

— Nu se permite imbricarea unei perechi de cereri PAUSE CKPT în interiorul altei perechi PAUSE CKPT.

— Nu se recomandă utilizarea cererilor :

LIRE  
M citare = EXT

în interiorul unei perechi de cereri PAUSE CKPT, atunci când se lucrează în mod concurrent (mai multe procesoare SOC active simultan).

— Tehnica reluării programului din punctul în care s-a întrerupt datorită apariției unui incident rămîne la latitudinea programatorului care cu ajutorul limbajului de manipulare

și poate construi algoritmul adecvat aplicației sale pentru reluare, după aducerea bazei de date în punctul de reluare prin intermediul utilitarului RSS.

-- O instrucțiune CKPT, care nu este precedată în secvența de execuție de către o cerere PAUSE, devine inefectivă.

## Cerere D

*Funcțiune* Inițializează variabile  $X_i$  sau stabilește numărul de zecimale al variabilelor  $W_i$

*Sintaxa*

D  $X_i = \left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \end{array} \right\}$

D  $W_i = n$

-- *citare-complexă* se referă la o caracteristică de tip : entitate, inel, invers sau referință ;

-- *citare-formal* referă o caracteristică formal de nivel 1 sau repetitivă.

-- n este un număr întreg pozitiv, cuprins între 0 și 18, care reprezintă numărul de zecimale dorit pentru variabila  $W_i$ .

*Comentarii*

-- Prima formă are efect numai în faza de analiză semantică, pentru poziționarea variabilei  $X_i$  în vederea calificării prin ea.

-- Cererea D în prima formă este utilă în subprograme SOCRATE pentru a permite calificarea printr-o variabilă  $X_i$  globală, poziționată în programul apelant.

-- Prima formă este inefectivă în faza de execuție a programului.

-- Reamintim că valoarea implicită a numărului de zecimale pentru o variabilă  $W_i$  este 3.

## Exemplu

```
Program apelat
M X1 = UN angajat AVEC nume == 'POPESCU' ;
EXEC subpro
Program apelat SUBPRO
:DEFPRO subpro
:EXP
D X1 = UN angajat
POUR X1
I nume I funcție
FIN
.....
:FDEF?
```

## Cerere DETACHE

*Funcțiune* Închide fișierul referit prin fișierul-logic BANDE n și eliberează zona bufer și articol a acestuia.

*Sintaxa*

DETACHE BANDE n

-- n este numărul fișierului-logic ( $n < 10$ ).

## Comentariu

- Cererea DETACHE este echivalenta Instrucțiunii CLOSE din COBOL.
- Cererea DETACHE trebuie să fie precedată de o instrucțiune ATTACHE.
- După o cerere DETACHE nu se mai pot efectua operații I/E cu fișierul logic BANDE n, decât dacă se lansează o nouă cerere ATTACHE.
- Cererea DETACHE eliberează zona articol ca și zona bufer rezervate pentru fișierul logic BANDE n.

## Cerere ECRIRE

**Funcțiune** Scrie un articol într-un fișier-logic BANDE n sau în fișierul de ieșire prevăzut la comanda de lansare a procesorului SOCRATE.

## Sintaxa

ECRIRE { BUFFER  $n_1$  DANS BANDE  $n_2$  }  
[  $n_3$  ]

- $n_1$  este numărul unui bufer din memoria centrală ( $0 \leq n_1 \leq 10$ )
- $n_2$  este numărul unui fișier-logic ( $1 \leq n_2 \leq 10$ )
- $n_3$  definește caracterul de salt al scrierii.

## Comentariu

— Deosebit două forme sintactice :

(1) ECRIRE [  $n_3$  ]

(2) ECRIRE BUFFER  $n_1$  DANS BANDE  $n_2$

— Forma (1) se folosește pentru scrierea unui articol pe fișierul de ieșire, declarat la lansarea procesorului SOCRATE :

>SOC fișier-de-ieșire [opțiunii] = fișier-de-intrare

— Pentru a scrie un articol în fișierul-de-ieșire trebuie să lansăm în prealabil cel puțin o cerere de tipărire I cu format.

— Cererile I cu format completează buferul implicit atașat fișierului-de-ieșire (BUFFER 0), care prin cererea ECRIRE este trecut efectiv în fișierul-de-ieșire.

— După o cerere ECRIRE buferul fișierului-de-ieșire este inițializat cu blancuri.

$n_3$  — definește caracterul de salt al scrierii și poate lua valori cuprinse în intervalele 0-7 sau 10-17 având următoarea semnificație

- 0 — salt la pagină nouă ;
- 1, 2, 3, 4, 5, 6 — scriere după  $n_3$  rânduri ;
- 7 — supraimprimare.

Valorile cuprinse în intervalul 10-17 au aceeași semnificație cu cele corespunzătoare din intervalul 0-7 indicând, în plus, păstrarea cursorului (fără <CR>).

- 10 — salt la pagină nouă fără <CR> ;
- 11, 12, 13, 14, 15, 16 — scriere după ( $n_3-10$ ) rânduri fără <CR>;
- 17 — supraimprimare fără <CR>.

Valoarea implicită a lui  $n_3$  este 1.

Dacă se dorește controlul ecranului prin scrierea unor caractere de control proprii introduse, de exemplu, într-o caracteristică se indică utilizarea formei

I(1) caracteristică ECRIRE 17

pentru a elimina caracterele <CR> și <LF> introduse de procesor.

— Forma (2) se folosește pentru scrierea unei zone articol BUFFER  $n_1$  într-un fișier fizic BANDE  $n_2$  ce a făcut obiectul unei cereri ATTACHE anterioare.

— BUFFER 0 semnifică zona articol a fișierului de ieșire.

— Zona articol BUFFER  $n_1$ ,  $n_1 \neq 0$ , trebuie alocată (comutator DSR) și pusă în corespondență cu o descriere de tip FORMAL, printr-o cerere de forma  $M X_1 = \text{FORMAL id-formal-niv-1 DANS BUFFER } n_1$ .

— Zona BUFFER 0 este alocată implicit (deci nu trebuie să intre în calculul zonei DSR) și opțional poate fi asociată unei descrieri FORMAL de nivel 1.

- În urma unei cereri ECRIRE forma (2) zona articol BUFFER  $n_1$  rămâne nemodificată.
- Inițializarea zonei BUFFER  $n_1$  se poate face prin cererea S  $X_1$  inițializarea făcându-se cu blăncuri pt  $n_1 = 0$  și cu zero pt  $n_1 \neq 0$ .

## Cerere EXEC

**Funcțiune** Execută un program-precompilat existent în bibliotecă.

### Sintaxa

EXEC identificator [( $X_1 X_2 \dots$ )]

- identificator este numele *programului-precompilat*.
- variabilele X din listă pot fi puse în orice ordine.

### Comentarii

— La întilnirea cererii EXEC în fișierul-de-intrare se suspendă execuția programului și se lansează un program în format interpretabil catalogat anterior.

— La sfirșitul execuției programului-precompilat controlul este redat următoarei cereri după EXEC.

— Un *program-precompilat* poate conține la rindul său cereri EXEC. Numărul maxim de apeluri imbricate este 30.

— Căutarea *programului-precompilat* în bibliotecă se face la execuție, astfel încit actualizarea bibliotecii de programe-precompilate nu necesită recompilarea programelor ce conțin cereri EXEC.

— O consecință directă a faptului că verificarea existenței programului precompilat se face la execuție, și nu la analiză, este posibilitatea apelului recursiv.

— Variabilele  $X_1, Y_1, W_1, Z_1$  sînt globale atit pentru programul apelant cit și pentru cel apelat.

— Dacă este prezentă lista ( $X_1 X_2 \dots$ ), variabilele X indicate vor fi salvate într-o stivă și restaurate după revenirea din programul precompilat.

## Cerere FAIRE

**Funcțiune** Printr-o cerere FAIRE putem obține execuția repetitivă a unei secvențe de cereri.

### Sintaxa

FAIRE cerere [cerere...] FIN

- cerere poate fi oricare din setul de cereri SOCRATE.

### Comentarii

— Secvența de cereri delimitată de cuvintele-cheie FAIRE și FIN se execută cerere după cerere pînă la întilnirea unei cereri REFAIRE sau SORTIE, sau pînă la terminarea secvenței de cereri.

— La întilnirea cererii REFAIRE, se reia execuția secvenței de cereri.

— La întilnirea cererii SORTIE, se continuă execuția cu cererea care urmează după cuvîntul-cheie FIN de sfirșit al cererii FAIRE.

— Instrucțiunile SORTIE și REFAIRE pot apare în interiorul unei cereri condiționale determinînd astfel părăsirea sau reluarea buclei pe o anumită condiție logică.



## Cerere G

**Funcțiune** Prin instrucțiunea G se generează un element nou al unei mulțimi de tip entitate sau invers.

**Sintaxa**

G citare-complexă-1  $\left[ = \left\{ \begin{array}{l} \text{citare-complexă-2} \\ X_1 \end{array} \right\} \right]$

- citare-complexă-1 se referă la mulțimea ce se va completa cu elemente noi
- citare-complexă-2 se referă la elementele noi ce se adaugă mulțimii
- $X_1$  desemnează realizarea unei entități ce se va include în mulțime.

**Comentarii**

– Deosebim două forme sintactice :

(1) G citare-complexă-1

(2) G citare-complexă-1:  $\left[ = \left\{ \begin{array}{l} \text{citare-complexă-2} \\ X_1 \end{array} \right\} \right]$

- Forma (1) se referă la generarea unei mulțimi de tip entitate. Utilizarea filtrelor pentru citare-complexă-1 nu este permisă.

Forma sintactică pentru citare-complexă-1 este :

$$\left\{ \begin{array}{l} \text{UN} \\ \text{TOUT} \end{array} \right\} \text{identificator } [X_1] \left\{ \begin{array}{l} \text{nr-intreg-poz} \\ Y_1 \end{array} \right\} [\text{calif}]$$

– Dacă este prezent nr-de-realizare atunci va fi generată o realizare a entității avînd numărul de realizare egal cu cel specificat.

– Generarea unei realizări deja existente provoacă tipărirea unui mesaj de eroare și ignorarea cererii de generare pentru acea realizare.

– Dacă nu se precizează nr-de-realizare atunci se generează prima realizare negenerată.

– Generarea unei realizări dintr-o entitate se materializează în poziționarea pe 1 a bitului, corespunzător numărului de realizare, în lanțul de biți de prezență atașat caracteristicii de tip entitate.

Cuantificatorul UN sau TOUT specifică dacă este vorba de generarea unei singure realizări sau a tuturor realizărilor entității.

- Forma (2) se referă la generarea unei mulțimi de tip invers.

– Cuantificatorii UN sau TOUT, din citare-complexă-1, nu au nici o semnificație în acest caz.

– Citare-complexă-2 poate fi o caracteristică de tip entitate, invers, inel sau referință ce selectează anumite realizări ale entității inversate. În funcție de realizările selectate se poziționează pe 1 biții din lanțul de biți al inversei.

– Citare-complexă-2 determină realizările entității, pentru care s-a prevăzut invers în structură și care vor intra în mulțimea de tip invers care se generează.

– Caracteristica de tip invers specificată prin citare-complexă-1 trebuie să fie de același nivel cu entitatea pe care o inversează.

– Nu este permisă utilizarea filtrelor și a nr-de-realizare pentru caracteristica de tip invers citare-complexă-1 a cărei formă sintactică devine :

$$\left\{ \begin{array}{l} \text{UN} \\ \text{TOUT} \end{array} \right\} \text{Identificator } [\text{calificare}]$$

– Utilizarea variabilei  $X_1$ , în partea dreaptă, presupune actualizarea ei în prealabil cu adresa virtuală a unei realizări din entitatea inversată.

## Exemple

### 1 G UN angajat X1

Se generează următoarea realizare a entității angajat și se poziționează X1 pe această realizare.

### 2 G UNE secție Y1

Se generează realizarea cu numărul din Y1 a entității secție.

### 3 G UN atelier DE UNE secție

Se generează realizarea următoare a entității atelier din prima realizare a entității secție.

### 4 G TOUT secție

Se generează toate realizările entității secție.

### 5 G UN tesa = TOUT angajat ayant funcție='contabil' ;

Se generează în inversul tesa toate realizările entității angajat ce verifică condiția din filtru.

### 6 G TOUT tesa = X1

Se adaugă în inversul tesa realizarea entității angajat punctată prin X1.

### 7 G TOUT tesa = TOUT pers-atelier DE UN atelier 5 DE UNE secție 2

Se adaugă în inversul tesa toate realizările entității angajat care fac parte din inelul pers-atelier din realizarea 5 a entității atelier care la rîndul ei face parte din realizarea 2 a entității secție.

## Cerere I

**Funcțiune** Cererea I este utilizată pentru *tipărirea* valorilor caracteristicilor simple, a variabilelor de lucru, a constantelor Z, precum și pentru *interogarea* caracteristicilor compuse și a celor de tip mulțime.

## Sintaxa

$$I [( [+ ] A [B])] \left\{ \begin{array}{l} \text{citare-complexă} \\ Y_1 \\ W_1 \\ Z_1 \\ X_1 \\ \text{constantă Z} \end{array} \right\}$$

- A este un număr întreg pozitiv cuprins între 0 și 132 sau o variabilă Y.
- B este un număr întreg cuprins între –30 și 30 sau o variabilă Y.
- Citare-complexă desemnează caracteristica ce se dorește interogată sau a cărei valoare trebuie tipărită.
- Constanta Z este un șir alfanumeric delimitat de apostroafe.

## Comentarii

— Deosebim două forme sintactice :

- (1) | ([+] A [B]) obiect  
 (2) | obiect

— Forma (1) va determina o editare cu format, a valorii obiectului specificat, în zona articol a fișierului de ieșire.

Obiectul tipăririi poate fi :

- .caracteristica-simplă (mot, listă-de-valori, numeric)
- .variabilă-de-lucru ( $Y_1, W_1, Z_1$ )
- .constantă Z.

Cîmpul [+] A determină poziția în buferul de ieșire unde se va face transferul valorii ce se tipărește.

Prezența semnului „+” înseamnă că se lasă A blankuri la dreapta ultimului caracter înscris în bufer.

Buferul de ieșire începe din poziția 1.

Numărul de caractere tipărite este minimul dintre :

B  
 numărul de caractere al valorii.

Dacă valoarea lui B este pozitivă, tipărirea are loc de la stînga la dreapta plecînd din poziția determinată de cîmpul [+] A, cu trunchierea eventual, la dreapta. Trunchierea valorii tipărite are loc dacă B este mai mic decît numărul de caractere al valorii ce trebuie tipărită.

Dacă valoarea lui B este negativă, tipărirea are loc de la dreapta la stînga, plecînd tot din poziția determinată de A, și eventual cu trunchiere, la stînga, a acelei valori.

Valoarea „nedefinit” este editată prin blankuri.

Editarea buferului de ieșire, formatat printr-o succesiune de cereri I, este făcută efectiv la înlînirea unei cereri ECRIRE simple.

După editarea buferului de ieșire, acesta va fi inițializat cu blank și se va face re poziționarea pe început (poziția 1).

— Forma (2) va determina editarea fără format a obiectului specificat în cadrul cererii.

Dacă obiectul solicitat în cererea I este:

- .caracteristică-simplă de tip mot, listă-de-valori, numeric
- .variabilă-de-lucru  $Y_1, W_1, Z_1$

atunci se va edita o linie de forma :

$$\left\{ \begin{array}{c} \text{nume-caracteristică} \\ Y_1 \\ W_1 \\ Z_1 \end{array} \right\} = \text{valoare}$$

Dacă obiectul solicitat este

- .caracteristică de tip referință
- .variabilă-de-lucru  $X_1$

se va edita o linie de forma :

nume-caract = nume-entitate nr-realiz [DE nume-ent nr ...]

În partea dreaptă a egalului apare numele entității referite, numărul său de realizare și eventual calificarea numerotată

Dacă obiectul cererii I este

caracteristică de tip entitate, Inel, Invers

se vor lista toate realizările acestor mulțimi, iar pentru fiecare realizare se vor lista toate caracteristicile sub forma :

nume-caract = valoare

pentru mot, listă-de-valori, numeric

nume-caract = ANNEAU

[nume-ent nr [DE nume-ent nr ...] ]...  
pentru *inel*.

nume-caract = REFERE [nume-ent nr [DE nume-ent nr ...]]  
pentru *referință*

nume-caract = INVERSE

[nume-ent nr] ...  
pentru *invers*

Dacă obiectul cererii este bloc se vor lista toate caracteristicile sale.

Dacă obiectul cererii este un șir de caractere între apostroafe, se va tipări o linie conținând în primele poziții, șirul de caractere specificat.

## Exemple

	secvența program	mesaj tipărit
1	I (4 3) I (+1 3)	<u>ABCDEF</u> <u>ZYXW'</u>
2	I (5 -3) M Y1 = 3 I (+1 Y1)	<u>ABCDEF</u> <u>DEF ZYX</u>
3	I (3 3) M Y1 = 0 I (+Y1 2)	<u>ABCDEF</u> <u>ABCZY</u>
4	I (10 4) I (5 -3) M Y1 = 2 I (+Y1 Y1)	<u>'012345'</u> <u>ABCDEF</u> <u>DEF ZY 0123</u>
5	I (3 3) I (+4 -2)	<u>ABCDEF</u> <u>ZYXW'</u> <u>ABC XW</u>
6	I (8 -3) M Y1 = -3 I (+0 Y1)	<u>ABCDEF</u> <u>ZYXW'</u> <u>YXW</u>
7	I (8 -3) M Y1 = 2 I (+Y1 -3)	<u>ABCDEF</u> <u>ZYXW'</u> <u>DYXW</u>
8	I (1 3) M Y1 = 6 M Y2 = -3 I (+Y1 Y2)	<u>ABCDEF</u> <u>ZYXW'</u> <u>ABC YXW</u>

### Comentarii

— Exemplele 1—8 se referă la cereri de tipărire cu format.

— Mesajele tipărite apar subliniate pentru a pune în evidență poziția caracterelor în cadrul liniei. Fiecare liniuță corespunde la o poziție în cadrul liniei, de la 1 la sfârșitul mesajului (max. 132).

9. I TOUT angajat  
Listează toate realizările entității angajat.
10. I TOUT atelier DE UNE secție X1  
Listează toate realizările entității atelier din prima secție și poziționează variabila X1.
11. I TOUT tesa  
Listează toate realizările entității angajat care fac parte din inversul tesa.
12. I șef-secție DE X1  
Listează realizarea entității angajat către care punctează referința șef-secție din realizarea entității secție punctată de X1.
13. I UN pers-atelier DE UN atelier 2 DE X1  
Listează realizarea entității angajat care se află pe prima poziție în inelul pers-atelier din realizarea 2 a entității atelier din realizarea entității secție punctată de X1.

*Comentarii*

— Exemplele 9—13 se referă la cereri de interogare a bazei de date.

**Cerere LIBERER**

*Funcțiune* Deblochează accesul la baza de date, acces interzis anterior printr-o cerere BLOQUER din același program.

*Sintaxa*

**LIBERER**

**Cerere LIRE**

*Funcțiune* Cererea LIRE determină citirea unui articol dintr-un fișier secvențial.

*Sintaxa*

**LIRE BANDE  $n_1$  DANS BUFFER  $n_2$**

$n_1$  este numărul unui fișier-logic ( $1 \leq n_1 \leq 10$ )

$n_2$  este numărul unui bufer din memoria centrală ( $1 \leq n_2 \leq 10$ ).

*Comentarii*

— Fișierul-logic BANDE  $n_1$  trebuie asignat, în prealabil, unui fișier fizic prin cererea ATTACHE.

— Zonei BUFFER  $n_2$ , în care s-a transferat articolul citit, i se poate asocia un format de nivel 1 prin cererea M asupra unei variabile  $X_1$ .

— Întilnirea mărcii de sfârșit de fișier este anunțată prin poziționarea valorii 1 în variabila ERREUR.

**Exemplu**

FAIRE

.....

LIRE BANDE 5 DANS BUFFER 2

SI ERREUR = 1 ALORS SORTIE FIN

.....

REFAIRE

FIN

**Cerere M**

*Funcțiune* Prin cererea M avem posibilitatea să actualizăm caracteristici din baza de date, variabile de lucru  $Y_1, W_1, Z_1, X_1$ , precum și zone articol din memorie peste care am suprapus descrieri formale.

## Sintaxa

<b>M</b>	$X_i =$	$\left\{ \begin{array}{l} \text{citare-formal} \\ \text{citare-complexă} \\ X_i \\ \text{SUIVANT DE citare-complexă} \\ U \end{array} \right\}$
	$Y_i =$	$\left\{ \begin{array}{l} D \text{ citare-complexă} \\ \text{NUMDE } \left\{ \begin{array}{l} \text{citare-complexă} \\ X_i \end{array} \right\} \\ \text{expresie-aritmetică} \\ \text{citare-complexă} \\ \text{citare-formal} \\ \text{EXT} \\ Z_i \\ U \end{array} \right\}$
	$W_i =$	$\left\{ \begin{array}{l} \text{expresie-aritmetică} \\ \text{citare-complexă} \\ \text{citare-formal} \\ \text{EXT} \\ Z_i \\ U \end{array} \right\}$
	$Z_i =$	$\left\{ \begin{array}{l} Z_i \\ \text{citare-complexă} \\ \text{citare-formal} \\ \text{constanta } Z \\ \text{EXT} \\ Y_i \\ W_i \\ U \end{array} \right\}$
	$Z_i(a) =$	$Z_j(b \ c)$
	$\left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \end{array} \right\} =$	$\left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ \text{EXT} \\ X_i \\ Z_i \\ \text{constantă } Z \\ \text{expresie-aritmetică} \\ U \end{array} \right\}$

## Comentarii

— Forma sintactică

(1) M operand-string — EXT

poate apare și sub forma

(2) M operand-string — EXT ( constantă Z )  
Z<sub>i</sub>

Forma (2) permite afișarea la terminal a unui șir de caractere dorit, în locul denumirii caracteristicii sau variabilei receptoare, înainte de a prelua informația de la tastatură.

— Dacă operandul drept are valoarea nedefinită, aceeași valoare se atribuie și operandul stâng.

Excepție face caracteristica de tip referință asociată unui *inel*, atunci când apare pe post de operand stâng, caz în care valoarea ei rămâne neschimbată.

— În cazul în care operandul drept poziționează, prin evaluarea sa, variabila ERREUR, operandul stâng primește de asemenea valoarea nedefinită.

Excepție fac situațiile în care :

a. operandul stâng este o caracteristică de tip referință asociată unui *inel*, valoarea ei rămânând neschimbată ;

b. operandul stâng poziționează, la rândul său, variabila ERREUR, caz în care instrucțiunea M devine inefectivă.

— Evaluarea expresiilor aritmetice se face calculând toate rezultatele intermediare cu o zecimală în plus față de numărul de zecimale al operandului stâng.

Operand 1 =  $X_1$

### Sintaxa

$$M X_1 = \left\{ \begin{array}{l} \text{citare-complexă-1} \\ \text{citare-formal} \\ X_j \\ \text{SUIVANT DE citare-complexă-2} \\ U \end{array} \right\}$$

— *citare-complexă-1* se referă la o caracteristică de tip mulțime (entitate, *inel*, invers) sau referință.

— *citare-formal* se referă la o caracteristică formal de nivel 1 sau la un formal repetitiv.

— *citare-complexă-2* se referă la o caracteristică de tip mulțime (entitate, *inel*, invers)

### Comentariu

Deosebim 5 forme sintactice :

Forma (1)  $M X_1 = \text{citare-complexă-1}$

va face din  $X_1$  un indicator spre realizarea indicată de *citare-complexă-1*

Forma (2)  $M X_1 = \text{citare-formal}$

va face din  $X_1$  un indicator spre o zonă din memoria centrală peste care se va suprapune o caracteristică formală.

Forma (3)  $M X_1 = X_j$

va determina un transfer al informației din  $X_j$  în  $X_1$ .

Forma (4)  $M X_1 = \text{SUIVANT DE citare-complexă-2}$

va face din  $X_1$  un indicator spre realizarea următoare celei spre care puncta deja  $X_1$ .

Dacă  $X_1$  conține valoarea nedefinită, atunci va puncta spre prima realizare indicată prin *citare-complexă-2*.

Dacă nu mai există succesor pentru realizarea curentă atunci  $X_1$  primește valoarea nedefinită.

În cazul în care *citare-complexă-2* se referă la o caracteristică îmbricată, calificarea ei va trebui să fie făcută fie direct cu o variabilă  $X_1$ , fie prin calificare explicită pînă la primul nivel.

Forma (5)  $M X_1 = U$

determină punerea pe nedefinită a variabilei  $X_1$ .

Dacă  $X_1$  referă o zonă de memorie (BUFFER n) aceasta din urmă va fi eliberată.

Citățile complexe care apar în dreapta unei atribuiri  $X_1$  vor specifica o singură realizare dintr-o mulțime (cuantificator UN).

**Exemple****1 M X1 = FORMAL F-ANGAJAT DANS BUFFER 2**

Asociază unei zone de memorie BUFFER 2 o structură logică descrisă prin caracteristica formal de nivel 1 f-angajat și atribuie variabilei X adresa de început a acestei zone.

**2 M X1 = UNE secție 3**

I se atribuie variabilei X1 adresa virtuală a realizării 3 din entitatea secție, dacă aceasta din urmă era generată, și valoarea nedefinit în caz contrar.

**3 M X1 = SUIVANT DE UNE secție**

Dacă X1 puncta spre realizarea 3 din entitatea secție, în urma acestei cereri va puncta spre următoarea realizare generată a entității secție.

Dacă nu există realizări cu număr > 3 atunci X1 va fi pus pe nedefinit.

Dacă X1 conținea valoarea nedefinit, în urma acestei cereri va puncta spre prima realizare generată a entității secție.

Operand 1 =  $Y_1$

*Sintaxa*

$$M Y_1 = \left\{ \begin{array}{l} D \text{ citare-complexă-1} \\ \text{NUMDE } \left\{ \begin{array}{l} \text{citare-complexă-2} \\ X_1 \end{array} \right\} \\ \text{expresie-aritmetică} \\ \text{citare-complexă-3} \\ \text{citare-formal} \\ \text{EXT} \\ Z_1 \\ U \end{array} \right\}$$

- Citare-complexă-1 se referă la o caracteristică de tip mulțime.
- Citare-complexă-2 se referă la o caracteristică de tip mulțime sau la o caracteristică de tip referință.
- Expresie-aritmetică este o mulțime de variabile de lucru  $Y_1, W_1$ , constante-numerice, legate prin operatori aritmetici uzuali și grupate eventual în subexpresii prin paranteze. Operatorii aritmetici, în afară de minus (–) sînt separatorii ai limbajului de manipulare.
- Citare-complexă-3 se referă la o caracteristică simplă de tip *numeric*.
- Citare-formal specifică o caracteristică formal simplă de tip *numeric*.

*Comentarii*

– Deosebim 9 forme sintactice :

Forma (1)  $M Y_1 = D \text{ citare-complexă-1}$   
ne permite să afectăm unei variabile  $Y_1$  numărul de realizări (elemente) ale unei mulțimi (entitate, inel, invers) răspunzînd eventual la un filtru.

Forma (2) și (3)  $M Y_1 = \text{NUMDE } \left\{ \begin{array}{l} \text{citare-complexă-2} \\ X_1 \end{array} \right\}$



ne permite să afectăm unei variabile  $Y_1$  numărul de realizare al unui element dintr-o mulțime sau o referință specificat prin citare-complexă-2 sau variabilă  $X_1$ .  
 citare-complexă va avea cuantificatorul UN.

- Forma (4)**  $M Y_1 = \text{expresie-aritmetică}$   
 permite să atribuim unei variabile  $Y_1$  valoarea unei expresii aritmetice.  
 Dacă în cadrul evaluării expresiei aritmetice apar depășiri sau valori nedefinite, atunci lui  $Y_1$  i se va atribui valoarea nedefinit.  
 Dacă valoarea expresiei aritmetice conține și parte zecimală, aceasta se va pierde prin atribuire la  $Y_1$ .
- Forma (5)**  $M Y_1 = \text{citare-complexă}$   
 permite atribuirea la  $Y_1$  a valorii unei caracteristici simple, de tip numeric, din baza de date.
- Forma (6)**  $M Y_1 = \text{citare-formal}$   
 permite atribuirea la  $Y_1$  a valorii unei caracteristici formal simple, de tip numeric, obținută dintr-o zonă de memorie la care am atașat o descriere cu formal.
- Forma (7)**  $M Y_1 = \text{EXT}$   
 va atribui lui  $Y_1$  o valoare numerică întreagă obținută de la terminal.  
 Dacă utilizatorul tastează <RET> fără a furniza o valoare, variabila  $Y_1$  rămâne nemodificată.  
 Dacă utilizatorul tastează o valoare nenumerică, solicitarea de la terminal este reluată.
- Forma (8)**  $M Y_1 = Z_j$   
 determină conversia șirului de caractere conținut de  $Z_j$  și atribuirea la  $Y_1$  a valorii binare obținute.  
 Dacă apar erori la conversie (caracter nenumeric etc.), variabila  $Y_1$  va fi pusă pe nedefinit.
- Forma (9)**  $M Y_1 = U$   
 permite punerea pe nedefinit a unei variabile  $Y_1$ .

**Exemple**

- 1**  $M Y_2 = D \text{ TOUT } \text{copil DE UN } \text{angajat avec } \text{nume} = \text{'POPESCU'} ;$   
 Va furniza în variabila  $Y_2$  numărul de copii ai angajatului POPESCU.
- 2**  $M Y_1 = D \text{ TOUT } \text{pers-atelier AYANT } \text{stare-politică} = \text{'UTC'} ;$   
 $\text{DE UN atelier 2 DE UNE } \text{secție}$   
 Va atribui lui  $Y_1$  numărul de angajați membri UTC din atelierul 2 al primei secții.
- 3**  $M Z_1 = \text{'123.24'}$   
 $M Y_1 = Z_1$   
 În urma acestor atribuiri  $Y_1$  va conține valoarea 123.
- 4**  $M Y_1 = 500 + Y_2 * 10 - (25 + Y_3 / Y_4)$   
 Va atribui lui  $Y_1$  valoarea întreagă a expresiei din membrul drept.
- 5**  $M Y_1 = \text{NUMDE } \text{șef-atelier DE UNE } \text{secție } X_1]$   
 Va atribui lui  $Y_1$  numărul de realizare al entității angajat referite prin caracteristica șef-atelier din prima realizare a entității secție.
- 6**  $M Y_2 = \text{NUMDE } X_1]$   
 Va atribui lui  $Y_2$  numărul de realizare al entității punctate de  $X_1$ .

7 M Y3 = marca DE UN angajat Y2

Va atribui lui Y2 valoarea caracteristicii numerice marca din realizarea cu numărul Y2 a entității angajat.

8; M X1 = FORMAL f-angajat DANS BUFFER 5  
LIRE BANDE 1 DANS BUFFER 5  
M Y6 = salariu DE X1

I se atribuie lui Y6 valoarea numerică dintr-o descriere formală suprapusă peste un articol citit din fișierul logic BANDE 1.

9 M Y10 = EXT

Va tipări la terminal șirul 'Y10=' după care se va aștepta tastarea unui număr a cărui valoare va fi atribuită lui Y10.

10; M Y7 = U

Va pune pe nedefinit variabila Y7.

Operand 1 =  $W_1$

Sintaxa

$$M W_1 = \left\{ \begin{array}{l} \text{expresie-aritmetică} \\ \text{citare-complexă} \\ \text{citare-formal} \\ \text{EXT} \\ Z_1 \\ U \end{array} \right\}$$

Rămân valabile toate precizările de la paragraful precedent cu privire la formele sintactice de mai sus atribuite unei variabile  $Y_1$ .

Singura diferență este aceea că nu mai are loc trunchierea valorii numerice din membrul drept.

Operand 1 =  $Z_1$

Sintaxa

$$M Z_1 = \left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ \text{constantă } Z \\ \text{EXT} \\ Z_1 \\ Y_1 \\ W_1 \\ U \end{array} \right\} \text{ sau } Z_1[(p_1)] = Z_1(p_2 \lg)$$

citare-complexă se referă la o caracteristică simplă de tip *mot* sau *listă-de-valori*.  
citare-formal specifică o caracteristică formal simplă, de tip alfanumeric.

- $p_1, p_2$  definesc poziții de la 1 la 30 dintr-o variabilă Z.  
 $p_1$  are valoarea implicită 1.  
 lg definește lungimea informației de transferat.

#### Comentarii

- Valoarea operandului drept, convertită eventual în format extern ( $Y_1, W_1$ ), este atribuită variabilei  $Z_1$ .
  - La forma cu EXT dacă nu se specifică nici o valoare și se tastează direct <RET>, atunci variabila  $Z_1$  va rămâne nemodificată.
  - La preluarea cu EXT a datelor, se poziționează cursorul la început de rând nou prin caracterele <CR> și <LF>.
- În cazul în care operandul stâng este o variabilă  $Z_1$ , se poate inhiba această poziționare cu <CR> și <LF> prin furnizarea caracterului \$ pe prima poziție în cadrul unui șir sau a unei variabile  $Z_1$ , utilizând una din formele

$M Z_1 = EXT ('\$...')$

sau

$M Z_1 = '\$...'$

$M Z_1 = EXT (Z_1)$

Caracterul \$ nu apare în cadrul mesajului tipărit; șirul de caractere sau variabila  $Z_1$  poate conține numai caracterul \$, caz în care se păstrează poziționarea cursorului fără tipărirea suplimentară a vreunui mesaj.

— La forma  $Z_1(p_1) = Z_1(p_2 \text{ lg})$ , șirul de caractere din variabila  $Z_1$ , din poziția  $p_2$ , de lungime lg, este transferat în variabila  $Z_1$ , începând cu poziția  $p_1$ . Celelalte poziții ale variabilei  $Z_1$ , atât la stînga cit și la dreapta, rămân nemodificate.

**Operand 1 = citare-complexă \***

#### Sintaxa

$$M \text{ citare-complexă-1} = \left\{ \begin{array}{l} \text{citare-complexă-2} \\ \text{citare-formal} \\ \text{EXT} \\ X_1 \\ Z_1 \\ \text{constantă } Z \\ \text{expresie-aritmetică} \\ U \end{array} \right\}$$

- citare-complexă-2 specifică o caracteristică simplă sau o caracteristică de tip mulțime sau referință.
- citare-formal specifică o caracteristică formal simplă.

#### Comentarii

- Forma (1) sau (4)  $M \text{ citare-complexă-1} = \left\{ \begin{array}{l} \text{citare-complexă-2} \\ X_1 \end{array} \right\}$

Dacă citare-complexă-2 sau  $X_1$  referă o singură realizare (cuantif. UN dintr-o mulțime, iar citare-complexă-1 identifică o caracteristică de tip referință, atunci se va actualiza referința și eventual inelul asociat. Aceasta este singura modalitate de actualizare a unui inel.

Citare-complexă-1 poate referi mai multe realizări prin cuantificatorul TOUT aplicat pe lanțul de calificări, caz în care se actualizează mai multe referințe iar în Inel se pot include mai multe elemente.

- Forma (1) și (2) Dacă citare-complexă-1, specifică o caracteristică-simplă din baza de date, atunci valoarea caracteristicii din membrul drept va fi atribuită caracteristicii-simple din membrul stîng. Caracteristicile din cei doi membri trebuie să fie compatibile ca tip, altfel se va semnala o eroare în faza de analiză.

Celelalte forme sintactice permit atribuirea valorii din membrul drept, caracteristicii simple, de același tip, din membrul stâng.

Dacă citare-complexă-1 este cheie unică și i se atribuie o valoare deja existentă în dicționar, se atribuie valoarea U, caracteristicii corespunzătoare și se semnalează acest lucru prin poziționarea codului 17 în variabila ERREUR.

Forma EXT permite atribuirea unei valori introduse de la tastatura terminalului.

Forma dialogului este :

nume-caracteristică-simplă = valoare <RET>

În care, ceea ce figurează după semnul = va fi introdus de utilizator de la terminal.

Dacă se tastează direct <RET>, valoarea caracteristicii simple din bază rămâne neschimbată.

Dacă se furnizează o valoare conformă cu tipul caracteristicii, și domeniul ei de valori, acea valoare va fi înscrisă în baza de date, iar în caz contrar se va cere o nouă valoare din terminal.

Dacă operandul 1 este o caracteristică simplă declarată cheie unică, iar valoarea furnizată din terminal nu poate fi înscrisă în dicționar pentru că există deja, variabila ERREUR va fi încărcată cu codul de eroare corespunzător și nu se mai revine pe o nouă solicitare din terminal (se presupune ca în acest caz, din program se poate lua o decizie mai bună decât din exterior).

#### Exemple

```
1] G UN angajat Y1 X2
   M X1 = FORMAL f-angajat DANS BUFFER 2
   LIRE BANDE 1 DANS BUFFER 2
   M nume DE X2 = nume DE X1
```

Printr-o astfel de secvență se pot actualiza caracteristici simple din baza de date prin valori ale unor caracteristici-formal extrase din zona articol a unui fișier.

```
2] M funcție DE X1 = EXT
```

Se actualizează caracteristica funcție cu valoarea obținută de la tastatura terminalului.

```
3] M șef-secție DE UNE secție X1 3 = U
```

Se pune pe nedefinit caracteristica șef-secție din realizarea 3 a entității secție.

```
4] M apartenența-atelier DE UN angajat Y1 = UN atelier DE X.
```

Se completează inelul pers-atelier din prima realizare a entității atelier din entitatea secție punctată de X2, cu un nou element (realizarea Y1 din entitatea angajat) și se actualizează referința apartenența-atelier ce va puncta spre entitatea atelier.

```
5] M apartenența-atelier DE TOUT angajat AYANT marca > 1200 ;
   = UN atelier DE X2
```

Determină completarea inelului pers-atelier cu toți angajații având marca > 1200 și actualizarea corespunzătoare a referinței acestora spre entitatea atelier.

```
6] M șef-secție DE X2 = UN angajat AVEC nume = 'IONESCU' ;
```

Permite actualizarea referinței din entitatea secție spre o realizare din entitatea angajat ce răspunde la un filtru.

**Operand 1 = citare-formal**

*Sintaxa*

$$M \text{ citare-formal-1} = \left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal-2} \\ \text{EXT} \\ Z_1 \\ \text{constantă } Z \\ \text{expresie-aritmetică} \\ U \end{array} \right\}$$

– Citare-complexă este o caracteristică simplă din baza de date, de tip *mot, listă-de-valori, numeric*.

– Citare-formal-1, citare-formal-2, sînt caracteristici formal simple de forma : identificator DE  $X_1$

*Comentarii*

Valoarea obiectului din membrul drept este atribuită, după eventuale conversii, unul cîmp dintr-o zonă de memorie , identificat prin citare-formal-1.

**Calculul expresiilor aritmetice cu rotunjire**

*Sintaxa*

$$M \left\{ \begin{array}{l} \text{citare-complexă} \\ \text{citare-formal} \\ Y_1 \\ W_1 \end{array} \right\} == \text{expresie-aritmetică}$$

– Citare-complexă sau citare-formal se referă la o caracteristică simplă, din baza sau formal, de tip numeric.

*Comentarii*

– Utilizarea operatorului '==' permite rotunjirea valorii calculate a expresiei aritmetice la numărul de zecimale al operandului sting.

– Operatorul '==' este echivalent cu opțiunea ROUNDED a instrucțiunii COMPUTE din COBOL.

– Evaluarea unei expresii aritmetice se face calculînd toate rezultatele intermediare cu o zecimală în plus față de numărul de zecimale al operandului sting. Din acest motiv pot apare erori de rotunjire în cazul nerespectării convenției COROL de scriere a expresiilor aritmetice.

**Exemple**

În urma evaluării următoarelor expresii aritmetice, se obțin rezultatele :

M Y1 == 1000 \* (1 + 1.3456789)

Y1 = 2300

M W1 == 1000 \* (1 + 1.3456789)

W1 = 2345.600

M W1 == 1000 + 1000 \* 1.3456789

W1 = 2345.679

## Cerere PAUSE

**Funcțiune** Pune în așteptare programul pînă cînd se poate lansa crearea punctului de reluare.

### Sintaxa

PAUSE [ $Y_1$ ]

—  $Y_1$  va primi după execuția cererii PAUSE valoarea numărului punctului de reluare dacă programul se execută într-o sesiune cu fișier de securitate și va rămîne nemodificat în caz contrar.

#### Comentariu

— Cererea PAUSE trebuie să preceadă setul de instrucțiuni care actualizează baza de date.

— Dacă în momentul atingerii de către program a instrucțiunii PAUSE s-a declanșat procesul de creare a punctului de reluare datorită cererilor efectuate de alte programe, programul este pus în așteptare pînă la terminarea procesului în curs.

— O cerere PAUSE care urmează în secvența de execuție unei alte cereri PAUSE, este inefectivă.

La terminarea execuției unui program de cereri, dacă a rămas un PAUSE fără un CKPT corespondent, se execută în mod automat un CKPT.

## Cerere POUR

**Funcțiune** Determină o secvență repetitivă de cereri calificată implicit printr-o citare.

### Sintaxa

POUR  $\left. \begin{array}{l} \text{citate-complexă} \\ X_1 \end{array} \right\}$  cerere [cerere...] FIN

— citare-complexă se referă la o caracteristică de tip mulțime (*entitate*, *Inel*, *Invers*) sau *referință*, bloc.

#### Comentariu

— Secvența de cereri se execută pentru fiecare element al mulțimii specificate prin citare-complexă.

— ieșirea forțată din buclă se poate face prin cererea SORTIE.

— Reluarea forțată a buclei pentru următorul element se face prin cererea SUIVANT (sau REFAIRE).

— Cererile din secvență vor avea operanzii (în afară de cazurile de calificare neimplicită) calificați implicit prin nivelul citărilor-complexe specificate.

— Folosirea lui  $X_1$  va determina execuția o singură dată a secvenței de cereri și va introduce un nivel de calificare implicit, corespunzător entității asupra căreia punctează variabila  $X_1$ .

În caz că  $X_1$  este nedefinit, nu se va mai executa secvența de cereri.

— Numărul de îmbricări permise pentru cererile POUR este limitat la 10.

— Dacă cererile POUR sînt prevăzute cu acces dicționar pe caracteristici de tip mulțime, numărul de îmbricări permise pentru astfel de cereri este limitat la 4.

## Exemple

1    POUR TOUT angajat AVEC nume='POPESCU' ;  
       I nume DE apartenență-atelier  
       FIN

Se listează numele atelierelor din care fac parte toți angajații cu numele POPESCU.

2    POUR UNE secție 5  
       I nume  
       POUR TOUT atelier  
           I nume DE TOUT pers-atelier.  
       FIN  
       I nume DE șef-secție  
       FIN

Se listează numele secției 5, numele angajaților din această secție precum și numele șefului de secție.

3    M X1 = UN angajat AYANT EXISTE UN copil ;  
       POUR X1 I TOUT copil FIN

Se listează toate realizările entității copil din prima realizare a entității angajat avînd generate cel puțin o realizare din entitatea copil.

4    POUR TOUT angajat  
       SI marca > 1000 ALORS SORTIE FIN  
       SI marca < 900 ALORS SUIVANT FIN  
       I nume  
       FIN

Se listează numele tuturor angajaților avînd marca cuprinsă între 900 și 1.000.

## Cerere REFAIRE

*Funcțiune* Reluarea forțată a unei bucle FAIRE sau POUR.

*Sintaxa*

**REFAIRE**

*Comentarii*

— Cererea REFAIRE poate apare doar într-o cerere FAIRE sau POUR.

## Cerere S

*Funcțiune* Ștergerea uneia sau mai multor realizări dintr-o entitate sau inițializarea unei zone articol.

*Sintaxa*

$$S \left\{ \begin{array}{l} \text{citare-complexă} \\ X_1 \end{array} \right\}$$

citare-complexă poate fi o caracteristică de tip mulțime (entitate, inel, invers) sau referință.





**Comentarii**

- Se evaluează mai întâi condiția. Rezultatul evaluării poate fi una din valorile: adevărat, fals, nedefinit, determinând următoarea acțiune:
- adevărat — se execută secvența-1 și se trece la cererea următoare după SI.
- fals — se execută secvența-2 și se trece la cererea următoare după SI.
- nedefinit — se trece la cererea următoare după SI.

**Cerere SORTIE**

*Funcțiune* determină ieșirea forțată dintr-o cerere FAIRE sau POUR.

**Sintaxa****SORTIE****Comentarii**

- Această cerere nu poate apare decît în interiorul unei cereri FAIRE sau POUR.
- Execuția acestei cereri va determina trecerea la următoarea cerere de pe același nivel cu FAIRE-ul sau POUR-ul din care s-a ieșit forțat.

**Cerere SUIVANT**

*Funcțiune* Determină reluarea unei bucle POUR sau FAIRE.

**Sintaxa****SUIVANT****Comentarii**

- Această cerere nu poate apare decît în interiorul unei cereri POUR sau FAIRE.

## 4. PROGRAMUL BIBLIOTECAR

### Generalități

Programul bibliotecar permite *introducerea, ștergerea și listarea* macroinstrucțiunilor și programelor precompilate în/în baza de date.

La lansarea programului, sesiunea trebuie să fie deschisă, activîndu-se domeniile PRG și DIC (totdeauna) și STR (numai la introducerea programelor precompilate în baza de date).

— Domeniul PRG va conține macroinstrucțiunile într-o formă foarte apropiată de sursă și programele precompilate în forma intermediară rezultată în urma compilării.

- Domeniul DIC va conține un repertoriu al numelor macroinstrucțiunilor și programelor.
- Domeniul STR va fi consultat la introducerea programelor precompilate. În domeniul PRG, deoarece înainte de introducere, programele sînt mai întîi compilate.

### Apelul programului bibliotecar

Bibliotecarul acceptă linii de comandă cu următorul format :

>MGS fislist = fisint<sub>1</sub>[,fisint<sub>2</sub>]...[,fisint<sub>n</sub>]

unde :

- fislist – este specificatorul fișierului de listare ce va conține listingul sursă al macrodefiniției sau programului precompilat, mesajele bibliotecarului și eventualele mesaje de eroare.
- fisint<sub>j</sub> – este specificatorul unui fișier de intrare ce conține una sau mai multe comenzi bibliotecar împreună cu sursele macrodefinițiilor și/sau precompilatelor.

Specificatorii de fișiere au forma standard, și anume :

aparatus : [grup, membru] nume fișier. tip fișier ; versiune/comutatori

Valorile implicite pentru elementele specificatorului sînt :

- SY : – pentru aparatul fizic pe care este montat volumul.
- UIC-ul terminalului – pentru [grup, membru]
- SOC – pentru tipul fișierului de intrare.
- LST – pentru tipul fișierului de listare.
- ULTIMA VERSIUNE – pentru versiunea fișierului.

Comutatorii permisi sînt :

- SP – pentru fișierul de listare ; permite utilizarea spoolerului la afișarea fișierului de listare. Cînd nu se dorește listarea cu spooler se neagă comutatorul (NOSP sau –SP). Valoarea implicită este SP.
- CVF – pentru fișierele de intrare ; în prezența sa doar primele 60 de coloane se consideră informație utilă. Valoarea sa implicită este NOCVF (sau –CVF).
- ME – pentru fișierul de ieșire, prezența sa indică listarea pe fișierul de ieșire a expandării macroinstrucțiunilor, indiferent de nivelul de imbricare. Valoarea sa implicită este NOME (sau –ME).

#### Comentarii

- nu este acceptată utilizarea caracterului '\*' în descrierea specificatorilor de fișiere ;
- nu este permisă utilizarea fișierelor de comenzi indirecte în linia de comandă a bibliotecarului.

## Introducerea macroinstrucțiunilor în baza de date

### Noțiunea de macroinstrucțiune SOCRATE

*Macroinstrucțiunile* constituie o facilitare ce asigură construirea pe baza limbajului de manipulare a unui limbaj simplu, specific unei aplicații și care să nu necesite cunoașterea de către utilizator a sintaxei limbajului de manipulare sau structura bazei de date, avînd avantajul de a fi foarte apropiat de limbajul natural.

Macroinstrucțiunile pot fi utilizate în cazul unor secvențe relativ scurte ale limbajului de manipulare care se repetă de un număr mare de ori cu modificări relativ mici (ex. : utilizarea facilităților de tipărire, listarea capetelor de tabel, construirea tabelelor de date etc.).

Principiul de funcționare îl constituie crearea unei echivalențe între un șir de caractere ce poate conține și elemente variabile (parametri), numit model de apel și o secvență a limbajului de manipulare ce reprezintă modelul de expandare.

Odată realizată această corespondență, orice apariție a modelului de apel va fi înlocuită cu modelul de expandare în care parametrii au fost substituiți cu valorile lor actuale.

### Definirea și inserarea unei macroinstrucțiuni

Definirea unei macroinstrucțiuni cuprinde MODELUL DE APEL, MODELUL DE EXPANDARE și SFÎRȘITUL MACRODEFINIȚIEI.  
MODELUL DE APEL are următoarea sintaxă :

```
:DEFMAC  nume  macro  [ { separator de apel } ] ...
```

unde :

- :DEFMAC — cuvânt cheie indicînd începutul definirii unei macroinstrucțiuni.
- nume macro — este un șir de maximum 30 caractere (litere, cifre și '—'), în care primul este obligatoriu literă, ce reprezintă numele macroinstrucțiunii. La fiecare apariție a acestui șir se va încerca înlocuirea lui cu modelul de expandare în care valorile eventualilor parametri au fost actualizate. Nu este permisă utilizarea ca nume de macroinstrucțiune a unui cuvînt cheie al limbajului (vezi Anexa B) sau a unui nume de macro deja existent.
- : — indică poziția în modelul de apel a unui parametru ce va fi utilizat în modelul de expandare prin rangul său. Numărul maxim de parametri ai unei macroinstrucțiuni este 30. La apelul macrodefiniției se va înlocui semnul ':' cu parametrul actual cu care se dorește expandarea. Parametri de macroinstrucțiune pot fi :
  - cuvinte cheie ale limbajului de manipulare ;
  - elemente ale structurii bazei de date ;
  - nume de alte macroinstrucțiuni ;
  - șiruri oarecare de caractere ;
- separator de apel — este un șir oarecare de maximum 30 de caractere ASCII delimitat de separatori ai limbajului de manipulare. Nu este permisă utilizarea ca separator de apel a cuvintelor cheie (vezi Anexa B) sau a șirurilor de caractere ce încep cu ':'. La apelul macroinstrucțiunii, separatorii de apel vor trebui să aibă aceeași poziție și valoare ca în modelul de apel.

**MODELUL DE EXPANDARE** are următoarea sintaxă:

:EXP	}	secvență a limbajului de manipulare	}	...
		apel de macro		
		: n :		
		' : n : '		

unde :

- :EXP — este cuvîntul cheie ce indică sfîrșitul modelului de apel și începutul modelului de expandare. Acesta poate conține secvențe ale limbajului de cereri, apeluri de macroinstrucțiuni sau șiruri de caractere.
- : n : — indică locul în care se va insera în modelul de expandare parametrul cu poziția n din modelul de apel.
- ' : n : ' — are aceeași semnificație ca : n : cu deosebirea că în acest caz parametrul va fi utilizat în valoare (deci ca un șir de caractere între apostroafe).

*Comentarii*

Același parametru poate apare de mai multe ori în expandarea unei macroinstrucțiuni, dar de fiecare dată trebuie să aibă aceeași formă (fie :n:, fie ':n:').

**SFÎRȘITUL MACROINSTRUCȚIUNII** are sintaxa următoare :

:FDEF [separatori ai limbajului de manipulare] ...?

unde :

- :FDEF — este cuvînt cheie ce indică sfîrșitul modelului de expandare.
- ? — indică terminarea definirii unei macroinstrucțiuni.

Macroinstrucțiunile se rețin în baza de date doar în forma intermediară, gestionarea formatului sursă revenind utilizatorului prin facilitățile de editare oferite de sistemul de operare al minicalculatorului.

### Apelul de macroinstrucțiuni

Un apel de macroinstrucțiune se compune din numele macroinstrucțiunii urmat de parametrii și separatorii de apel în aceeași ordine și în același număr ca în modelul de apel.

Apelul de macroinstrucțiune poate apare într-o secvență a limbajului de manipulare, ca parametru într-un alt apel de macroinstrucțiuni sau ca element al modelului de expandare al unei macroinstrucțiuni.

La apariția unui apel de macroinstrucțiune, modelul de apel al acesteia va fi înlocuit în textul sursă cu modelul de expandare în care parametrii au fost actualizați cu valorile din apel.

În cazul în care modelul de apel conține un parametru ce nu este urmat de un separator de apel, la apelul macroinstrucțiunii parametrul va fi alcătuit din textul sursă pînă la primul separator al limbajului de manipulare. Dacă în modelul de apel parametrul este urmat de un separator de apel, atunci la apelul macroinstrucțiunii parametrul este constituit din textul sursă al apelului pînă la separatorul de apel (acesta fiind încadrat de doi separatori ai limbajului de manipulare).

Cînd parametrul ce este urmat de un separator de apel este un apel de macroinstrucțiune conținînd un separator de apel identic cu cel al primei macroinstrucțiuni,

primul separator de apel întâlnit este considerat ca aparținând macroinstrucțiunii de nivelul cel mai scăzut.

Pentru parametrii utilizați în modelul de expandare în valoare, la apelul macroinstrucțiunii parametrul se va constitui din textul sursă al apelului pînă la primul separator al limbajului de manipulare.

În cazul în care parametrul nu este urmat de un separator de apel, valoarea sa este textul sursă începînd de la primul caracter diferit de blanc pînă la următorul caracter blanc, exclusiv.

### Observații importante

1. Dacă definirea macroinstrucțiunii a decurs normal se afișează mesajul :

```
MACRO <nume macro> INSERAT
```

2. Nu este permisă utilizarea numelui macroinstrucțiunii în modelul ei de apel sau la apel, ca parametru actual.

3. La introducerea macroinstrucțiilor nu se face nici un control sintactic sau semantic, din punct de vedere al limbajului de manipulare, asupra elementelor modelului de expandare.

4. Nu sînt permise două macroinstrucțiuni cu același nume.

5. După modificarea unei macroinstrucțiuni în domeniul PRG (prin ștergere și inserare cu același nume), programele sursă sau macroinstrucțiunile care o apelează vor beneficia de noua variantă, dar programele precompilate introduse în domeniul PRG înainte modificării, vor conține în continuare varianta veche.

### [ Exemple de macroinstrucțiuni ]

Exemplele următoare vor utiliza structura definită în Anexa E. Pe această structură vom defini următoarele macroinstrucțiuni :

D1.

```
:DEFMAC lista-personalului din atelierul : secția :
:EXP nume DE TOUT personal-atelier DE UN atelier :1:
      DE UNE secție :2:
:FDEF ?
```

S-a definit macroinstrucțiunea lista-personalului ce are doi parametri (numărul atelierului și numărul secției), iar ca separatori de apel sînt utilizate șirurile de caractere : din, atelierul, secția.

D2.

```
:DEFMAC angajați-cu-copii :
:EXP TOUT angajat AYANT EXISTE UN copil :1:
:FDEF ?
```

Se definește macroinstrucțiunea cu numele angajați-cu-copii ce utilizează un singur parametru. Modelul de apel nu conține separatori de apel.

D3.

```
:DEFMAC născuți între : și :
:EXP AYANT an-de-naștere >= :1: ET an-de-naștere <= :2:
:FDEF ?
```

Macroinstrucțiunea definită cu numele născuți are doi parametri, iar ca separatori de apel se utilizează șirurile : între, și.

D4.

```
:DEFMAC retribuția :
:EXP salariu DE UN angajat :1: ET nume==':2:':
:FDEF ?
```

Numele macroinstrucțiunii este retribuția ; are doi parametri și nu utilizează separatori de apel.

D5.

```
:DEFMAC tovarășului :
:EXP AYANT sex=='B' ET funcție=='1:':
:FDEF ?

:DEFMAC tovarășel :
:EXP AYANT sex=='F' ET funcție=='1:':
:FDEF ?
```

Macroinstrucțiunile cu numele tovarășului, tovarășei au un singur parametru iar modelul de apel nu conține separatori de apel.

D6.

```
:DEFMAC stabilit în :
:EXP AYANT PAS loc DE naștere ==':1:':
:FDEF ?
```

Macroinstrucțiunea stabilit are un singur parametru iar modelul de apel conține ca separator șirul în.

D7.

```
:DEFMAC listează :
:EXP I :1:
:FDEF ?
```

Macroinstrucțiunea listează are un singur parametru și modelul ei de apel nu conține separatori.

Vom prezenta în continuare câteva exemple de apelare a acestor macroinstrucțiuni, precum și expandările lor.

A1.

I nume DE TOUT angajat stabilit în CONSTANȚA ?

Macroinstrucțiunea stabilit va fi înlocuită cu modelul ei de expandare în care parametrul utilizat va fi actualizat cu valoarea CONSTANȚA.

A2.

I nume DE TOUT angajat AYANT PAS loc DE naștere=='CONSTANȚA' ; ?  
 Istează lista-personalului din atelierul 5 secția 2 ?

Macroinstrucțiunea listează are ca parametru un nume de macroinstrucțiune (lista-personalului). În cazul îmbricării macroinstrucțiunilor separatorii de apel: din, atelierul, secția și parametrii cu valoarea 5 și 2 se consideră aparținând macroinstrucțiunii de nivelul cel mai scăzut (lista-personalului). Expandarea acestei cereri este următoarea :

A3.

I nume DE TOUT personal-atelier DE UN atelier 5 DE UNE secție 2 ?  
 Istează retribuția tovarășului Inginer. IONESCU ?

În acest exemplu se scoate în evidență mai bine mecanismul de stivă utilizat în expandarea macroinstrucțiunilor imbricate.

Parametrii macroinstrucțiunii retribuția sînt macroinstrucțiunea tovarășului și șirul de caractere IONESCU. Parametrul pentru macroinstrucțiunea tovarășului este șirul de caractere inginer. Expandarea cererii va fi următoarea :

```
† salariu DE UN angajat AYANT sex='B' ET funcție='inginer'
  ET nume='IONESCU' ?
```

A4.

listează angajați-cu-copii născuți între 1970 și 1976 ?

Macroinstrucțiunea angajați-cu-copii are ca parametru macroinstrucțiunea născuți ; aceasta din urmă conține parametrul 1970, 1976 separați de numele macroinstrucțiunii și între ei, de doi separatori de apel (între, și).

Expandarea acestor macroinstrucțiuni este următoarea :

```
† TOUT angajat AYANT EXISTE UN copil AYANT an-de-naștere >=1970
  ET an-de-naștere <=1976 ?
```

În încheiere vom prezenta un exemplu de macroinstrucțiune ce conține o secvență de comenzi ale limbajului de manipulare care calculează și afișează vîrsta medie a angajaților.

```
:DEFMAC vîrsta-medie-a-angajaților ( : : : : : ) în :
```

```
:EXP
  M :1:=:8:
  M :2:=0
  M :3:=0
  POUR TOUT angajat :5:
  M :6:=data DE naștere DE :5:
  M :7:(1 2)=:6:(7 8)
  M :4:=:6:
  M :4:=:1: - :4:
  M :3:=:3: + :4:
  M :2:=:2: + 1
  FIN
  M :3:=:3: / :2:
  I (3 20) 'VÎRSTA MEDIE ÎN :8: ='
  I (+1 3) :3:
  ECRIRE
```

```
:FDEF ?
```

Apelarea se face în felul următor :

vîrsta-medie-a-angajaților (Y1 Y2 Y3 Y4 X1 Z1 Z2) în 84 ?

iar expandarea este următoarea :

```
M Y1=84
M Y2=0
M Y3=0
POUR TOUT angajat X1
M Z1=data DE naștere DE X1
M Z2 (1 2)=Z1 (7 8) M Y4=Z2
M Y4=Y1 - Y4
M Y3=Y3 + Y4
M Y2=Y2 + 1
FIN
M Y3=Y3 / Y2
I (3 20) 'VÎRSTA MEDIE ÎN 84 ='
I (+1 3) Y3
ECRIRE ?
```

## Introducerea programelor precompilate în baza de date

### Noțiunea de program precompilat

Programele precompilate reprezintă secvențe ale limbajului de manipulare organizate ca subprograme ce pot fi apelate prin instrucțiunea EXEC. Sînt utilizate în cazul secvențelor mari ale limbajului de manipulare ce îndeplinesc o anumită funcțiune și sînt apelate de un număr mare de ori, cu modificări mici.

### Definirea și inserarea programelor precompilate

Se face cu următoarea sintaxă :

:DEFPRO nume program :EXP corp precompilat ? :FDEF ?

unde :

- :DEFPRO — este cuvînt cheie ce indică definirea unui program precompilat.
- nume program — este un șir de maximum 30 de caractere (litere, cifre și '—') din care primul trebuie să fie obligatoriu literă. Restricțiile sînt aceleași ca la nume macro.
- :EXP, :FDEF — cuvinte cheie cu aceeași semnificație ca la definirea macroinstrucțiunilor.
- corp precompilat — reprezintă secvența limbajului de manipulare ce realizează funcțiunea dorită. Este obligatoriu ca corpul precompilatului să fie urmat de caracterul '?

### Observații importante

1. La inserarea programelor precompilate se face analiza sintactică și semantică a corpului programului. Dacă nu s-au detectat erori și inserarea a decurs normal se tipărește mesajul :

```
PROGRAM <nume program>]INSERAT];
```

2. În baza de date se inserează doar codul intermediar al programului, gestiunea textului sursă revenind utilizatorului.

3. În această versiune nu sînt permise modificări ale contextului pentru programele precompilate.

4. Macroinstrucțiunile sînt tratate ca șiruri de caractere și din acest motiv ele pot fi inserate oriunde în cadrul unei pagini din domeniul PRG, spre deosebire de programele precompilate care se inserează totdeauna aliniate la început de pagină. Din această cauză pentru a evita pierderea de spațiu virtual și fizic se recomandă ca la inserare să se grupeze macroinstrucțiunile separat de programele precompilate.

5. Nu sînt permise două programe precompilate cu același nume.

6. Nu sînt permise nume de macroinstrucțiuni sau de programe precompilate identice cu numele caracteristicilor din structură.

7. După modificarea unui program precompilat în domeniul PRG (prin ștergere și introducere cu același nume), toate programele precompilate și macroinstrucțiunile care îl apelează, introduse în domeniul PRG înainte modificării, vor beneficia de noua variantă.

8. Ca o măsură suplimentară de securitate, programul bibliotecar inițiază automat la începutul și la sfîrșitul unei catalogări, o cerere de creare a unui punct de securitate. Cînd



condițiile specificate la deschiderea sesiunii sînt îndeplinite, se creează punctul de securitate. În caz de incident, folosind utilitarul RSS, se poate restaura domeniul PRG în ultimul punct de securitate.

După restaurare trebuie cerută lista macrourilor și a precompilatelor catalogate, pentru a vedea care este ultimul macro sau precompilat introdus în domeniul PRG. Reluarea catalogării se face de la ultimul macro sau program introdus.

În cazul lucrului fără fișier de securitate, cererea de creare a punctului de securitate este inefectivă.

## Ștergerea macroinstrucțiunilor

Ștergerea din baza de date a unei macroinstrucțiuni se face cu următoarea comandă :

```
:SUPMAC nume macro ?
```

unde :

- :SUPMAC — este cuvîntul cheie indicînd comanda de ștergere.
- nume macro — are semnificația cunoscută.

Prin această comandă se șterge numele macroinstrucțiunii din domeniul DIC iar forma intermediară din domeniul PRG se invalidează. În urma ștergerii nu se eliberează spațiul virtual dar se poate elibera în anumite situații spațiul fizic.

Dacă ștergerea a decurs normal este tipărit mesajul :

```
MACRO <nume macro> STERS
```

## Ștergerea programelor precompilate

Ștergerea unui program precompilat din baza de date se face cu următoarea comandă :

```
:SUPPRO nume program ?
```

unde :

- :SUPPRO — este cuvîntul cheie ce indică comanda de ștergere.
- nume program — are semnificația cunoscută.

În urma acestei comenzi se șterge numele precompilatului din domeniul DIC și codul intermediar din domeniul PRG. Prin ștergere nu se eliberează spațiul virtual ci doar spațiul fizic ocupat de program.

### Comentariu

Dat fiind că spațiul virtual nu se eliberează în urma ștergerilor de macroinstrucțiuni și programe precompilate se poate ajunge, prin inserări și ștergeri repetate, la situația de saturare a spațiului virtual. În acest caz se recomandă folosirea utilitarului de salvare-restaurare care are ca funcțiune și compactarea domeniilor salvate.

## Listarea macrodefinițiilor

Această funcție realizează listarea numelor tuturor macroinstrucțiunilor din baza de date împreună cu informații privind lungimea (în caractere) și data inserării lor (an, lună, zi, oră, minut).

Sintaxa comenzii de listare este :

:LISMAC ?

## Listarea programelor precompilate

Se realizează listarea numelor tuturor programelor precompilate din baza de date și informații cu privire la lungime (în număr de pagini de 500 octeți) și data inserării (an, lună, zi, oră, minut).

Sintaxa comenzii este următoarea :

:LISPRO ?

## Listarea completă

Oferă informațiile cumulate de la listarea macroinstrucțiunilor și a programelor precompilate precum și informații cu caracter general ca :

- data creării bibliotecii ;
- data ultimei inserări ;
- numărul total de macroinstrucțiuni existente în bază ;
- numărul total de programe precompilate existente în bază ;

Sintaxa acestei comenzi este :

:LISALL ?

## 5. INTERFEȚE DE COMUNICARE CU ALTE LIMBAJE (Cu exemple de programe SOCRATE, COBOL, FORTRAN, PASCAL, BASIC)

### Necesitate, obiective

Limbajul de manipulare SOCRATE a fost conceput în special pentru prelucrări organizate în mod conversațional, această caracteristică impunând necesitatea unuia sau mai multor tipuri de acces care să permită, suplimentar, prelucrări greu de realizat în mod conversațional datorită fie duratei prelucrării, fie volumului de informații în intrare/ieșire, fie provenienței sau destinației lor.

Pe de altă parte chiar în timpul unei prelucrări în mod conversațional se poate ivi necesitatea rezolvării unor probleme care să se soluționeze mult mai facil într-un alt limbaj de programare.

În același timp s-a considerat utilă asigurarea unei posibilități de comunicare din interiorul unui program SOCRATE cu interpretorul liniilor de comandă (CLI) — în cazul sistemelor compatibile RSX-11M(MCR...) — pentru a oferi o nouă fereastră spre sistemul de operare.

Sintetizînd cele de mai sus s-au realizat următoarele trei interfețe :

— Interfața care permite apelarea unor subprograme SOCRATE catalogate din limbajele evaluate care respectă convenția RSX-11M privitor la parametrii de apel al unui subprogram (în RS adresa tabelii parametrilor de acces) COBOL, FORTRAN, PASCAL, BASIC-PLUS-2, MAC etc.

— Interfața care permite apelarea unui subprogram scris într-un limbaj ce respectă condițiile de la punctul anterior, din interiorul unui program de cereri SOCRATE.

— Interfața cu interpretorul liniilor de comandă (MCR...) al sistemului de operare.

## Interfața Limbaje evaluate — SOCRATE-MINI

Este vorba pe de o parte de a defini prin intermediul unor primitive analoge celor de gestiune a fișierelor clasice un acces logic la baza de date, (aceste primitive trebuie să permită accesul la toate caracteristicile elementare ale bazei urmînd orice relație definită prin structură), pe de altă parte de a defini o metodă de acces ușor de utilizat din limbajele evaluate COBOL, FORTRAN, PASCAL, BASIC-PLUS-2 sau din MAC.

În continuare vom prezenta exemple de subprograme SOCRATE-MINI apelate din COBOL și FORTRAN, PASCAL și BASIC considerînd ca apelarea din celelalte limbaje se va putea înțelege destul de ușor avînd exemplele de plecare din COBOL și FORTRAN.

Lucrul cu o bază de date de tip SOCRATE-MINI efectuat prin intermediul unui limbaj evoluat se desfășoară respectîndu-se toate condițiile și facilitățile lucrului autonom :

— este necesară, anterior prelucrării, deschiderea sesiunii de către LGI și închiderea ei după terminarea prelucrării de către LGO (lucru posibil de efectuat atît în mod independent de program, cît și din programul scris în limbaj evoluat, prin interfața cu MCR...);

— sînt verificate drepturile de acces ale utilizatorului;

— dacă subprogramele SOCRATE au instrucțiuni specifice creerii punctelor de reluare, iar sesiunea a fost deschisă cu fișier de securitate se crează puncte de reluare pe parcursul prelucrării;

— programele scrise în limbaje evaluate pot fi abordate numai cu ABS (este interzisă utilizarea comenzii ABO pentru programe cu interfața SOCRATE);

— modificările de structură ale bazei sau recatalogarea subprogramului SOCRATE nu necesită reeditarea de legături pentru programul scris în limbaj evoluat;

— pe întreg parcursul prelucrării (între deschiderea accesului la baza de date — SOPEN — și închiderea accesului la baza de date — SCLOSE) variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$  sînt globale.

## Primitivele de legătură cu baza de date

### Deschiderea bazei de date

Primitiva de deschidere a bazei de date asigură apelarea interfeței cu baza de date, verificarea drepturilor de acces la bază, a faptului dacă sesiunea SOCRATE-MINI este deschisă, extinzând procesorul cu dimensiunile zonelor de buffere utilizator necesare și punând pe nedefinit variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$ .

A nu se confunda această funcțiune cu deschiderea sesiunii SOCRATE care trebuie făcută anterior ei prin intermediul utilitarului LGI.

În limbajul COBOL deschiderea bazei de date se face astfel :

CALL „SOPEN“ USING Art-open cod-eroare  
unde :

77 COD-EROARE PIC S(9(4) COMP.

01 ART-OPEN.

02 DSB PIC 9(4) COMP.

02 DSR PIC 9(4) COMP.

02 LNGMAX PIC 9(4) COMP.

02 NUME-PROCESOR PIC X(6).

DSB — dimensiunea bufferelor fișierelor utilizator (analog Switch DSB de la apelarea procesorului SOC) ;

DSR — dimensiunea bufferelor de date utilizator (analog Switch DSR de la apelarea procesorului SOC) ;

LNGMAX — dimensiunea maximă a unui buffer de date de legătură între limbaj și SOCRATE ;

NUME-PROCESOR — Numele procesorului ce asigură interfața cu baza de date.

Dacă utilizatorul are UIC privilegiat iar pe discul LB: al sistemului se află în UIC-ul [1,55] procesorul LES atunci nu mai este necesară comunicarea numelui procesorului de interfața în câmpul respectiv mîtîndu-se LOW-VALUE (zero binar).

În acest caz primitiva instalează ea procesorul LES dîndu-i numele :

LESnn nn — nr terminalului TI: utilizator.

Dacă utilizatorul nu este privilegiat sau procesorul LES nu se află pe LB: [1,55], este necesară instalarea sa anterioară și în acest caz NUME-PROCESOR se mută anterior apelării primitivei numele atribuit la instalare (...LES valoare implicită).

Din FORTRAN deschiderea unei baze de date se face astfel :

CALL SOPEN (Dsb, coderr)

Anterior sînt necesare următoarele declarații :

INTEGER CODERR

INTEGER DSB, DSR, LNGMAX, NUMPRO(3)

COMMON /SOCBUF/ DSB, DSR, LNGMAX, NUMPRO

DSB, DSR, LNGMAX avînd aceeași semnificație cu cîmpurile similare din COBOL :

— CODERR este analog cu COD-EROARE

— NUMPRO este analog cu NUME-PROCESOR

Instrucțiunea COMMON s-a folosit doar pentru a asigura aranjarea datelor în memorie în ordinea dorită. Pentru alte limbaje apelul se va face conform sintaxei limbajului respectiv, avînd grijă ca datele DSB, DSR, LNGMAX, NUME-PROCESOR să ocupe o zonă de memorie contiguuă iar DSB, DSR, LNGMAX, COD-EROARE să fie de tip binar pe un cuvînt.

**Accesul la baza de date.**

Primitiva de acces la baza de date permite :

- apelarea unui subprogram SOCRATE catalogat anterior în baza de date ;
- transmiterea către acesta a unui buffer de date care va fi dus în BUFFER 10, care este bufferul de legătură cu alte limbaje ;
- retransmiterea rezultatelor prelucrării, obținute prin apelarea subprogramului SOCRATE, programului apelant.

Apelul din COBOL al primitivei se face astfel :

```
CALL „SGBD“ USING Art-SGBD Cod-eroare
```

unde :

```
77 COD-EROARE PIC S9(4) COMP.
```

```
01 ART-SGBD.
```

```
02 NUME-PRECOMPILAT PIC X(24).
```

```
02 LUNG-BUFFER PIC 9(4) COMP.
```

```
02 BUFFER-DATE.
```

```
03.....
```

- COD-EROARE are aceeași semnificație ca la primitiva SOPEN ;
- NUME-PRECOMPILAT – numele subprogramului SOCRATE de apelat avînd cel mult 24 de caractere RADIX-50 ;
- LUNG-BUFFER – lungimea bufferului de date atașat precompilatului, acest număr trebuie să se dividă la 26 și să fie mai mic sau egal cu LNGMAX de la primitiva SOPEN ; dacă este 0 înseamnă că nu atașăm nici un buffer de legătură între programul apelant și subprogramul SOCRATE ;
- BUFFER-DATE – este bufferul de date de legătură cu subprogramul SOCRATE, tipurile de date posibil de introdus în acest buffer sînt DISPLAY și COMP.

Apelul în FORTRAN al primitivei va avea următoarea formă :

```
CALL SGBD (Nmprec, Coderr)
```

Anterior sînt necesare următoarele declarații :

```
INTEGER CODERR
```

```
INTEGER NMPREC (12), LNGBFD
```

```
COMMON /SOCBUF/NMPREC, LNGBFD, Date-din-bufferul-de-date
```

unde :

- CODERR este analog COD-EROARE de la apelarea din COBOL a primitivei ;
- NMPREC – similar lui NUME-PRECOMPILAT ;
- LNGBFD – similar lui LUNG-BUFFER ;

– Date-din-bufferul-de-date fiind datele elementare.

Tipurile de date posibil de introdus în bufferul de interfață cu subprogramul SOCRATE sînt :

```
– INTEGER*2
```

```
– INTEGER*4
```

```
INTEGER*8
```

Ca fișier de ieșire pentru subprogramul SOCRATE este considerat TI:-ul utilizatorului.

Variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$  sînt variabile globale între deschiderea și închiderea bazei.

Pentru a transmite valori ale variabilelor  $Y_1$ ,  $Z_1$ ,  $W_1$  programului apelant este necesară mutarea lor în cîmpuri din bufferul de interfață.

Declarația COMMON are același rol ca și la primitiva anterioară.

Subprogramele SOCRATE apelabile din programe scrise în alte limbaje sînt programe precompilate obișnuite catalogate cu procesorul MGS care folosesc ca buffer de interfață BUFFER 10 pe care nu au voie să-l elibereze pe parcursul execuției lor.

Este deci necesar să conțină o instrucțiune :

M  $X_k$  = FORMAL Buff-date DANS BUFFER 10

și nu poate conține instrucțiuni de forma :

M  $X_k$  = U sau D  $X_k$ ,  $X_k$  fiind  $X$ -ul atribuit lui BUFFER 10.

Observație :

Dacă nu este necesară existența unui buffer de legătură, atunci aceste ultime indicații devin fără obiect.

### Inchiderea bazei de date

Primitiva de închidere a bazei de date asigură terminarea normală a procesorului de interfață cu baza de date.

Apelul din COBOL este :

CALL „SCLOSE“ USING COD-EROARE

din FORTRAN apelul are forma :

CALL SCLOSE (CODERR)

COD-EROARE și CODERR avînd forma și semnificația de la celelalte primitive.

### Tratarea erorilor

Codurile de eroare furnizate în COD-EROARE sau CODERR în urma apelului uneia din primitive sînt următoarele :

- 1 — operația s-a desfășurat cu succes ;
- 2 — eroare gravă detectată de procesorul de interfață ;
- 4 — procesorul de interfață s-a terminat anormal datorită unei erori sistem sau abortării lui ;
- 5 — eroare la transmiterea de parametri către primitivă ;
- 6 — procesorul de interfață nu poate fi apelat
- 7 — nume procesor eronat
- 8 — nu s-a putut asigura trimiterea informațiilor de la programul apelant la procesorul de interfață
- 9 — nu s-a putut asigura trimiterea informațiilor de la procesorul de interfață la programul apelant
- 10 — lungime buffer de date eronată
- 11 — nu s-a putut dezinstala procesorul de interfață
- 12 — apel după închiderea bazei de date sau înainte de deschiderea bazei de date
- 13 — nu se poate conecta procesorul de interfață cu programul apelant
- 14 — în subprogramul SOCRATE apelat a fost eliberat BUFFER 10

În afară de aceste coduri de eroare sînt furnizate codurile de eroare tip SOCRATE (vezi Anexa A din Manualul de utilizare) avînd însă valoare negativă.

Erorile recuperabile la apelul primitivei SGBD pot fi tratate în interiorul subprogramului SOCRATE de către utilizator.

Procesorul nu oprește în mod automat programul la întîlnirea unei erori recuperabile și furnizează programul apelant doar codul ultimei erori detectate pe parcursul prelucrării subprogramului.

### Primitiva de legătură cu interpreterul liniilor de comandă

Pentru a mări posibilitățile de prelucrare din interiorul unor programe evaluate se oferă utilizatorilor o primitivă care permite transmiterea unei linii de comandă către MCR..., așteptarea executării prelucrării cerute și comunicarea către programul apelant dacă tranzacția s-a terminat corect.

Apelul primitivei din COBOL are forma :

```
CALL „MCR“ USING linie-comandă cod-eroare
```

unde :

```
77 COD-EROARE PIC S9999 COMP.
```

```
77 LINIE-COMANDĂ PIC X(80).
```

iar apelul din FORTRAN are forma :

```
CALL MCR(LINCMD, CODERR) ]
```

După execuția funcțiunii cerute în linia de comandă se furnizează în COD-EROARE (COD-ERR) următoarele coduri de eroare :

- 0 – avertisment eroare negravă în execuția funcțiunii
- 1 – executarea cu succes a funcțiunii
- 2 – eroare gravă detectată în timpul execuției funcțiunii
- 4 – s-a abortat sau s-a terminat cu o eroare sistem funcțiunea cerută

### Exemple de programe care folosesc interfața cu SOCRATE

Programele, atît cel scris în COBOL cît și cele scris în FORTRAN, PASCAL, BASIC citesc un fișier de date și încarcă datele în baza de date.

După terminarea încărcării se execută un subprogram SOCRATE care afișează la T1: numărul angajaților încărcăți și repune baza de date în starea anterioară apelului.

Vor fi deci două programe SOCRATE : ÎNCARC-UN-ANGAJAT și AFIȘARE

```
:DEFPRO ÎNCARC-UN-ANGAJAT
:EXP
M X1 = FORMAL F-ANGAJAT-1 DANS BUFFER 10
SI Y1=U ALORS M Y1=1 SINON M Y1=Y1+1 FIN
G UN ANGAJAT X2 Y1
M NUME DE X2=NUME DE X1
M FUNCTIE DE X2=FUNCTIE DE X1
M SEX DE X2=SEX DE X1
SI STARE-CIVILĂ DE X1='C' ALORS M STARE-CIVILĂ DE X2='CĂSĂTORIT'
```

SINON

SI STARE-CIVILĂ DE X1='N' ALORS M STARE-CIVILĂ DE X2='NECĂSĂTORIT'  
 SINON

SI STARE-CIVILĂ DE X1='D' ALORS M STARE-CIVILĂ DE X2='DIVORȚAT'  
 SINON M STARE-CIVILĂ DE X2='VADUV'

FIN

FIN

FIN

M STARE-POLITICĂ DE X2=STARE-POLITICĂ DE X1

M TALIE DE X2 = TALIE DE X1

M SALARIU DE X2 = SALARIU DE X1

M MARCA DE X2 = MARCA DE X1

M RATE-SPORURI DE X2=RATE-SPORURI DE X1

SI FUNCTIE DE X2 = 'ECONOMIST' OU FUNCȚIE DE X2 = 'CONTABIL' ALORS  
 G UN TESA = X2

FIN

M BINAR DE X1 = TALIE DE X2

?

:FDEF ?

:DEFPRO AFISARE

:EXP

I (1) 'NUMĂR ANGAJATI=' I (+0) Y1 ECRIRE

SE TOUT TESA

S TOUT ANGAJAT ?

:FDEF ?

#### Exemplu de program apelant scris în COBOL

IDENTIFICATION DIVISION.

PROGRAM-ID. INCARC.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT FANGAJAT1 ASSIGN TO „SY:[1,100]FANGAJAT1.DAT“.

DATA DIVISION.

FILE SECTION.

FD FANGAJAT1 LABEL RECORD STANDARD.

01 ART-ANGAJAT PIC X(80).

WORKING-STORAGE SECTION.

77 COD-EROARE PIC S9(4) COMP.

77 COD-ER-AF PIC +9(4).

77 BINAR-AF PIC +(6).99.

01 ART-OPEN.

02 DSB PIC 9(4) COMP VALUE 0.

02 DSR PIC 9(4) COMP VALUE 104.

02 LNGMAX PIC 9(4) COMP VALUE 104.

02 NUME-PROCESOR PIC X(6) VALUE LOW-VALUE.

01 ART-SGBD.

02 NUME-PRECOMPILAT PIC X(24) VALUE „INCARC-UN-ANGAJAT“.

02 LUNG-BUFFER PIC 9(4) COMP VALUE 104.

02 BUFFER-DATE.

03 NUME PIC X(30)

03 MARCA PIC 9(4).

03 FUNCȚIE PIC X(10).

03 SEX PIC X.

03 STARE-POLITICĂ PIC XXX.

03 STARE-CIVILĂ PIC X.

03 TALIE PIC 99V99.

03 SALARIU PIC 9(4).



```

03 RATE-SPORURI PIC 9(4).
03 FILLER PIC X(15).
03 BINAR PIC 9(6)V99 COMP.
03 FILLER PIC X(24).
01 LINIE-COMANDA PIC X(80) VALUE „PIP /FR“.
PROCEDURE DIVISION.
A0.
  CALL „SOPEN“ USING ART-OPEN COD-EROARE
  IF COD-EROARE NOT = 1 GO TO A3.
  OPEN INPUT FANGAJAT1.
A1.
  READ FANGAJAT1 AT END GO TO A2.
  MOVE ART-ANGAJAT TO BUFFER-DATE
  IF NUME = „EOF“ GO TO A1.
  CALL „SGBD“ USING ART-SGBD COD-EROARE
  IF COD-EROARE NOT = 1 GO TO A3.
  MOVE BINAR TO BINAR-AF
  DISPLAY NUME „ “ BINAR-AF GO TO A1.
A2.
  CLOSE FANGAJAT1
  MOVE „AFISARE“ TO NUME-PRECOMPILAT
  MOVE 0 TO LUNG-BUFFER
  CALL „SGBD“ USING ART-SGBD COD-EROARE
  IF COD-EROARE NOT = 1 GO TO A3.
  CALL „SCLOSE“ USING COD-EROARE
  IF COD-EROARE = 1 GO TO A4.
A3.
  MOVE COD-EROARE TO COD-ER-AF
  DISPLAY „COD-ERR“ COD-ER-AF.
A4.
  CALL „MCR“ USING LINIE-COMANDĂ COD-EROARE
  STOP RUN.

```

#### Exemplu de program apelant scris in FORTRAN

```

INTEGER NMPRE1(12),NMPRE2(12)
INTEGER LINCMD(40), CODERR
INTEGER DSB,DSR, LNGMAX, NUMPRO(3)
INTEGER NMPREC(12), LNGBFD, NUME(15), FUNCT(5)
INTEGER*4 MARCA, FILER(6), BINAR
LOGICAL*1 SEX, STAPOL(3), FILER1(15), STACIV
LOGICAL*1 TALIE(4), SALAR(4), RATESP(4)
COMMON /SOCBUF/DSB, DSR, LNGMAX, NUMPRO
COMMON /SOCBUF/NMPREC, LNGBFD
COMMON /SOCBUF/NUME, MARCA, FUNCT, SEX, STAPOL, STACIV, TALIE
COMMON /SOCBUF/SALAR, RATESP, FILER1, BINAR, FILER2
  DATA NMPRE1/'IN','CA','RC','—U','N—','AN','GA','JA','T'/
  *,' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' /
  DATA LINCMD/'PI','P','/F','R',' ',' ',' ',' ',' ',' ',' ',' ','
  *,' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','
  *,' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','
  *,' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' /
  DATA NMPRE2/'AF','IS','AR','E',' ',' ',' ',' ',' ','
  *,' ',' ',' ',' /
  DSB=0
  DSR=104
  LNGMAX=104
  LNGBFD=104
  NUMPRO(1)=0

```

```

CALL SOPEN(DSB,CODERR)
IF(CODERR.NE.1)GOTO 10
DO 2, I=1, 12
2 NMPREC(I)=NMPRE1(I)
3 READ(1, 1,END=4)NUME, MARCA,FUNCT,SEX,STAPOL,STACIV,TALIE,SALAR,
  *RATESP,FILER1,BINAR
1 FORMAT(15A2,A4,5A2,32A1,I4)
  CALL SGBD(NMPREC,CODERR)
  IF(CODERR.NE.1)GO TO 10
  WRITE(5,6)NUME,BINAR
6 FORMAT (' ',15A2,I8)
  GOTO 3
4 DO 5,I=1, 12
5 NMPREC(I)=NMPRE2(I)
  LNGBFD=0
  CALL SGBD(NMPREC,CODERR)
  IF(CODERR.NE.1)GOTO 10
  CALL SCLOSE(CODERR)
  IF(CODERR.EQ.1)GOTO 20
10 WRITE(5, 7)CODERR
7 FORMAT (' COD-ERR=', I4)
20 CALL MCR(LINCMD,CODERR)
  STOP
  END

```

Exemplu de program apelant scris în PASCAL

Program Incarc ;

Type

Tip-buffer-date =

Record

Nume	: Packed array [1..30] of char ;
Marca	: Packed array [1..4] of char ;
Funcție	: Packed array [1..10] of char ;
Sex	: Char ;
Stare-politică	: Packed array [1..3] of char ;
Stare-civilă	: Char ;
Talie	: Packed array [1..4] of char ;
Salariu	: Packed array [1..4] of char ;
Rate-sporuri	: Packed array [1..4] of char ;
Filler-1	: Array [1..15] of char ;
Binar-l	: Integer ;
Binar-h	: Integer ;

End ;

Tip-art-sgbd =

Record

Nume-precompilat	: Packed array [1..24] of char ;
Lung-buffer	: Integer ;
Buffer-date	: Tip-buffer-date ;
Filler-2	: Array [1..24] of char

End ;

Tip-art-open =

Record

Dsb	: Integer ;
Dsr	: Integer ;
Lngmax	: Integer ;
Nume-procesor	: Array [1..3] of integer

End ;

ch80 = Packed array [1..80] of char ;

```

Var      I,Cod-eroare      : Integer ;
        Linie-comanda     : ch80 ;
        F-angajat-1      : File of tip-buffer-date ;
        Art-sgbd         : Tip-art-sgbd ;
        Art-open         : Tip-art-open ;
Procedure Sopen (Var x : Tip-art-open ; Var Y : Integer) ; NonPascal ;
Procedure Sgbd (Var x : Tip-art-sgbd ; Var y : Integer) ; NonPascal ;
Procedure SClose (Var x : Integer) ; NonPascal ;
Procedure MCR (Var x : ch80 ; Var y : Integer) ; NonPascal ;
Begin
  With art-open do
    Begin
      Dsb:=0 ; Dsr:=0 ; Lngmax:=104 ;
      For l:=1 to 3 do nume-procesor[l]:=0 ;
    End ;
    Sopen (art-open,cod-eroare) ;
    If cod-eroare <> 1
      then
        Writeln ('Cod-err ',cod-eroare:1)
      else
        Begin
          Reset (f-angajat-1,'SY:[1,100]FANGAJAT1,DAT') ;
          With art-sgbd do
            Begin
              nume-precompilat:='ÎNCARC-UN-ANGAJAT  ' ;
              lung-buffer:=104 ;
              While not eof (f-angajat-1) do
                Begin
                  Read (f-angajat-1,buffer-date) ;
                  With buffer-date do
                    If nume <> 'EOF
                      Begin
                        Sgbd (art-sgbd,cod-eroare) ;
                        If cod-eroare <> 1 then Writeln ('Cod-err ',cod-eroare:1)
                        else Writeln (nume,' ',binar-1)
                      End
                    End ;
                  nume-precompilat:='AFIȘARE  ' ;
                  lung-buffer:=0 ;
                  SGBD (art-sgbd,cod-eroare) ;
                  If cod-eroare <> 1 then
                    Writeln ('cod-err ',cod-eroare)
                  End ;
                  SCLOSE(cod-eroare) ;
                  If cod-eroare <> 1 then
                    Writeln ('cod-err ',cod-eroare) ;
                  Write ('comanda mcr ? ') ;
                  Break (output) ;
                  Readln (linie-comanda) ;
                  MCR (linie-comanda,cod-eroare) ;
                  If cod-eroare <> 1 then
                    Writeln ('cod-err ',cod-eroare)
                  End ;
                End ;
              End ;
            End ;
          End ;
        End ;
      End ;
    End ;
  End ;
End.

```

Exemplu de program apelant scris in BASIC

```

100  PROGRAM ÎNCĂRC
102  !
104  ! PROGRAM DE ÎNCĂRCARE A UNEI BAZE DE DATE
106  !          UTILIZIND INTERFAȚA CU
108  !          SOCRATE-MINI

```

```

109      !
110      DIM LINCMD$ 80
111      !
112      ! DESCRIERE BUFFER INTERFAȚA
113      !
114      !
115      !
116      !
117      !
118      !
119      !
120      COM (SOCBUF) DSB%, DSR%, LNGMAX%, NUME. PRO$=6, NUME. PRE$=24, &
          LNGBUF%, BUF$=80
121      !
122      !
123      !
124      ! DESCRIERE FISIER DATE
125      !
126      !
127      !
128      !
129      !
130      MAP (ARFIS) ART$ 80
131      !
132      !
133      !
134      !
135      !
136      !
137      !
138      !
139      !
140      MAP (ARFIS) NUME.$=30, MARCA$=4, FUNCȚIE$=10, SEX$=1, STARE.P$=3, &
          STARE.C$=1, TALIE$=4, SAL$=4, RAT.S$=4, FILL$=15, BIN$=4
141      !
142      !
143      !
144      ! INIȚIALIZĂRI
145      !
146      !
147      !
148      !
149      !
150      DSB%=0
151      !
152      !
153      !
154      !
155      !
156      DSR%, LNGMAX%, LNGBUF%=104%
157      !
158      !
159      !
160      NUME. PRO$=STRING$(6,0)
161      !
162      !
163      !
164      !
165      !
166      !
167      !
168      !
169      !
170      !
171      !
172      !
173      !
174      ! DETASARE TERMINAL
175      !
176      !
177      !
178      !
179      !
180      CALL DETACH
181      !
182      !
183      !
184      ! DESCHIDE SESIUNE SOCRATE
185      !
186      !
187      !
188      !
189      !
190      CALL SOPEN(DSB%, CODER%)
191      !
192      !
193      !
194      !
195      !
196      !
197      !
198      !
199      !
200      IF CODER% #1 THEN 370
201      !
202      !
203      !
204      !
205      OPEN '[1,100]FANGAJAT1' AS FILE 1, SEQUENTIAL, MAP ARFIS, ACCESS READ
206      !
207      !
208      !
209      !
210      NUMEPRE$='ÎNCĂRC-UN-ANGAJAT'
211      !
212      !
213      !
214      !
215      !
216      !
217      !
218      !
219      !
220      GET #1, INVALID 290
221      !
222      !
223      !
224      !
225      !
226      !
227      !
228      !
229      !
230      BUF$=ART$
231      !
232      !
233      !
234      !
235      !
236      !
237      !
238      !
239      !
240      IF SEG$(NUME$, 1, 3)=='EOF' THEN 220
241      !
242      !
243      !
244      ! APEL PRECOMPILAT SOCRATE (ÎNCARC-UN-ANGAJAT)
245      !
246      !
247      !
248      !
249      !
250      !
251      !
252      !
253      !
254      !
255      !
256      !
257      !
258      !
259      !
260      CALL SGBD(NUME. PRE$, CODER%)
261      !
262      !
263      !
264      !
265      !
266      !
267      !
268      !
269      !
270      IF CODER% #1 THEN 370
271      !
272      !
273      !
274      !
275      !
276      !
277      !
278      !
279      !
280      PRINT NUME$; ' '; BIN$
281      !
282      !
283      !
284      !
285      !
286      !
287      !
288      !
289      !
290      GOTO 220
291      !
292      !
293      !
294      !
295      !
296      !
297      !
298      !
299      !
300      RESUME 300
301      !
302      !
303      !
304      !
305      !
306      !
307      !
308      !
309      !
310      CLOSE 1
311      !
312      !
313      !
314      !
315      !
316      !
317      !
318      !
319      !
320      NUME. PRE$='AFIȘARE'
321      !
322      !
323      !
324      !
325      !
326      !
327      !
328      !
329      !
330      LNGBUF%=0
331      !
332      !
333      !
334      !
335      !
336      !
337      !
338      !
339      !
340      !
341      !
342      !
343      !
344      ! APEL PRECOMPILAT SOCRATE (AFIȘARE)
345      !
346      !
347      !
348      !
349      !
350      !
351      !
352      !
353      !
354      !
355      !
356      !
357      !
358      !
359      !
360      !
361      !
362      !
363      !
364      !
365      !
366      !
367      !
368      !
369      !
370      CALL SGBD(NUME. PRE$, CODER%)
371      !
372      !
373      !
374      !
375      !
376      !
377      !
378      !
379      !
380      IF CODER% #1 THEN 370
381      !
382      !
383      !
384      !
385      !
386      !
387      !
388      !
389      !
390      ! APEL TERMINARE SESIUNE SOCRATE
391      !
392      !
393      !
394      !
395      !
396      !
397      !
398      !
399      !
400      !
401      !
402      !
403      !
404      !
405      !
406      !
407      !
408      !
409      !
410      !
411      !
412      !
413      !
414      !
415      !
416      !
417      !
418      !
419      !
420      !
421      !
422      !
423      !
424      !
425      !
426      !
427      !
428      !
429      !
430      !
431      !
432      !
433      !
434      !
435      !
436      !
437      !
438      !
439      !
440      !
441      !
442      !
443      !
444      !
445      !
446      !
447      !
448      !
449      !
450      !
451      !
452      !
453      !
454      !
455      !
456      !
457      !
458      !
459      !
460      !
461      !
462      !
463      !
464      !
465      !
466      !
467      !
468      !
469      !
470      !
471      !
472      !
473      !
474      !
475      !
476      !
477      !
478      !
479      !
480      !
481      !
482      !
483      !
484      !
485      !
486      !
487      !
488      !
489      !
490      !
491      !
492      !
493      !
494      !
495      !
496      !
497      !
498      !
499      !
500      !
501      !
502      !
503      !
504      !
505      !
506      !
507      !
508      !
509      !
510      !
511      !
512      !
513      !
514      !
515      !
516      !
517      !
518      !
519      !
520      !
521      !
522      !
523      !
524      !
525      !
526      !
527      !
528      !
529      !
530      !
531      !
532      !
533      !
534      !
535      !
536      !
537      !
538      !
539      !
540      !
541      !
542      !
543      !
544      !
545      !
546      !
547      !
548      !
549      !
550      !
551      !
552      !
553      !
554      !
555      !
556      !
557      !
558      !
559      !
560      !
561      !
562      !
563      !
564      !
565      !
566      !
567      !
568      !
569      !
570      !
571      !
572      !
573      !
574      !
575      !
576      !
577      !
578      !
579      !
580      !
581      !
582      !
583      !
584      !
585      !
586      !
587      !
588      !
589      !
590      !
591      !
592      !
593      !
594      !
595      !
596      !
597      !
598      !
599      !
600      !
601      !
602      !
603      !
604      !
605      !
606      !
607      !
608      !
609      !
610      !
611      !
612      !
613      !
614      !
615      !
616      !
617      !
618      !
619      !
620      !
621      !
622      !
623      !
624      !
625      !
626      !
627      !
628      !
629      !
630      !
631      !
632      !
633      !
634      !
635      !
636      !
637      !
638      !
639      !
640      !
641      !
642      !
643      !
644      !
645      !
646      !
647      !
648      !
649      !
650      !
651      !
652      !
653      !
654      !
655      !
656      !
657      !
658      !
659      !
660      !
661      !
662      !
663      !
664      !
665      !
666      !
667      !
668      !
669      !
670      !
671      !
672      !
673      !
674      !
675      !
676      !
677      !
678      !
679      !
680      !
681      !
682      !
683      !
684      !
685      !
686      !
687      !
688      !
689      !
690      !
691      !
692      !
693      !
694      !
695      !
696      !
697      !
698      !
699      !
700      !
701      !
702      !
703      !
704      !
705      !
706      !
707      !
708      !
709      !
710      !
711      !
712      !
713      !
714      !
715      !
716      !
717      !
718      !
719      !
720      !
721      !
722      !
723      !
724      !
725      !
726      !
727      !
728      !
729      !
730      !
731      !
732      !
733      !
734      !
735      !
736      !
737      !
738      !
739      !
740      !
741      !
742      !
743      !
744      !
745      !
746      !
747      !
748      !
749      !
750      !
751      !
752      !
753      !
754      !
755      !
756      !
757      !
758      !
759      !
760      !
761      !
762      !
763      !
764      !
765      !
766      !
767      !
768      !
769      !
770      !
771      !
772      !
773      !
774      !
775      !
776      !
777      !
778      !
779      !
780      !
781      !
782      !
783      !
784      !
785      !
786      !
787      !
788      !
789      !
790      !
791      !
792      !
793      !
794      !
795      !
796      !
797      !
798      !
799      !
800      !
801      !
802      !
803      !
804      !
805      !
806      !
807      !
808      !
809      !
810      !
811      !
812      !
813      !
814      !
815      !
816      !
817      !
818      !
819      !
820      !
821      !
822      !
823      !
824      !
825      !
826      !
827      !
828      !
829      !
830      !
831      !
832      !
833      !
834      !
835      !
836      !
837      !
838      !
839      !
840      !
841      !
842      !
843      !
844      !
845      !
846      !
847      !
848      !
849      !
850      !
851      !
852      !
853      !
854      !
855      !
856      !
857      !
858      !
859      !
860      !
861      !
862      !
863      !
864      !
865      !
866      !
867      !
868      !
869      !
870      !
871      !
872      !
873      !
874      !
875      !
876      !
877      !
878      !
879      !
880      !
881      !
882      !
883      !
884      !
885      !
886      !
887      !
888      !
889      !
890      !
891      !
892      !
893      !
894      !
895      !
896      !
897      !
898      !
899      !
900      !
901      !
902      !
903      !
904      !
905      !
906      !
907      !
908      !
909      !
910      !
911      !
912      !
913      !
914      !
915      !
916      !
917      !
918      !
919      !
920      !
921      !
922      !
923      !
924      !
925      !
926      !
927      !
928      !
929      !
930      !
931      !
932      !
933      !
934      !
935      !
936      !
937      !
938      !
939      !
940      !
941      !
942      !
943      !
944      !
945      !
946      !
947      !
948      !
949      !
950      !
951      !
952      !
953      !
954      !
955      !
956      !
957      !
958      !
959      !
960      !
961      !
962      !
963      !
964      !
965      !
966      !
967      !
968      !
969      !
970      !
971      !
972      !
973      !
974      !
975      !
976      !
977      !
978      !
979      !
980      !
981      !
982      !
983      !
984      !
985      !
986      !
987      !
988      !
989      !
990      !
991      !
992      !
993      !
994      !
995      !
996      !
997      !
998      !
999      !
1000      !

```

## Interfața SOCRATE-MINI – Limbaje evaluate

Apelul unui subprogram scris în alt limbaj de programare se face din SOCRATE prin intermediul instrucțiunii APPEL, a cărei sintaxă generală este :

$$\text{APPEL Nume-subprogram} \left[ \text{AVEC} \left\{ \begin{array}{l} \text{Constantă Z} \\ Z_1 \\ \text{BUFFER } n \end{array} \right\} \right]$$

unde :

- Nume-subprogram – este numele subprogramului în alt limbaj atribuit la editarea de legături avînd maxim 6 caractere RADIX-50
- constanta Z,  $Z_1$  sau BUFFER n sînt parametrii de apel.

Peste BUFFER n poate fi suprapus un *formal* avînd date de tip :  
MOT, BINAIRE, DILATE.

Subprogramul în alt limbaj respectă toate caracteristicile impuse de limbajul respectiv pentru un subprogram, în plus numele său va fi în mod obligatoriu RUTINA, urmînd ca la editarea de legături să i se atribuie adevăratul său nume.

Numărul de parametri formali ai subprogramului este  $\leq 1$ , dar din COBOL transmițîndu-se un cîmp grupat se pot transmite de fapt oricîți parametri se doresc.

Pentru FORTRAN există posibilitatea primirii parametrilor prin intermediul unui bloc de COMMON dacă în subprogram există o declarație COMMON/BUFSOC/Parametri-de-apel

- BUFSOC – este numele predefinit al blocului COMMON și deci este obligatoriu de respectat.

Pentru MAC datele pot fi obținute fie de la adresa din tabelă cu parametrii de apel, a cărei adresă este conținută de registrul R5 fie de la adresa BUFSOC, fie descriind o secțiune reacoperibilă BUFSOC.

Pentru alte limbaje se va adopta una din căile prezentate mai sus funcție de specificul limbajului respectiv.

### Exemplu de subprogram COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. RUTINA.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
DATA DIVISION.
LINKAGE SECTION.
```

```
01 ART.
```

```
02 NUMAR PIC 9(4) COMP.
```

```
02 RĂDĂCINA PIC 99V99 COMP.
```

```
PROCEDURE DIVISION USING ART.
```

```
A0.
```

```
COMPUTE RĂDĂCINA ROUNDED = NUMĂR ** 3 / 1357.
```

```
A1.
```

```
EXIT PROGRAM.
```

## Exemplu de subprogram FORTRAN

```

SUBROUTINE RUTINA
INTEGER RADACI
COMMON /BUFSOC/NUMAR, RADACI
RADACI=(NUMAR*1000000.):** (1,3.)
RETURN
END

```

## Programul apelant SOCRATE

În limbajul de descriere avem următoarele :

```

D
FORMAL ART
DEBUT
NUMAR BINAIRE 4
RADACINA BINAIRE 2 V 2
FIN
FIN?

```

Programul în limbajul de manipulare este următorul :

```

M X1=FORMAL ART DANS BUFFER 1
M NUMAR DE X1=28
APPEL EXTRAD AVEC BUFFER 1
I RADACINA DE X1?

```

## Interfața SOCRATE-MINI – Interpreterul liniilor de comandă

Suplimentar față de cele două interfețe prezentate pentru limbajul SOCRATE-MINI s-a realizat și interfața cu interpreterul liniilor de comandă (MCR...) prin care se transmite o linie de comandă MCR-ului, se așteaptă executarea funcțiunii dorite și se comunică dacă funcțiunea s-a executat corect.

Interfața se asigură prin intermediul instrucțiunii :

$$\text{APPEL MCR} \left[ \text{AVEC} \left\{ \begin{array}{l} \text{Constanta Z} \\ Z_1 \\ \text{BUFFER } n \end{array} \right\} \right]$$

În continuare vom prezenta un program SOCRATE-MINI care folosind această interfață sortează un fișier de lucru.

```

ATTACHE BANDE 1 A 'MN.DAT' AVEC 'O,512,80,V'
M X1=FORMAL F-ANGAJAT-1 DANS BUFFER 1
POUR TOUT ANGAJAT X2 AVEC FONCTIE = 'PROGRAMAT.' ;
  M NUME DE X1=NUME DE X2
  M SALARIU DE X1=SALARIU DE X2
  ECRIRE BUFFER 1 DANS BANDE 1
FIN

```

```

DETACHE BANDE 1
ATTACHE BANDE 1 A 'MN.DAT' AVEC 'I,512,80,V'
APPEL MCR AVEC 'SRT MN =: MN/FO: VAR: 80,KE: CN1.30'
M X1=FORMAL F-ANGAJAT-1 DANS BUFFER 1
FAIRE
  LIRE BANDE 1 DANS BUFFER 1
  SI ERREUR = 1 ALORS SORTIE FIN
  I(1) NUME DE X1 I (30) SALARIU DE X1 ECRIRE
  REFAIRE
FIN
APPEL MCR AVEC 'PIP MN.DAT; *;DE'
?
```

Analog acestui exemplu pot fi transmise MCR-ului toate tipurile de comenzi permise de sistemul de operare.

## 6. INTERFAȚĂ PENTRU LUCRUL ÎNTR-O REȚEA DE BAZE DE DATE (cu exemple de programe în SOCRATE, COBOL, FORTRAN)

### Necesitate, obiective

Odată cu sporirea parcului de minicalculatoare, cu dezvoltarea unor aplicații din ce în ce mai complexe, aplicații care necesită prelucrări de date atât locale cât și distribuite pe largi spații geografice, a apărut necesitatea de a proiecta aplicații care să funcționeze într-o rețea de minicalculatoare.

Cu atât mai stringent se pune problema pentru SGBD-uri care prin natura lor presupun prelucrări de volume mari de date cu interconectări multiple și complexe.

Față de natura problemelor proprii economiei noastre, ținând cont și de caracteristicile tehnice ale rețelelor posibil de implementat în prezent, s-a considerat ca un prim pas spre realizarea unei baze de date distribuite să se realizeze o interfață care să permită regăsirea datelor din baze de date SOCRATE-MINI distribuite într-o rețea de minicalculatoare.

Această interfață este gândită asemănător cu interfața dintre limbajele evaluate și SOCRATE-MINI, deci se vor apela subprograme SOCRATE, anterior fiind necesară deschiderea bazei de date respective, iar la terminarea lucrului cu acea bază se impune închiderea ei.

Subprogramele apelate sînt programe SOCRATE-MINI catalogate în baza respectivă, datele de interfață dintre programul apelant și cel apelat respectînd aceleași convenții ca la interfața Limbaje evaluate — SOCRATE-MINI.

Se poate lucra cu 10 baze de date deschise simultan și deoarece bazele sînt implementate pe minicalculatoare diferite, modul de apel al unui subprogram SOCRATE-MINI a fost conceput astfel încît să permită executarea simultană a mai multor subprograme din baze de date diferite.

Ținând cont de diversitatea de situații în care este necesară o astfel de legătură, interfața a fost gândită să poată fi folosită atât din SOCRATE-MINI cât și din alt limbaj care respectă convențiile de transmitere a parametrilor unui subprogram (adresa tabelii de adrese în R5).

Dintre limbajele de nivel înalt din care poate fi apelată această interfață amintim : COBOL, FORTRAN, PASCAL, BASIC, MAC etc.

Bazele de date legate printr-o astfel de interfață sînt *baze de date eterogene total independente*, care pot fi exploatare în mod local chiar și în paralel cu execuția unui subprogram SOCRATE-MINI cerută dintr-un alt nod al rețelei.

Pentru a asigura această interfață trebuie definit, prin intermediul unor primitive analoge celor de la interfața Limbaje evoluat — SOCRATE-MINI, un acces logic la o bază de date aflată într-un alt nod al unei rețele de minicalculatoare.

Aceste primitive trebuie să permită accesul la toate caracteristicile elementare ale bazei urmînd orice relație definită prin structură.

Pe de altă parte se impune definirea unei metode de acces ușor de utilizat din SOCRATE-MINI, COBOL, FORTRAN sau alt limbaj de nivel înalt.

## Interfața apelată din limbajul de manipulare

Lucrul cu o bază de date SOCRATE-MINI aflată în alt nod al unei rețele se desfășoară respectîndu-se toate condițiile și facilitățile lucrului autonom :

- este necesară, anterior prelucrării, deschiderea sesiunii de către LGI și închiderea ei după terminarea prelucrării cu ajutorul lui LGO (în toate nodurile de rețea conectate prin intermediul acestei interfețe) ;

- la conectarea cu un anumit nod se verifică drepturile de acces ale utilizatorului care face conectarea, de către sistemul de operare (utilizatorul există în ACNT și are parola comunicată), în plus sînt verificate de către SGBD drepturile de acces la bază ale aceluiași utilizator ;

- dacă subprogramele SOCRATE-MINI au instrucțiuni specifice creării de puncte de reluare, iar sesiunea a fost deschisă cu fișier de securitate, se creează puncte de reluare pe parcursul prelucrării ;

- modificările de structură ale bazei sau recatalogarea subprogramului SOCRATE-MINI nu necesită recatalogarea programului apelant ;

- pe întreg parcursul prelucrării (între conectarea bazei NT-CNT și deconectarea ei NT-DSC) variabilele  $X_i$ ,  $Y_i$ ,  $Z_i$ ,  $W_i$  sînt globale.

Observație : Privitor la controlul drepturilor de acces trebuie subliniat faptul că administratorul bazei din nordul destinație este cel care decide dacă se face controlul sau nu de către softul de rețea și dacă dorește în plus și controlul efectuat de SGBD.

• Apelarea unei funcțiuni de interfață cu rețeaua are următoarea formă sintactică :

**APPEL NT-xxx AVEC BUFFER n**

unde : xxx poate lua următoarele valori : CNT, SOC, SYN, SCS, DSC.



Peste BUFFER n este necesară plierea unui FORMAL avînd următoarea structură :

```

FORMAL Nume-formal
DEBUT
Nume-nod MOT 6
Dsb BINAIRE (8 4)
Dsr BINAIRE 4
Lung-max-buff-date BINAIRE 4
Cod-er-w1 BINAIRE 4
Cod-er-w2 BINAIRE 4
Nume-utilizator MOT (28 16)
Parolă MOT (46 8)
Nume-precompilat MOT (56 24)
Lung-buffer-date BINAIRE 4
Date utilizator
.
.
.
.
FIN ?

```

unde :

**Nume-nod** — este format din 1—6 caractere ASCII și reprezintă numele nodului destinație în care se află baza de date ce trebuie conectată ;

**Dsb** — dimensiunea bufferelor fișierelor utilizator (analog switch DSB de la apelare SOC) ;

**Dsr** — dimensiunea bufferelor de date utilizator (analog switch DSR de la apelare SOC) ;

**Lung-max-buff-date** — dimensiunea maximă a unui buffer de date de legătură în timpul conectării la baza respectivă ;

**Cod-er-w1** — primul și al doilea cuvînt de stare prin care se comunică modul în care a decurs prelucrarea ;

**Cod-er-w2** —

**Nume-utilizator** — numele unui utilizator recunoscut de sistemul de operare aflat în nodul destinație, avînd 1—16 caractere ASCII ;

**Parolă** — parola aceluși utilizator (1—8 caractere ASCII) ;

**Nume-precompilat** — numele precompilatului ce se dorește apelat avînd 1—24 caractere RADIX-50 ;

**Lung-buffer-date** — lungimea bufferului de date de legătură atașat precompilatului de mai sus.

Se observă că din acest FORMAL nu sînt folosite următoarele zone : 6—7, 18—27, 44—45, 54—55 necesare pentru lucru intern al interfeței (numărătoarea se face începînd cu 0).

În afară de Nume-precompilat și Lung-buffer-date celelalte cîmpuri prezentate mai sus nu pot fi modificate pe parcursul prelucrării (între NT-CNT și NT-DSC).

Plierea acestui FORMAL în BUFFER n se face prin instrucțiunile :

M X<sub>1</sub>=FORMAL Nume-formal DANS BUFFER n

S X<sub>1</sub>

urmate apoi de instrucțiunile de încărcare a cîmpurilor din FORMAL.

## Primitivele de legătură

### Conectarea unei baze de date

*Primitiva de conectare* cu o bază de date aflată într-o rețea de minicalculatoare asigură apelarea interfeței de legătură, verificarea drepturilor de acces la bază, a faptului că sesiunea SOCRATE-MINI este deschisă, extinzînd procesorul de interfață din nodul destinație cu dimensiunile zonelor de buffere utilizator cerute și punînd pe nedefinit variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$ .

A nu se confunda această funcțiune cu deschiderea sesiunii SOCRATE-MINI care trebuie făcută anterior prin intermediul utilitarului LGI.

Conectarea unei baze de date se face astfel :

### APPEL NT-CNT AVEC BUFFER n

în BUFFER n aflîndu-se un FORMAL de tipul celui prezentat mai sus încărcat cu datele necesare.

Pot fi conectate simultan de către un program SOCRATE-MINI maxim 10 alte baze de date aflate în nodurile unei rețele de minicalculatoare.

### Accesul la o bază de date conectată

*Primitivele de acces* la o bază de date conectată permit :

– apelarea unui subprogram SOCRATE-MINI catalogat anterior în baza de date, această apelare putîndu-se face cu așteptarea terminării prelucrării în nodul destinație (NT-SCS) sau fără așteptarea terminării prelucrării (NT-SOC), în acest caz sincronizarea cu terminarea prelucrării făcîndu-se printr-o primitivă de sincronizare (NT-SYN) ;

– transmiterea către subprogram a unui buffer de date, preluat de acesta în BUFFER 10, care este bufferul de legătură al unui subprogram SOCRATE ;

– retransmiterea rezultatelor prelucrării, obținute prin apelarea subprogramului SOCRATE-MINI, programului apelant.

Ca fișier de ieșire pentru subprogramul SOCRATE-MINI este considerat CO :-ul sistemului din nodul destinație.

Variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$  sînt variabile globale între conectarea și deconectarea bazei.

Pentru a transmite valori ale variabilelor  $Y_1$ ,  $Z_1$ ,  $W_1$  programului apelant este necesară mutarea lor în cîmpuri din formalul suprapus peste bufferul de interfață.

Subprogramele SOCRATE-MINI apelabile din alt nod al unei rețele de minicalculatoare sînt programe precompilate obișnuite catalogate cu procesorul MGS în domeniul PRG al bazei din nodul respectiv, care folosesc ca buffer de interfață BUFFER 10 pe care nu au voie să-l elibereze pe parcursul execuției lor.

Este deci necesar ca un astfel de program să conțină o instrucțiune :

M  $X_k$  = FORMAL Buff-date DANS BUFFER 10

și nu poate conține instrucțiuni de forma :

M  $X_k$  = U sau D  $X_k$ ,  $X_k$  fiind X-ul atribuit lui BUFFER 10

Observație : Dacă nu este necesară existența unui buffer de legătură, atunci aceste ultime indicații devin fără obiect.

*Apelarea unui subprogram SOCRATE-MINI cu sincronizare pe sfârșitul prelucrării*

Dacă necesitățile programului apelant impun așteptarea terminării prelucrării din nodul destinație, se folosește primitivă NT-SCS cu următoarea formă sintactică :

#### **APPEL NT-SCS AVEC BUFFER n**

în BUFFER n fiind încărcate anterior apelului toate datele necesare inclusiv numele precompilatului ce trebuie, apelat și dimensiunea bufferului de date de legătură.

Prin această primitivă se impune oprirea execuției programului apelant pînă la terminarea execuției subprogramului apelat.

*Apelarea unui subprogram SOCRATE-MINI fără sincronizare*

Dacă necesitățile programului apelant nu impun așteptarea terminării prelucrării înainte de trecerea la următoarea instrucțiune, este de preferat ca apelarea unui subprogram SOCRATE-MINI din alt nod al rețelei să se facă fără sincronizare permițînd astfel executarea în paralel (multiprocesarea) a mai multor prelucrări.

Această posibilitate este permisă de primitiva NT-SOC al cărui apel are următoarea formă sintactică :

#### **APPEL NT-SOC AVEC BUFFER n**

Sincronizarea cu terminarea prelucrării din nodul destinație se face folosind primitiva NT-SYN cu forma sintactică de apel :

#### **APPEL NT-SYN AVEC BUFFER n**

De subliniat că între cele două instrucțiuni se pot executa orice alte instrucțiuni, mai puțin instrucțiuni care pot altera cîmpuri din BUFFER n.

Prin folosirea inteligentă a acestei facilități se pot obține cîștiguri spectaculoase de timp.

#### **Deconectarea unei baze de date**

*Primitiva de deconectare* a unei baze de date asigură terminarea normală a procesorului de interfață din nodul destinație, eliberarea resurselor programului apelant pentru o eventuală conectare cu altă bază, precum și închiderea rețelei dacă s-a cerut deconectarea ultimei legături rămasă conectată. De subliniat că se poate lucra cu maxim 10 (zece) baze de date conectate simultan deci dacă dorim să lucrăm cu o a unsprezecea bază de date este necesară deconectarea uneia din cele 10 baze de date conectate anterior.

Apelul acestei primitive se face astfel :

#### **APPEL NT-DSC AVEC BUFFER r**

unde : BUFFER n este un buffer prin intermediul căruia s-a conectat baza de date pe care dorim să o deconectăm.

#### **Tratarea erorilor**

Valorile celor două cuvinte de stare au următoarele semnificații :

1. COD-ER-W1 = 1                   ⇒ Primitiva s-a terminat cu succes.
2. COD-ER-W1 = 0                   ⇒ Eroare detectată de modulele de interfață sau de procesorul de interfață RES din nodul destinație.

- 2.1. COD-ER-W2 = 1 ⇒ Se cere conectarea unei a unsprezecea baze de date simultan.
- 2.2. COD-ER-W2 = 2 ⇒ Cerere de deconectare sau de apel subprogram pentru o bază de date căreia nu i s-a făcut conectarea sau care fusese anterior deconectată.
- 2.3. COD-ER-W2 = 3 ⇒ Cerere de deconectare sau apel subprogram pentru o bază care nu a terminat de prelucrat un subprogram lansat anterior cererii.
- 2.4. COD-ER-W2 = 4 ⇒ A fost dezalocat BUFFER 10 de către o instrucțiune existentă în ultimul subprogram apelat.
- 2.5. COD-ER-W2 = -n ⇒ Eroare tip SOCRATE-MINI  
Privitor la semnificația lui n vezi Anexa ce conține mesajele de eroare.
3. COD-ER-W1 = m ⇒ Eroare detectată de modulele și task-urile proprii softului de rețea, m avînd valori negative.  
Pentru  $m > -100$ , semnificația codurilor de eroare se găsește în Anexa B din manualul I/O DRIVERS pentru DECnet/RXS-11M sau în Anexa C din manualul INTERTASK COMMUNICATION pentru MININET/MIX.  
Pentru  $m \neq -74.(IE.NRJ)$  și  $M \neq -15.(IE.ABO)$  COD-ER-W2 are valoarea 0.  
Pentru  $m = -74.(IE.NRJ)$  și  $m = -15.(IE.ABO)$  COD-ER-W2 = l unde l are semnificația din Anexa A (DISCONNECT OR REJECT REASON CODES) a manualului USER'S GUIDE DECnet/RXS-11M sau Anexa D (DISCONNECT OR REJECT REASON CODES) din manualul INTERTASK COMMUNICATION pentru MININET/MIX.

Erorile recuperabile la apelul unui subprogram (vezi eroarea 2.5) pot fi tratate în interiorul programului SOCRATE-MINI de către utilizator cu ajutorul opțiunii ERREUR.

Procesorul nu oprește în mod automat subprogramul la întâlnirea unei erori recuperabile și furnizează programului apelant doar codul ultimei erori detectate pe parcursul prelucrării.

### Exemplu de program pentru rețea scris în limbajul de manipulare

Programul citește un fișier de date și încarcă în paralel două baze de date aflate în două noduri diferite ale unei rețele de minicalculatoare.

După terminarea încărcării se execută tot în paralel un subprogram SOCRATE-MINI care afișează la CO:-ul sistemului din nodul respectiv numărul de angajați încărcăți și repune baza de date în starea anterioară apelului. Vor fi deci două subprograme SOCRATE-MINI :

INCARC-UN-ANGAJAT și AFISARE care sînt cele prezentate în acest manual.

Formalul F-ANGAJAT-1 are următoarea structură :

FORMAL F-ANGAJAT-1  
DEBUT  
NUME

MARCA	DILATE 4
FUNȚIE	MOT 10
SEX	MOT 1
STARE-POLITICA	MOT 3
STARE-CIVILA	MOT 1
TALIE	DILATE 2 V 2
SALARIU	DILATE 4
RATE-SPORURI	DILATE 4
FILL	MOT 15
BINAR	BINAIRE 6 V 2
FIN	

iar formalele F-NET-ANGAJAT-1 și F-BUFFER-DATE au structura următoare :

```

D
FORMAL F-NET-ANGAJAT-1
DEBUT
    NUME-NOD           MOT 6
    DSB                BINAIRE (8 4)
    DSR                BINAIRE 4
    LUNG-MAX-DATE      BINAIRE 4
    COD-ER-W1          BINAIRE 3
    COD-ER-W2          BINAIRE 3
    NUME-UTILIZATOR    MOT (28 16)
    PAROLA              MOT (46 8)
    NUME-PRECOMPILAT  MOT (56 24)
    LUNG-BUFFER-DATE  BINAIRE 4
/* DATE UTILIZATOR */
    NUME                MOT 30
    MARCA               DILATE 4
    FUNȚIE              MOT 10
    SEX                 MOT 1
    STARE-POLITICA     MOT 3
    STARE-CIVILA       MOT 1
    TALIE               DILATE 2 V 2
    SALARIU             DILATE 4
    RATE-SPORURI       DILATE 4
    FILL                MOT 15
    BINAR               BINAIRE 6 V 2
    BUFFER-DATE        MOT (82 80)
FIN
FORMAL F-BUFFER-DATE
DEBUT
    BUFFER-DATE        MOT 80
FIN FIN ?

```

Programul apelant este următorul :

```

/*
/* P225
/* PROGRAMUL TESTEAZĂ APELUL UNOR SUBPROGRAME SOCRATE AFLATE
/* ÎN DOUA NODURI ALE UNEI REȚELE DE MINICALCULATOARE DE
/* CATRE UN PROGRAM SOCRATE AFLAT ÎNTR-UN ALT NOD AL REȚELEI.
/* PROGRAMUL SOCRATE CITEȘTE FIȘIERUL „FANGAJAT1.DAT” TRANSMIȚÎND
/* DATELE SUBPROGRAMELOR SOCRATE.
/* ACESTE SUBPROGRAME REALIZEAZA ÎNCARCAREA CARACTERSTICILOR
/* ENTITATII „ANGAJAT” DIN CELE DOUA BAZE DE DATE EXISTENTE
/* ÎN NODURILE RESPECTIVE.
/* FOLOSIND FUNCȚIUNILE „NT-SOC” ȘI „NT-SYN” ÎNCARCAREA CELOR
/* DOUA BAZE DE DATE SE FACE ÎN PARALEL.
/*
M_X1=FORMAL F-BUFFER-DATE DANS BUFFER 1
M_X2=FORMAL F-NET-ANGAJAT-1 DANS BUFFER 2

```

```

S X2
M X4=FORMAL F-NET-ANGAJAT-1 DANS BUFFER 3
S X4
NUME-NOD DE X2 = EXT ('NUME NOD 1 : ')
M NUME-UTILIZATOR DE X2 = EXT ('NUME UTILIZATOR 1 : ')
APPEL MCR AVEC 'SET /NOECHO=TI:'
M PAROLA DE X2 = EXT ('PAROLA 1 : ')
APPEL MCR AVEC 'SET /ECHO=TI:'
I ' '
M DSB DE X2 = 0
M DSR DE X2 = 80
M LUNG-MAX-DATE DE X2 = 80
M LUNG-BUFFER-DATE DE X2 = 80
M NUME-NOD DE X4 = EXT ('NUME NOD 2 : ')
M NUME-UTILIZATOR DE X4 = EXT ('NUME UTILIZATOR 2 : ')
APPEL MCR AVEC 'SET /NOECHO=TI:'
M PAROLA DE X4 = EXT ('PAROLA 2 : ')
APPEL MCR AVEC 'SET /ECHO=TI:'
I ' '
M DSB DE X4 = 0
M DSR DE X4 = 80
M LUNG-MAX-DATE DE X4 = 80
M LUNG-BUFFER-DATE DE X4 = 80
/*
/* CONECTEAZĂ PROCESORUL SOCRATE DIN NODUL 1.
/*
/* APPEL NT-CNT AVEC BUFFER 2
SI COD-ER-W1 DE X2 = 1 ALORS
  I (1) 'COD EROARE CONECTARE ' I (+0) NUME-NOD DE X2 I(+1) COD-ER-W1
    DE X2
  I (+0) ' * ' I(+0) COD-ER-W2 DE X2 ECRIRE
FIN
/*
/* CONECTEAZĂ PROCESORUL SOCRATE DIN NODUL 2.
/*
/* APPEL NT-CNT AVEC BUFFER 3
SI COD-ER-W1 DE X4 = 1 ALORS
  I (1) 'COD EROARE CONECTARE ' I(+0) NUME-NOD DE X4 I(+1) COD-ER-W1 DE X4
  I (+0) ' * ' I(+0) COD-ER-W2- DE X4 ECRIRE
FIN
ATTACHE BANDE 1 A 'SY:[1, 100]FANGAJAT1.DAT' AVEC 'I,512,80,V'
M NUME-PRECOMPILAT DE X2 = 'ÎNCARC-UN-ANGAJAT'
M NUME-PRECOMPILAT DE X4 = 'ÎNCARC -UN-ANGAJAT'
FAIRE
  LIRE BANDE 1 DANS BUFFER 1
  SI ERREUR = 1 ALORS SORTIE FIN
  M BUFFER-DATE DE X2 = BUFFER-DATE DE X1
/*
/* APELEAZĂ SUBPROGRAMUL „ÎNCARC-UN-ANGAJAT” DIN NODUL 1.
/*
/* APPEL NT-SOC AVEC BUFFER 2
SI COD-ER-W1 DE X2 = 1 ALORS
  I (1) 'COD EROARE SOC ' I(+0) NUME-NOD DE X2 I(+1) COD-ER-W1 DE X2
  I (+0) ' * ' I(+0) COD-ER-W2 DE X2 ECRIRE
FIN
M BUFFER-DATE DE X4 = BUFFER-DATE DE X1
/*
/* APELEAZA SUBPROGRAMUL „ÎNCARC-UN-ANGAJAT” DIN NODUL 2
/* SINCROIZÎNDU-SE APOI CU TERMINAREA ACESTUIA.
/*
/* APPEL NT-SCS AVEC BUFFER 3
SI COD-ER-W1 DE X4 = 1 ALORS

```

```

    (1) 'COD EROARE SCS' I(+0) NUME-NOD DE X4 I(+1) COD-ER-W1 DE X4
    I (+0) ' * ' I(+0) COD-ER-W2 DE X4 ECRIRE
  SINON
    I (1) 'S-A ADĂUGAT PENTRU ' I(+0) NUME-NOD DE X4
    I (+1) NUME DE X4 I(+0) BINAR DE X4 ECRIRE
  FIN

/*
/* SE SINCRONIZEAZĂ CU TERMINAREA PRELUCRĂRII DIN NODUL 1.
/*

  APPEL NT-SYN AVEC BUFFER 2
  SI COD-ER-W1 DE X2 = 1 ALORS
    I (1) 'COD EROARE SYN ' I(+0) NUME-NOD DE X2 I(+0) COD-ER-W1 DE X2
    I (+0) ' * ' I(+0) COD-ER-W2 DE X2 ECRIRE
  SINON
    I (1) 'S-A ADĂUGAT PENTRU ' I(+0) NUME-NOD DE X2
    I (+1) NUME DE X2 I(+0) BINAR DE X2 ECRIRE
  FIN
  REFAIRE
  FIN
  M LUNG-BUFFER-DATE DE X2 = 0
  M LUNG-BUFFER-DATE DE X4 = 0
  M NUME-PRECOMPILAT DE X4 = 'AFIȘARE'
  M NUME-PRECOMPILAT DE X2 = 'AFIȘARE'
/*
/* APELEAZĂ SUBPROGRAMUL „AFIȘARE” DIN NODUL 1.
/*
  APPEL NT-SOC AVEC BUFFER 2
  SI COD-ER-W1 DE X2 = 1 ALORS
    I(1) 'COD EROARE SOC' I(+0) NUME-NOD DE X2 I(+1) COD-ER-W1 DE X2
    I(+0) ' * ' I(+0) COD-ER-W2 DE X4 ECRIRE
  FIN
/*
/* APELEAZA SUBPROGRAMUL „AFIȘARE” DIN NODUL 2
/* SINCRONIZINDU-SE APOI CU TERMINAREA LUI.
/*
  APPEL NT-SCS AVEC BUFFER 3
  SI COD-ER-W1 DE X4 = 1 ALORS
    I(1) 'COD EROARE SCS ' I(+0) NUME-NOD DE X4 I(+1) COD-ER-W1 DE X4
    I(+0) ' * ' I(+0) COD-ER-W2 DE X4 ECRIRE
  FIN
/*
/* SE SINCRONIZEAZĂ CU TERMINAREA PRELUCRĂRII DIN NODUL 1.
/*
  APPEL NT-SYN AVEC BUFFER 2
  SI COD-ER-W1 DE X2 = 1 ALORS
    I(1) 'COD EROARE SYN ' I(+0) NUME-NOD DE X2 I(+1) COD-ER-W1 DE X2
    I(+0) ' * ' I(+0) COD-ER-W2 DE X2 ECRIRE
  FIN
/*
/* DECONECTEAZĂ LEGATURA CU PROCESORUL SOCRATE DIN NODUL 1.
/*
  APPEL NT-DSC AVEC BUFFER 2
  SI COD-ER-W1 DE X2 = 1 ALORS
    I (1) 'COD EROARE DECONECTARE PENTRU' I (+1) NUME-NOD DE X2
    I (+1) COD-ER-W1 DE X2 I(+0) ' * ' I (+0) COD-ER-W2 DE X2 ECRIRE
  FIN
/*
/* DECONECTEAZĂ LEGĂTURA CU PROCESORUL SOCRATE DIN NODUL 2.
/*

```

```

APPEL NT-DSC AVEC BUFFER 3
SI COD-ER-W1 DE X4 = 1 ALORS
  I(1) 'COD EROARE DECONECTARE PENTRU' I (+1) NUME-NOD DE X4
  I(+1) COD-ER-W1 DE X4 I(+0) ' * ' I (+0) COD-ER-W2 DE X4 ECRIRE
.FIN ?

```

## Interfața apelată din limbaje evolute

Lucrul cu o bază de date SOCRATE-MINI aflată într-un nod al unei rețele, cerut de un program scris într-un limbaj evoluat aflat în alt nod al rețelei, se desfășoară respectând toate condițiile prezentate în manual, condiții referitoare la apelul din SOCRATE-MINI.

Apelarea unei primitive de interfață se face funcție de limbaj, ca apelarea unui subprogram.

De exemplu :

```

CALL "NETxxx" USING ...           din COBOL
CALL NETxxx(...)                 din FORTRAN

```

unde xxx are una din următoarele forme : CNT, SOC, SYN, SCS, DSC.

### Primitivele de legătură

#### Conectarea unei baze de date

Primitiva de conectare cu o bază de date aflată într-un nod al unei rețele de minicalculatoare asigură apelarea interfeței de legătură, verificarea drepturilor de acces la bază, a faptului că sesiunea SOCRATE-MINI este deschisă, extinzând procesorul din nodul destinație cu dimensiunile zonelor de buffere utilizator cerute și punând pe nedefinit variabilele  $X_1$ ,  $Y_1$ ,  $Z_1$ ,  $W_1$ .

A nu se confunda această funcțiune cu deschiderea sesiunii SOCRATE-MINI care trebuie făcută anterior prin intermediul utilitarului LGL.

Apelul din COBOL al primitivei se face astfel :

```
CALL „NETCNT“ USING ART-CNT NR-LEG ART-CRB COD-ER
```

anterior fiind necesare următoarele descrieri de date :

77 NR-LEG	PIC 9(4) COMP.
01 ART-CNT.	
02 DSB	PIC 9(4) COMP.
02 DSR	PIC 9(4) COMP.
02 LNGMAX	PIC 9(4) COMP.
01 ART-CRB.	
02 NUME-NOD	PIC X(6).
02 FILLER	PIC X(22).
02 NUME-UTILIZATOR	PIC X(16).
02 FILLER	PIC XX.
02 PAROLA	PIC X(8).
02 FILLER	PIC X(18).



```
01 COD-ER.
  02 COD-ER-W1          PIC S9(4) COMP.
  02 COD-ER-W2          PIC S9(4) COMP.
```

unde :

DSB — dimensiunea bufferelor utilizator (analog switch DSB de la apelare SOC) ;

DSR — dimensiunea bufferelor de date utilizator (analog switch DSR de la apelare SOC) ;

LNGMAX — dimensiunea maximă a unui buffer de date de legătură între programul apelant și subprogramul SOCRATE-MINI ;

ART-CRB — o zonă de 72 BYTE (CONNECT REQUEST BLOCK) cu informații referitoare la nodul destinație ;

NUME-NOD — numele nodului destinație (1—6 caractere ASCII) ;

NUME-UTILIZATOR — numele unui utilizator recunoscut de sistemul de operare din nodul destinație ;

PAROLA — parola acestui utilizator ;

COD-ER — două cuvinte de stare prin care se comunică modul cum a decurs prelucrarea ;

COD-ER-W1 — primul cuvânt de stare ;

COD-ER-W2 — al doilea cuvânt de stare ;

NR-LEG — numărul bazei conectate (este complectat de modulele de interfață la apelarea lui NETCNT și se folosește pentru identificarea legăturii.

Din FORTRAN conectarea unei baze de date se face astfel :

```
CALL NETCNT(DSB,NRLEG,NUMNOD,CODER)
```

Anterior fiind necesare următoarele declarații :

```
INTEGER DSR,DSB,LNGMAX
INTEGER NUMNOD(3),FILLER1(11),NUMUTI(8),FILER2,PAROLA(4),FILER3(9)
INTEGER CODER(2),NRLEG
COMMON /ARTCNT/DSB,DSR,LNGMAX
COMMON /ARTCRB/NUMNOD,FILER1,NUMUTI,FILER2,PAROLA,FILER3
```

Instrucțiunea COMMON s-a folosit doar pentru a asigura aranjarea datelor în memorie în ordinea dorită.

În cele prezentate mai sus DSB, DSR, CODER, NRLEG, LNGMAX, PAROLA au aceleași semnificații cu omonimele lor din exemplul de apelare din COBOL, NUMNOD fiind similar cu NUME-NOD iar NUMUTI cu NUME-UTILIZATOR.

Pentru alte limbaje, apelul se face conform sintaxei limbajului, avînd grijă ca datele DSB, DSR, LNGMAX respectiv NUMNOD, FILER1, NUMUTI, FILER2, PAROLA, FILER3 să ocupe zone de memorie contiguu, iar DSB, DSR, LNGMAX, CODER(1), CODER(2), NRLEG să fie date de tip binar pe doi octeți.

#### Accesul dintr-un program scris într-un limbaj evoluat la baza de date conectată

Primitivele de acces, dintr-un program scris într-un limbaj de nivel înalt, către o bază conectată permit aceleași facilități și au aceleași restricții ca și primitivele similare din SOCRATE-MINI, toate afirmațiile făcute rămînînd valabile și pentru cazul în care apelarea se face din alt limbaj decît SOCRATE-MINI.

*Apelarea unui subprogram SOCRATE-MINI cu sincronizare pe sfârșitul prelucrării*

Dacă necesitățile programului apelant impun așteptarea terminării prelucrării din nodul destinație, se va apela primitiva NETSCS al cărei apel din COBOL are forma :

CALL „NETSCS“ USING ART-SOC NR-LEG COD-ER  
anterior fiind necesare următoarele descrieri de date :

```
01 ART-SOC.
  02 NUME-PRECOMPILAT      PIC X(24).
  02 LUNG-BUFF-DATE        PIC 9(4) COMP.
  02 Buffer-date.
  .
  .
  .
```

unde :

NR-LEG și COD-ER — au aceeași semnificație ca la primitiva NETCNT ;  
NUME-PRECOMPILAT — numele precompilatului, din nodul destinație, care trebuie executat ;  
LUNG-BUFF-DATE — dimensiunea bufferului de interfață pentru precompilatul apelat ;  
Buffer-date — bufferul de date de interfață al precompilatului.

Din FORTRAN apelarea primitivei se face astfel :

CALL NETSCS(NUMPRE,NRLEG,CODER)  
anterior fiind necesare următoarele instrucțiuni :

```
INTEGER NRLEG,CODER(2),NUMPRE(12),LNGBFD
COMMON /ARTSOC/NUMPRE, LNGBFD,...
```

unde :

NRLEG și CODER — au aceleași semnificații cu omonimele lor de la apelarea din COBOL a primitivei ;  
NUMPRE — similar cu NUME-PRECOMPILAT de mai sus ;  
LNGBFD — analog cu LUNG-BUFF-DATE prezentat anterior.

Instrucțiunea COMMON are același rol ca și pentru primitiva NETCNT.

*Apelarea unui subprogram SOCRATE-MINI fără sincronizare*

Dacă necesitățile programului apelant nu impun așteptarea terminării prelucrării imediat după apelare, este de preferat ca apelul unui subprogram SOCRATE-MINI din alt nod al rețelei să se facă fără sincronizare, permițând astfel executarea în paralel a mai multor prelucrări.

Din COBOL apelarea unui subprogram SOCRATE-MINI fără sincronizare se face astfel :

CALL „NETSOC“ USING ART-SOC NR-LEG COD-ER

↳ Sincronizarea cu terminarea prelucrării din nodul destinație se realizează după cum urmează :

CALL „NETSYN“ USING NR-LEG COD-ER

unde :

ART-SOC, NR-LEG și COD-ER au aceeași semnificație ca și la apelarea primitivei NETSCS

De subliniat că între cele două apeluri se pot executa orice alte instrucțiuni, mai puțin cele care pot altera COD-ER, NR-LEG, ART-SOC, ART-CRB atașate bazei către care s-a făcut apelul.

Din FORTRAN apelarea unui subprogram SOCRATE-MINI se face astfel :

```
CALL NETSOC (NUMPRE,NRLEG,CODER)
```

Sincronizarea cu terminarea prelucrării din nodul destinație se realizează astfel :

```
CALL NETSYN (NRLEG,CODER)
```

unde NUMPRE, NRLEG, și CODER au aceeași formă și aceeași semnificație cu omonimele lor de la apelarea primitivei NETSCS

#### Deconectarea unei baze de date

Primitiva de deconectare apelată dintr-un program scris într-un limbaj de nivel înalt îndeplinește aceleași funcțiuni ca și cea similară ei, apelată din SOCRATE-MINI, respectînd aceleași restricții ca și aceasta.

Apelul din COBOL al primitivei este următorul :

```
CALL „NETDSC“ USING NR-LEG COD-ER
```

unde NR-LEG și COD-ER au semnificații similare cu ale omonimelor lor de la apelarea celorlalte primitive.

Din FORTRAN apelarea se face astfel :

```
CALL NETDSC(NRLEG, CODER)
```

NRLEG și CODER avînd aceeași semnificație ca și la apelarea celorlalte primitive.

#### Tratarea erorilor

Codurile de eroare pentru interfața din limbajele evoluat sînt identice cu cele descrise la interfața din SOCRATE-MINI, în plus există un cod de eroare suplimentar pentru cazul cînd  $COD-ER-W1 = 0$  :

2.6  $COD-ER-W2 = 5 \Rightarrow$  mai mulți parametri de apel decît erau necesari.

#### Exemple de programe pentru rețea scrise în limbaje evoluat

##### Exemplu de program scris în COBOL

Programul citește un fișier de date și încarcă în paralel două baze de date aflate în două noduri diferite ale unei rețele de minicalculatoare.

După terminarea încărcării se execută tot în paralel un subprogram SOCRATE-MINI care afișează la CO:-ul sistemului din nodul respectiv numărul de angajați încărcăți și repune bazele de date amintite mai sus în starea anterioară apelului.

Vor fi deci două subprograme INCARC-UN-ANGAJAT și AFISARE care sînt perfect identice cu cele prezentate anterior, formalul F-ANGAJAT-1 fiind descris mai sus.

Programul apelant scris în COBOL este următorul :

IDENTIFICATION DIVISION.  
PROGRAM-ID. P226.

```
*
* P226
*
* PROGRAMUL TESTEAZA APELUL UNOR SUBPROGRAME SOCRATE AFLATE
* ÎN DOUĂ NODURI ALE UNEI REȚELE DE MINICALCULATOARE DE
* CATRE UN PROGRAM COBOL AFLAT ÎNTR-UN ALT NOD AL REȚELEI.
* PROGRAMUL COBOL CITEȘTE FIȘIERUL „FANGAJAT1.DAT” TRANSMITÎND
* DATELE SUBPROGRAMELOR SOCRATE.
* REȚELEI.
* ACESTE SUBPROGRAME REALIZEAZĂ ÎNCĂRCAREA CARACTERISTICILOR
* ENTITĂȚII „ANGAJAT” DIN CELE DOUĂ BAZE DE DATE EXISTENTE
* ÎN NODURILE RESPECTIVE.
* FOLOSIND FUNCȚIUNILE „NETSOC” ȘI „NETSYN” ÎNCĂRCAREA CELORE --
* DOUĂ BAZE DE DATE SE FACE ÎN PARALEL.
*
```

ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.

SELECT FANGAJAT1 ASSIGN TO „SY:[1, 100]FANGAJAT1.DAT”.

DATA DIVISION.  
FILE SECTION.

FD FANGAJAT1 LABEL RECORD STANDARD.

01 ART-ANGAJAT.

```
02 NUME PIC X(30).
02 MARCA PIC 9(4).
02 FUNCTIE PIC X(10).
02 SEX PIC X.
02 STARE-POLITICĂ PIC XXX.
02 STARE-CIVILĂ PIC X.
02 TALIE PIC 99V99.
02 SALARIU PIC 9(4).
02 RATE-SPORURI PIC 9(4).
02 FILLER PIC X(15).
02 BINAR PIC 9(6)V99 COMP.
02 FILLER PIC X(24).
```

WORKING-STORAGE SECTION.

```
77 NR-LEG-1 PIC 9(4) COMP.
77 NR-LEG-2 PIC 9(4) COMP.
77 BINAR-AF PIC +(6).99.
77 COD-ER-W1-AF PIC +9(3).
77 COD-ER-W2-AF PIC +9(3).
77 COD-ER-MCR PIC S9(4) COMP.
01 ART-CNT.
02 DSB PIC 9(4) COMP VALUE 0.
02 DSR PIC 9(4) COMP VALUE 104.
02 LNGMAX PIC 9(4) COMP VALUE 104.
01 COD-ER-1.
02 COD-ER-W1-1 PIC S9(4) COMP.
02 COD-ER-W2-1 PIC S9(4) COMP.
01 COD-ER-2.
02 COD-ER-W1-2 PIC S9(4) COMP.
02 COD-ER-W2-2 PIC S9(4) COMP.
01 ART-CRB-1.
02 NUME-NOD-1 PIC X(6).
```

```

02 FILLER          PIC X(22).
02 NUME-UTI-1     PIC X(16).
02 FLILER         PIC XX.
02 PAROLA-1      PIC X(8).
02 FILLER         PIC X(18).

01 ART-CRB-2.
02 NUME-NOD-2    PIC X(6).
02 FILLER        PIC X(22).
02 NUME-UTI-2    PIC X(16).
02 FILLER        PIC XX.
02 PAROLA-2     PIC X(8).
02 FILLER        PIC X(18).

01 ART-SOC-1.
02 NUMPRE-1     PIC X(24) VALUE „ÎNCARC-UN-ANGAJAT".
02 LUNG-BUF-1  PIC 9(4) COMP VALUE 104.
02 BUF-DATE-1.

03 FILLER       PIC X(76).
03 BINAR-1     PIC 9(6)V99 COMP.
03 FILLER       PIC X(24).

01 ART-SOC-2.
02 NUMPRE-2    PIC X(24) VALUE „ÎNCARC-UN-ANGAJAT".
02 LUNG-BUF-2  PIC 9(4) COMP VALUE 104.
02 BUF-DATE-2.

03 FILLER       PIC X(76).
03 BINAR-2     PIC 9(6)V99 COMP.
03 FILLER       PIC X(24).

01 LINIE-COMANDA PIC X(80).
PROCEDURE DIVISION.

```

A0.

```

DISPLAY „TRANSMITEȚI INFORMAȚIILE PENTRU PRIMUL NOD AL REȚELEI"
DISPLAY „NUME NOD : " NO ADVANCING ACCEPT NUME-NOD-1
DISPLAY „NUME UTILIZATOR : " NO ADVANCING ACCEPT NUME-UTI-1
MOVE „SET /NOECHO=TI:" TO LINIE-COMANDĂ
CALL „MCR" USING LINIE-COMANDĂ COD-ER-MCR
PERFORM VERMCR
DISPLAY „PAROLA : " NO ADVANCING ACCEPT PAROLA 1
MOVE „SET /ECHO=TI:" TO LINIE-COMANDĂ
CALL „MCR" USING LINIE-COMANDĂ COD-ER-MCR
PERFORM VERMCR
DISPLAY „ "
DISPLAY „TRANSMITEȚI INFORMAȚIILE PENTRU AL DOILEA NOD AL REȚELEI"
DISPLAY „NUME NOD : " NO ADVANCING ACCEPT NUME-NOD-2
DISPLAY „NUME UTILIZATOR : " NO ADVANCING ACCEPT NUME-UTI-2
MOVE „SET /NOECHO=TI:" TO LINIE-COMANDĂ
CALL „MCR" USING LINIE-COMANDA COD-ER-MCR
PERFORM VERMCR
DISPLAY „PAROLA : " NO ADVANCING ACCEPT PAROLA-2
MOVE „SET ELCHO=TI:" TO LINIE-COMANDA
CALL „MCR" USING LINIE-COMANDĂ COD-ER-MCR
PERFORM VERMCR
DISPLAY „ "

```

\*  
\*  
\*

CONECTEAȚĂ PROCESORUL SOCRATE DIN NODUL 1

CALL „NETCNT“ USING ART-CNT NR-LEG-1 ART-CRB-1 COD-ER-1  
PERFORM VERNET-1

\*  
\*  
\*

CONECTEAZA PROCESORUL SOCRATE DIN NODUL 2

CALL „NETCNT“ USING ART-CNT NR-LEG-2 ART-CRB-2 COD-ER-2  
PERFORM VERNET-2.  
OPEN INPUT FANGAJAT1.

A1.

READ FANGAJAT1 AT END GO TO A2.  
MOVE ART-ANGAJAT TO BUF-DATE-1

\*  
\*  
\*

APELEAZĂ SUBPROGRAMUL „ÎNCARC-UN-ANGAJAT“ DIN NODUL 1

CALL „NETSOC“ USING ART-SOC-1 NR-LEG-1 COD-ER-1  
PERFORM VERNET-1  
MOVE ART-ANGAJAT TO BUF-DATE-2

\*  
\*  
\*

APELEAZA SUBPROGRAMUL „ÎNCARC-UN-ANGAJAT“ DIN NODUL 2

CALL „NETSOC“ USING ART-SOC-2 NR-LEG-2 COD-ER-2  
PERFORM VERNET-2

\*  
\*  
\*

AȘTEAPTĂ TERMINAREA PRELUCRĂRII DIN NODUL 1

CALL „NETSYN“ USING NR-LEG-1 COD-ER-1  
PERFORM VERNET-1  
IF COD-ER-W1-1 = 1 MOVE BINAR-1 TO BINAR-AF  
DISPLAY NUME „ “ BINAR-AF „ ÎN NODUL “ NUME-NOD-1.

\*  
\*  
\*

AȘTEAPTĂ TERMINAREA PRELUCRĂRII ÎN NODUL 2

CALL „NETSYN“ USING NR-LEG-2 COD-ER-2  
PERFORM VERNET-2  
IF COD-ER-W1-2 = 1 MOVE BINAR-2 TO BINAR-AF  
DISPLAY NUME “ “ BINAR-AF “ ÎN NODUL “ NUME-NOD-2.  
GO TO A1.

A2.

CLOSE FANGAJAT1  
MOVE „AFIȘARE“ TO NUMPRE-1 NUMPRE-2  
MOVE 0 TO LUNG-BUF-1 LUNG-BUF-2

\*  
\*  
\*

APELEAZĂ SUBPROGRAMUL „AFIȘARE“ DIN NODUL 1

CALL „NETSCC“ USING ART-SCC-1 NR-LEG-1 COD-ER-1  
PERFORM VERNET-1

\*  
\*  
\*  
\*

APELEAZĂ (AȘTEPTÎND TERMINAREA PRELUCRĂRII)  
SUBPROGRAMUL „AFIȘARE“ DIN NODUL 2

CALL „NETSCS“ USING ART-SOC-2 NR-LEG-2 COD-ER-2  
PERFORM VERNET-2

\*  
\*  
\*

AȘTEAPTĂ TERMINAREA PRELUCRĂRII DIN NODUL 2

```

CALL „NETSYN“ USING NR-LEG-1 COD-ER-1
PERFORM VERNET-1.
A3.
*   DECONECTEAZA LEGATURA CU PROCESORUL SOCRATE DIN NODUL 1
*
*   CALL „NETDSC“ USING NR-LEG-1 COD-ER-1 PERFORM VERNET-1
*
*   DECONECTEAZĂ LEGĂTURA CU PROCESORUL SOCRATE DIN NODUL 2
*
*   CALL „NETDSC“ USING NR-LEG-2 COD-ER-2 PERFORM VERNET-2
STOP RUN.
VERMCR.
    IF COD-ER-MCR NOT = 1
    MOVE COD-ER-MCR TO COD-ER-W2-AF
    DISPLAY „COD-EROARE-MCR = “ COD-ER-W2-AF.
VERNET-1.
    IF COD-ER-W1-1 NOT = 1
    MOVE COD-ER-W1-1 TO COD-ER-W1-AF
    MOVE COD-ER-W2-1 TO COD-ER-W2-AF
    DISPLAY „COD EROARE REȚEA NODUL „ NUME-NOD-1 “ “
        COD-ER-W1-AF „ * “ COD-ER-W2-AF.
VERNET-2.
    IF COD-ER-W1-2 NOT = 1
    MOVE COD-ER-W1-2 TO COD-ER-W1-AF
    MOVE COD-ER-W2-2 TO COD-ER-W2-AF
    DISPLAY „COD EROARE REȚEA NODUL „ NUME-NOD-2 “ “
        COD-ER-W1-AF „ * “ COD-ER-W2-AF.

```

#### Exemplu de program scris în FORTRAN

Programul citește un fișier de date și încarcă o bază de date aflată în alt nod al rețelei.

După terminarea încărcării se execută un subprogram care afișează la CO:-ul sistemului din nodul destinație numărul de angajați încărcați și readuce baza de date respectivă în starea inițială.

Programele apelate sînt aceleași ca și la apelarea din COBOL.

Programul apelant scris în FORTRAN este următorul :

```

C
C   P231
C
C   PROGRAMUL TESTEAZĂ APELUL UNUI SUBPROGRAM SOCRATE AFLAT
C   ÎNTR-UN NOD AL UNEI REȚELE DE MINICALCULATOARE DE CĂTRE
C   UN PROGRAM FORTRAN AFLAT ÎN ALT NOD AL REȚELEI.
C   PROGRAMUL FORTRAN CITEȘTE FIȘIERUL „FOR001.DAT“ (IDENTIC CU
C   FIȘIERUL „FANGAJAT1.DAT“) TRANSMITIND DATELE SUBPROGRAMULUI
C   SOCRATE CARE ÎNCARCĂ CARACTERISTICILE ENTITAȚII „ANGAJAT“
C   DIN BAZA EXISTENTĂ ÎN NODUL RESPECTIV.
C
INTEGER NMPRE1(12),NMPRE2(12),DSB,DSR,LNGMAX,NUMNOD(3),NRLEG
INTEGER CODERN(2),FILER1(11),FILER2,FILER3(9),NUMUTI(8)
INTEGER PAROLA(4),NMPREC(12),NUME(15),FUNCT(5),LNGBFD
INTEGER*4 MARCA,FILER5(6),BINAR
BYTE SEX,STAPOL(3),FILER4(15),STACIV
BYTE TALIE(4),SALAR(4),RATESP(4)
COMMON /ARTCNT/DSB,DSR,LNGMAX

```





## 7. INTERFAȚĂ SOCRATE-MINI – SOCRATE Felix

### Necesitate, obiective

Interfața a fost realizată ca urmare a solicitării unor utilizatori ai SGBD SOCRATE-MINI, care au în dotare și calculatoare de tip Felix C-256/1024 și care exploatează pe aceste calculatoare SGBD-uri de tip SOCRATE.

Acești utilizatori au simțit necesitatea ca, interconectînd un calculator Felix cu un minicalculator, să aibă acces la informațiile dintr-o bază de date instalată pe unul dintre calculatoare, pornind de la baza de date de pe celălalt. Proiectarea și realizarea interfeței a fost facilitată de faptul că SGBD-urile gazdă au concepte asemănătoare și limbaje de definire a structurii și de manipulare a datelor, practic identice.

Interfața permite ca, dintr-un program scris în limbaj de manipulare SOCRATE-MINI, să poți solicita transmisia unor informații către calculatorul Felix, lansarea în execuție a unui program precompilat catalogat într-o bază de date de acolo, care să prelucreze acele informații și recepția informațiilor obținute în urma prelucrării. Programul precompilat poate avea ca scop fie interogarea, fie actualizarea bazei de date de pe Felix. SGBD-ul de pe Felix poate fi SOCRATE V1.5 CII, SOCRATE V1.6R ITC sau orice alt SGBD de același tip, care respectă normele de apel pentru interfața COBOL-SOCRATE.

Pentru a asigura accesul la informațiile din baza de date de pe Felix, limbajul de manipulare SOCRATE-MINI pune la dispoziția utilizatorului primitive de acces, analoge celor de la interfața Limbaje evaluate – SOCRATE-MINI.

Confidențialitatea datelor este asigurată prin sistemul de drepturi de acces și de parole specific SGBD SOCRATE Felix.

### Primitivele de legătură

Apelul unei funcțiuni de interfață cu SOCRATE Felix are următoarea formă sintactică :

**APPEL** FX-xxx **AVEC** **BUFFER** n

unde xxx poate lua valorile **CNT**, **EXC**, **DSC**.

Peste **BUFFER** n este necesară plierea unui formal avînd următoarea structură :

FORMAL Nume-formal

DEBUT

Numar-cuplor

BINAIRE 2

Nume-baza

MOT 12

Nume-utilizator

MOT 8

Parola	MOT	8
Cod-eroare-1	BINAIRE	4
Cod-eroare-2	BINAIRE	4
Lung-buffer-emis	BINAIRE	4
Lung-buffer-recepționat	BINAIRE	4
Nume-precompilat	MOT	30
Date-utilizator-emise		
.		
.		
.		
Date-utilizator-recepționate		
.		
.		
.		
FIN		

unde :

– *Număr-cuplor* este numărul cuplorului XW (interfața sincronă) utilizat. Uzual are valoarea 0.

– *Nume-baza*, *Nume-utilizator* și *Parola* sînt valorile necesare pentru accesu la baza de date de la Felix. Ele corespund parametrilor BN, PN și PW de la cartela % SOC.

– *Cod-eroare-1* și *Cod-eroare-2* sînt cuvinte de stare prin care se comunică modul în care a decurs prelucrarea.

– *Lung-buffer-emis* este lungimea bufferului cu datele ce vor fi transmise către Felix spre prelucrare.

– *Lung-buffer-recepționat* este lungimea bufferului cu datele ce vor fi recepționate de la Felix ca rezultat al prelucrării.

*Lung-buffer-emis* și *Lung-buffer-recepționat* trebuie să îndeplinească condițiile :

– să fie valori pare (fiecare din ele poate fi eventual 0) ;

– suma lor (deci lungimea totală a formalului corespunzător de la Felix) trebuie să fie multiplu de 4.

– *Nume-precompilat* este numele programului precompilat, catalogat în baza de date de la Felix, ce se dorește lansat în execuție.

– *Date-utilizator-emise* și *Date-utilizator-recepționate* sînt informațiile propriu-zise ce se transferă între cele două calculatoare. Lor trebuie să-i corespundă la Felix un formal cu caracteristicile aranjate în aceeași ordine. Aceste caracteristici nu pot fi decît de tip MOT și DILATE.

Primele cîmpuri din formal, inclusiv *Parola*, nu pot fi modificate între FX-CNT și FX-DSC.

Plierea acestui formal în BUFFER n se face prin instrucțiunile :

M X<sub>1</sub> = FORMAL nume-formal DANS BUFFER n

S X<sub>1</sub>

urmate de instrucțiunile de încărcare a cîmpurilor din formal.

## Primitiva de conectare

### Sintaxa

#### APPEL FX-CNT AVEC BUFFER n

unde în BUFFER n se află un formal cu structura celui descris mai sus. Cîmpurile necesare pentru conectare sînt : Număr-cuplor, Nume-baza, Nume-utilizator și Parola.

#### Comentarii

Primitiva de conectare realizează următoarele operații :

- lansează în execuție task-ul de interfață, cu numele SOMIFE (task-ul corespunzător de la Felix se presupune deja lansat în execuție în acest moment) ;
- deschide linia de comunicație și conectează cele două calculatoare ;
- transmite parametrilor Nume-bază, Nume-utilizator și Parola către Felix ;
- verifică drepturile de acces ;
- deschide baza de date de la Felix (LOGIN) ;
- pune pe nedefinit variabilele  $X_1$ ,  $Y_1$ ,  $W_1$  și  $Z_1$  în procesorul de interfață de la Felix ;
- poziționează cuvintele de stare Cod-eroare-1 și Cod-eroare-2 pentru a anunța dacă totul s-a desfășurat corect.

Pe toată durata legăturii (pînă la FX-DSC) variabilele  $X_1$ ,  $Y_1$ ,  $W_1$  și  $Z_1$  de la Felix sînt globale pentru toate precompilatele executate în această perioadă.

Nu este posibil transferul automat al variabilelor de lucru între programul apelant de la mini și precompilatul apelat de la Felix. Acest lucru se poate realiza numai utilizînd formalul de legătură.

Un singur utilizator poate lucra la un moment dat folosind această interfață. Pînă la execuția primitivei FX-DSC, orice altă solicitare de conectare prin FX-CNT este refuzată.

## Primitiva de acces

### Sintaxa

#### APPEL FX-EXC AVEC BUFFER n}

unde, în BUFFER n sînt încărcate cîmpurile din formal pînă la, inclusiv, Date-utilizator-emise.

#### Comentarii

Primitiva de acces realizează următoarele operații :

- transmite către Felix informațiile din cîmpurile din formal ;
- lansează în execuție programul precompilat dorit, care prelucrează informațiile din prima parte a bufferului și încarcă rezultatele prelucrării în partea a doua ;
- recepționează de la Felix informațiile din partea a doua a bufferului, de lungime totală egală cu parametrul Lung-buffer-recepționat ;
- poziționează cuvintele de stare Cod-eroare-1 și Cod-eroare-2.

Programul precompilat de la Felix trebuie să înceapă cu o instrucțiune de forma

$$M \ X_i = \text{FORMAL Nume-formal-2 DANS BUFFER } 0$$

unde cîmpurile din Nume-formal-2 sînt cele din Date-utilizator-emise, urmate de cele din Date-utilizator-recepționate, păstrînd ordinea, tipul și dimensiunea cîmpurilor din Nume-formal.

Aceste caracteristici nu pot fi decît de tip MOT și DILATE.

Lungimea totală a cîmpurilor din Nume-formal-2 trebuie să fie multiplu de 4 octeți.

## Primitiva de deconectare

### Sintaxa

#### APPEL FX-DSC AVEC BUFFER n

unde în BUFFER n sînt necesare cîmpurile : Număr-cuplor, Nume-baza, Nume-utilizator și Parola.

#### Comentarii

- Primitiva de deconectare realizează următoarele operații :
- transmite către Felix informațiile din formal ;
- închide baza de date de la Felix (LOGOUT) ;
- închide linia de comunicație ;
- termină procesorul de interfață de la Felix ;
- termină task-ul de interfață SOMIFE ;
- poziționează cuvintele de stare Cod-eroare-1 și Cod-eroare-2.

## Tratarea erorilor

Valorile celor două cuvinte de stare au următoarele semnificații :

- Cod-eroare-1 = 1            Primitiva s-a terminat cu succes.
- Cod-eroare-1 = 0            Eroare de programare :
  - Cod-eroare-2 = 1            CNT deja efectuat (la FX-CNT) ;
  - Cod-eroare-2 = 2            CNT neefectuat (la FX-EXC sau FX-DSC) ;
  - Cod-eroare-2 = 3            Lungime formal prea mică. Se efectuează următoarele verificări :
    - FX-CNT și FX-DSC : lg-formal  $\geq$  34
    - FX-EXC : lg-formal  $\geq$  68 + lg-buf-emis + lg-buf-recepționat
  - Cod-eroare-2 = 4            Lg-buf-emis sau Lg-buf-recepționat impară.
- Cod-eroare-1 = -1            Eroare detectată în legătura inter-task dintre procesorul SOC și SOMIFE.
  - Cod-eroare-2 = -n            Semnificația codului de eroare se găsește în anexa B din Executive Reference Manual.
- Cod-eroare-1 = -2            Eroare detectată în legătura dintre mini și Felix
  - Cod-eroare-2 = -n            Semnificația codului de eroare se găsește în tab. 3 din Manualul de utilizare TELEMINI.
- Cod-eroare-1 = -3            Eroare detectată în execuția precompilatorului de la Felix. În acest caz, datele încărcate în formal pînă în acest moment sînt transmise integral de la Felix.
  - Cod-eroare-2 = n            Semnificația codului de eroare se găsește în cap. 6 din SOCRATE Manuel d'utilisation.

## Exemplu de program

Programul apelează în buclă un precompilat catalogat într-o bază de date de la Felix care, primind ca informație de intrare marca unui angajat, furnizează drept rezultat numele și retribuția acestuia.

Formalul catalogat în baza de date SOCRATE-MINI are structura :

```
D
FORMAL  FELIX
        DEBUT
        NR-CUPLOR      BINAIRE 2
        NUME-BAZA      MOT      12
        NUME-UTI        MOT      8
        PAROLA          MOT      8
        ERR1            BINAIRE 4
        ERR2            BINAIRE 4
        LUNG1           BINAIRE 4
        LUNG2           BINAIRE 4
        NUME-PRECOMP    MOT      30
        MARCA           DILATE 4
        NUME            MOT      28
        RETRIB          DILATE 4
        FIN
```

FIN ?

Programul apelant este următorul :

```
/*
/* P235
/* PROGRAMUL TESTEAZĂ INTERFAȚA SOCRATE-MINI – SOCRATE-FELIX
/*
M X1 = FORMAL FELIX DANS BUFFER 1
S X1
M NR-CUPLOR DE X1 = 0
M NUME-BAZA DE X1 = 'S8B01
M NUME-UTI DE X1 = EXT ( 'NUME UTILIZATOR :' )
APPEL MCR AVEC 'SET /NOECHO=TI:'
M PAROLA DE X1 = EXT ( 'PAROLA :')
APPEL MCR AVEC 'SET /ECHO=TI:'
/*
/* CONECTEAZĂ PROCESORUL DE INTERFAȚĂ ȘI DESCHIDE BAZA DE DATE DE LA
FELIX
/*
APPEL FX-CNT AVEC BUFFER 1
SI ERR1 DE X1 ↑ = 1 ALORS I(1) 'CNT – ERR1=' I(+1) ERR1 DE X1
      I(+1) ' ERR2=' I(+1) ERR2 DE X1 ECRIRE
      FIN
M LUNG1 DE X1 = 4
M LUNG2 DE X1 = 32
M NUME-PRECOMP DE X1 = 'PROTEST
FAIRE
M MARCA DE X1 = U
M MARCA DE X1 = EXT
SI MARCA DE X1 = U ALORS SORTIE FIN
/*
/* APELEAZĂ PROGRAMUL PRECOMPILAT DE LA FELIX CU NUMELE PROTEST
/*
APPEL FX-EXC AVEC BUFFER 1
SI ERR1 DE X1 = 1 ALORS I(1) '*** PERSOANA CU MARCA ' I(+1) MARCA DE X1
      I(30) ' ESTE ' I(+1) NUME DE X1
      I(68) 'SI ARE SALARIUL ' I(+1) RETRIB DE X1 ECRIRE
      SINON I(1) 'EXC – ERR1=' I (+1) ERR1 DE X1
      I (+1) ' ERR2=' I(+1) ERR2 DE X1 ECRIRE
      FIN
REFAIRE
FIN
```

```

/*
/* INCHIDE BAZA DE DATE SI DECONECTEAZĂ PROCESORUL DE INTERFATĂ DE
/* LA FELIX
APPEL FX-DSC AVEC BUFFER 1
SI ERR1 DE X1 ↑ = 1 ALORS I(1) 'DSC — ERR1=' I (+1) ERR1 DE X1
      I(+1) ' ERR2=' I(+1) ERR2 DE X1 ECRIRE
      FIN
?

```

Formalul catalogat în baza de date de la Felix are structura :

```

D
FORMAL FISTEST
  DEBUT
  MARCA      DILATE 4
  NUME       MOT   28
  RETRIB     DILATE 4
  FIN
FIN ?

```

Precompilatul catalogat în baza de date de la Felix este următorul :

```

::DEFPRO PROTEST
::EXP
  D X1 D X2
  M X1 = FORMAL FISTEST DANS BUFFER 0
  M Y1 = MARCA DE X1
  M X2 = UN ANGAJAT AVEC MARCA = Y1 ;
  M NUME DE X1 = NUME DE X2
  M RETRIB DE X1 = SALARIU DE X2
  D X1 D X2
::FDEF ?

```

## ANEXA A

### MESAJE DE EROARE

#### Erori detectate la catalogarea structurii

##### Erori detectate în analiza sintactică

\*ER.D00 NL yy ... <atom> ... IN LOC DE : <atom> [/<atom> ...]  
 [...REPARAT CU : ...<atom>]  
 [... → IN NL yy ...<atom>]

Pentru ca în continuare, analiza sintactică să decurgă cât mai corect, se încearcă mai întâi repararea erorii, presupunându-se că <atom> este un cuvânt cheie cu un singur caracter eronat.

Dacă eroarea nu este reparabilă, se încearcă sincronizarea pe începutul următoarei caracteristici (indicată prin mesajul „→“).

##### Erori detectate în analiza semantică

\*ER.Dxx NL yy ...<atom> [...<text explicativ>]  
 unde :

- xx este codul erorii și va avea valori între 1 –99 ;
- yy este numărul liniei din programul sursă;
- <atom> este elementul sintactic în tratarea căruia a fost detectată eroarea : un cuvânt cheie, un identificator, o constantă sau o etichetă virtuală.

Analiza structurii se continuă până la sfârșit, fără ca structura să fie catalogată. La sfârșitul analizei va apare mesajul :

\*\* ERORI DETECTATE : n

##### Erori fatale

\*ER.Dxxx

Erorile din această categorie au ca efect oprirea imediată a procesorului DFS. Codurile de eroare xxx sînt cuprinse între 101 –200.

Mesajele de eroare ce pe listing sînt însoțite de mesaje explicite afișate la TI :  
 , prezentate în Manualul de operare.

## Coduri de eroare

Cod	Cauza erorii și text explicativ
01	Eticheta virtuală care precede DEBUT, la o catalogare inițială, este diferită de \$FIS.
02	Eticheta virtuală \$xxx definește un domeniu negenerat cu GFM.
03	Eticheta virtuală interzisă într-un bloc sau într-o entitate inclusă.
04	Eticheta virtuală *xxx este diferită de ultima etichetă \$xxx.
05	Depășire a 4 nivele de imbricare domenii.
06	Etichete virtuale pentru domenii interzise : STR, PRG, DIC.
07	Eticheta virtuală nu are 3 caractere alfanumerice.
08	Un domeniu deschis cu \$xxx într-o entitate de nivel 1 nu a fost închis cu *xxx înainte de FIN. Text explicativ : DOMENIUL xxx
09	O entitate de nivel 1 are caracteristici descrise pe mai mult de 4 domenii. Text explicativ : DOMENIUL xxx
10	Depășire a 4 nivele de imbricare pentru entități sau formale.
11	La adăugarea de structură sau tratare comutator /MAP, domeniul STR este gol (formatat).
12	Număr realizări pentru o entitate este 0 sau $> 2^{**} 24 - 1$ .
13	Lungime bloc sau entitate $> 2^{**} 24 - 1$ . Text explicativ : DOMENIUL xxx Adresa virtuală domeniu $> 2^{**} 31 - 1$ .
14	Text explicativ : DOMENIUL xxx Spațiul virtual domeniu împărțit la spațiul real dă un număr pliu $> 127$ . Text explicativ : DOMENIUL xxx Soluție : creștere spațiu real.
16	Lungime declarată pentru un MOT sau pentru valorile dintr-o listă de valori este 0 sau $> 30$ .
17	Identificatori identici la același nivel.
18	La DECIMAL sau la PACKE, DILATE și BINAIRE din formale, declarația „n1 V n2” eronată : $n1 + n2 = 0$ sau $> 18$ .
19	La valori numerice bornate, borna inferioară sau superioară eronată : – la BINAIRE, mai mult de 18 cifre ; – la DECIMAL, depășire declarația „n1 V n2”.
20	La valori numerice bornate : – borna superioară $<$ borna inferioară, sau – max. borna superioară $<$ borna superioară.
21	La lista de valori, numărul de valori declarat este 0 sau $> 254$ .
22	La lista de valori, numărul de valori din listă $>$ numărul de valori declarat.
23	Entitatea inversată, inelul, entitatea referită sau un calificator al acestora – negăsit în structură.
24	Lungime FILLER $> 65535$ .
25	Caracteristica declarată cheie nu este subordonată unei entități de nivel 1.
26	Depășire adresă virtuală în DIC : $> 2^{**} 31 - 1$ .
27	Inelul sau referința nu sînt subordonate direct unei entități.
28	Depășire a 4 nivele de calificare la INVERSE sau REFERE.
29	Depășire tabela de referințe încrucișate la o compilare. Soluție : fragmentarea structurii și catalogare prin adăugări succesive.



- 30 Tip CHAINE (SIMPLE/ DOUBLE) diferit la ANNEAU și REFERE asociate.  
 31 Număr repetiții FORMAL = 0 sau > 65535.  
 32 Dimensiune FORMAL > 65535.  
 33 Lungime MOT formal este 0 sau > 80.  
 34 Poziționarea pentru redefinire, la un BINAIRE formal, nu este multiplu de 2.  
 35 Un MOT cu opțiune RAD50 nu poate fi declarat cheie.  
 36 Inversa nu este declarată la același nivel în structură cu entitatea inversată.  
 37 Entitate subordonată unui bloc.  
 38 Lungime bloc sau realizare entitate = 0.  
 39 Două referințe punctează același ANNEAU.  
 101 Eroare sintaxă în linia de comandă.  
 102 Apel în afara sesiunii.  
 103 DFS deja apelat.  
 104 UIC fără drept de acces la bază.  
 105 UIC fără drept de actualizare a bazei.  
 106 Eroare SGF pe fișierul de intrare.  
 107 Eroare SGF pe fișierul de listare.  
 108 Eroare SGF pe fișierul de manevră temporar.  
 109 Eroare în citire-scriere pe domeniul STR.  
 110 Domeniul STR inactiv.  
 111 Spațiul real saturat pentru domeniul STR.  
 112 „DEBUT” sau „D” absent.  
 113 „DEBUT” interzis peste o structură existentă.  
 114 Structură incoerentă.

## **Erori detectate de procesorul limbajului de manipulare**

Erorile detectate de procesorul SOC pot fi clasificate astfel :

- erori detectate în analiza programului sursă (erori ce pot apare fie dacă se dorește execuția unui program, prin intermediul procesorului SOC, fie la catalogarea unui program precompilat, prin intermediul procesorului MGS) :
  - erori detectate în analiza lexicală ;
  - erori detectate în analiza sintactică ;
  - erori detectate în analiza semantică ;
- erori detectate în execuția unui program :
  - erori recuperabile, comunicate în variabila ERREUR ;
  - erori fatale, ce au ca efect oprirea procesorului SOC ;
- erori de operare, ce au drept cauze evenimente externe procesorului SOC.

### **Erori detectate în analiza lexicală**

\*ER.ROL NL yy ...<caracter\ ... IN LOC DE : <caracter>[/<caracter> ...]  
 [ \*\*\* REPARAT CU <caracter> ]

- yy este numărul liniei din programul sursă ;
- <caracter> semnifică separator, literă (<li>) sau cifră (<cf>).

Se încearcă, atunci când este posibil, repararea erorii.

### Erori detectate în analiza sintactică

\*\*ER.R00 NL yy ...<atom> ... IN LOC DE : <atom>[!<atom> ...]  
 [\*\*\* REPARAT CU <atom>]  
 – <atom> reprezintă un cuvînt-cheie sau un atom lexical (identificator, constantă, variabilă X, Y, Z, W etc.).

### Erori detectate în analiza semantică

\*ER.R xxx NL yy  
 – xxx este codul erorii ; are valori cuprinse între 200–299 ;  
 – yy este numărul liniei din programul sursă.  
 După erorile din această categorie, programul este analizat în continuare, dar el nu va intra în execuție. La sfîrșitul analizei va apare mesajul :  
 \*\* ERORI DETECTATE : n

### Erori recuperabile detectate în execuție. Variabila ERREUR

Aceste erori nu opresc execuția programului în curs. Nu se afișează nici un mesaj pe listing sau la T1: . Singurul lor efect este poziționarea codului corespunzător erorii într-o variabilă particulară cu numele ERREUR. Erorile din această categorie au coduri cu valori între 1–99.

Pentru ca utilizatorul să nu fie obligat să inițializeze prin program variabila ERREUR, se realizează automat inițializarea ei cu „nedefinit” înainte de fiecare cerere, cu excepția următoarelor :

$$\text{și ERREUR} \quad \left\{ \begin{array}{l} = \\ \uparrow = \end{array} \right\} \quad \left\{ \begin{array}{l} \text{constantă-Y} \\ Y_1 \\ W_1 \\ U \end{array} \right\}$$

și

$$M \quad \left\{ \begin{array}{l} Y_1 \\ W_1 \end{array} \right\} = \text{ERREUR}$$

Alte cereri (de ex.: I ERREUR) nu sînt posibile deoarece în momentul execuției lor, ERREUR conține deja „nedefinit”.

Obs. : Variabila ERREUR este, de fapt, o variabilă de tip W și este tratată ca atare în analiză. Orice eroare detectată în analiza sintactică datorită variabilei ERREUR, va avea în corpul mesajului o referire la o variabilă  $W_1$ .

### Erori fatale detectate în execuție

\*ER.R xxx [<text explicativ>]  
 – xxx este codul erorii ; are valori cuprinse între 101–150 ;  
 – <text explicativ> oferă unul sau mai multe elemente suplimentare pentru depanarea erorii : numele caracteristicii, variabila  $X_1$ , numele programului la EXEC sau numele task-ului la APPEL etc.

Erorile din această categorie sînt erori de programare grave, care au ca efect oprirea imediată a procesorului SOC.

### Erori de operare

\*ER.R xxx [<text explicativ>]

— xxx este codul erorii; are valori cuprinse între 151 – 199;

— <text explicativ> poate fi : numărul fișierului utilizator sub forma BANDE n, codul SGF al erorii etc.

Erorile din această categorie sînt de obicei erori de operare sau datorate stării hardware a echipamentelor. Ele au ca efect oprirea imediată a procesorului SOC.

Mesajele de eroare de pe listing sînt însoțite, în acest caz, de mesaje explicite afișate la TI : , prezentate în Manualul de operare.

### Coduri de eroare

Prescurtări uzuale :

car	caracteristică
ENT	ENTITE
INV	INVERSE
v. num	valoare numerică bornată, inclusiv DECIMAL
l. val	listă de valori
ref. s	referință simplă
ref. inel	referință cu inel
MOT-for	MOT în formal
v. num. for	valoare numerică în formal
cit. com	citare complexă
U	nedefinit
ct. Y	constantă numerică întregă
ct. W	constantă zecimală
ct. Z	constantă alfanumerică (șir de caractere)
op <sub>1</sub>	operand din partea stîngă
op <sub>2</sub>	operand din partea dreaptă

Cod	Cerere	Cauza erorii și text explicativ
1	LIRE	EOF detectat
2	CKPT	Fișier jurnal saturat sau în eroare
3	cit. com	Nr. realizare ( $Y_1$ sau ct. Y) > nr. max specificat în structură
4	cit. com	Nr. realizare ( $Y_1$ ) = U
5	cit. com	Nr. realizare ( $Y_1$ sau ct. Y) = 0
6	G	Nr. realizare deja generat
7	cit. com	Nr. realizare negenerat
9	M $Y_1$ , $W_1 = Z_1$	Valoare nenumerică în $Z_1$
10	M... = DILATE	Valoare nenumerică în cîmpul din formal
11	M v. num. for = ...	Depășire receptor pentru v. num din formal

12	M v. num = ...	V. num în afara bornelor declarate în structură
14	M l. val = ...	Valoare inexistentă în structură
15	M ref. inel = ...	Referință deja prezentă într-un inel
16	cit. com AVEC	Valoare cheie inexistentă în DIC
17	M cheie = ...	Scriere cheie unică deja existentă
18	cit. com AVEC	Valoare $opd_2$ = „nedefinit“
20	G	Toate realizările deja generate
21	cit. com	Calificare sau $opd_2$ de tip referință = nedefinit
22	...AYANT...	Nici o realizare nu îndeplinește condiția
101	... DE $X_1$	$X_1 = U$
	M $Y_1 = NUMDE X_1$	Text explicativ : <[nume-caracteristică] $X_1$
102	... DE $X_1$	$X_1$ poziționat pe un formal, la calificarea unei caracte-
	M $Y_1 = NUMDE X_1$	ristici din bază
	M $X_1 = SUIVANT DE$	Text explicativ : [nume – caracteristică] $X_1$
103	... DE $X_1$	$X_1$ poziționat pe o altă entitate decît cea căreia îi apar
	M $X_1 = SUIVANT DE$	ține caracteristica
		Text explicativ : nume-caracteristică $X_1$
104	... DE $X_1$	$X_1$ nepoziționat pe formal, la calificarea unei caracte-
		ristici dintr-un formal
		Text explicativ : nume-caracteristică $X_1$
105	ATTACHE	Specificator fișier eronat
		Text explicativ : BANDE n
106	ATTACHE	Șir de atribute fișier eronat
		Text explicativ : BANDE n
107	ATTACHE	Depășire a 10 fișiere deschise simultan
		Text explicativ : BANDE n
108	LIRE	Absență ATTACHE
		Text explicativ : BANDE n
109	ECRIRE	Absența ATTACHE
		Text explicativ : BANDE n
110	DETACHE	Absență ATTACHE
		Text explicativ : BANDE n
111	ECRIRE	$X_1$ nepoziționat pe un formal
		Text explicativ : BANDE n
112	EXEC	Program precompilat absent
		Text explicativ : nume-program
113	EXEC	Nr. apeluri imbricate > 30
114	EXEC	Depășire stivă la salvare $X_1$ din listă
115	–	Tabela de simboluri ocupată
		Soluție : fragmentați programul
116	–	Tabela de constante plină
		Soluție : fragmentați programul
117	POUR, FAIRE	Depășire stivă sistem la apeluri imbricate cu AVEC sau PAR
118	M... = PACKE	PACKE în formal netratat
119	M PACKE = ...	PACKE în formal netratat
120	APPEL	– Lungimea zonei emițătoare este eronată (ex. : impară) sau nu este egală cu lung. zonei receptoare din task

		Text explicativ : nume-task-fiu
		– Parametri de apel eronați la o funcțiune de rețea
121	LIRE, ECRIRE	Text explicativ : nume-funcțiune-rețea Pentru un fișier cu RCF:F, lungime formal diferită de valoarea RCS.
122	M, G	Text explicativ : BANDE n Actualizare referință sau generare inversă cu o entitate diferită de cea folosită în structură
151	–	UIC fără drept de acces la bază
152	–	Apel în afara sesiunii
153	–	Domeniu inactiv
154	–	Spațiu real saturat
155	–	Eroare în citire domeniu
156	–	Eroare în scriere domeniu
157	–	Eroare pe fișierul de securitate (primul bloc)
158	–	Număr pliu > 127
159	–	UIC fără drept de actualizare a bazei
160	–	Rupere înlanțuiri dicționar
161	–	Saturare spațiu virtual alocat cheii în DIC
162	–	Domeniu STR vid
163	–	Eroare pe fișierul de intrare Text explicativ : COD: –n
164	–	Eroare pe fișierul de ieșire Text explicativ : COD : --n
165	–	Eroare în citire pe fișier de manevră Text explicativ : COD : –n
166	–	Eroare în scriere pe fișier de manevră Text explicativ : COD : –n
167	ATACHE	Eroare OPEN pe fișier utilizator Text explicativ : BANDE n COD : –n
168	LIRE	Eroare citire pe fișier utilizator Text explicativ : BANDE n COD : –n
169	ECRIRE	Eroare scriere pe fișier utilizator Text explicativ : BANDE n COD : –n
170	DETACHE	Eroare CLOSE pe fișier utilizator Text explicativ : BANDE n COD : –n
171	–	Extensie de memorie imposibilă
172	–	Valoare /DSR insuficientă
173	–	Valoare /DSB insuficientă Text explicativ : BANDE n
174	APPEL	Eroare lansare în execuție task Text explicativ : nume-task-fiu COD : –n
175	APPEL	– Task-ul apelat s-a terminat anormal Text explicativ : nume-task-fiu – Denumire funcțiune rețea inexistentă Text explicativ : nume-funcțiune-rețea
200	cit. com	UN, TOUT interzise pt. car. diferite de ENT, INV, inel
201	cit. com	UN, TOUT obligatorii pt. ENT, INV, inel
202	cit. com	Filtru permis numai la ENT, INV, inel

203	cit. com	Acces prin cheie permis numai la ENT, INV
204	cit. com	Adjectiv $X_1$ permis numai la ENT, INV, inel, ref
205	cit. com	TOUȚ interzis pt. ENT + nr. realizare
206	cit. com	Nr. realizare permis numai pt. ENT, INV, FORMAL
207	cit. com	Calificare permisă numai cu ENT, INV, inel, ref, bloc, $X_1$
208	M	Dacă $opd_1 = MOT$ , l. val, MOT-for, este permis numai $opd_2 = MOT$ , l. val, MOT-for, $Z_1$ , ct. Z, EXT, U
209	M	Dacă $opd_1 = Z_1$ , este permis numai $opd_2 = MOT$ , l. val, MOT-for, ct. Z, $Z_1$ , $Y_1$ , $W_1$ , EXT, U
210	M	Dacă $opd_1 = v.$ num, este permis numai $opd_2 = v.$ num, v. num. for, $Y_1$ , $W_1$ , ct. num, expr. aritm, EXT, U
211	M	Dacă $opd_1 = v.$ num. for, este permis numai $opd_2 = v.$ num, v. num. for, $Y_1$ , $W_1$ , ct. num, expr. aritm, EXT, U
212	M	Dacă $opd_1 = Y_1$ , $W_1$ , este permis numai $opd_2 = v.$ num, v. num. for, $Y_1$ , $W_1$ , ct. num, expr. aritm, $Z_1$ , D, NUMDE, EXT, U
213	M	Dacă $opd_1 = ref$ , este permis numai $opd_2 = ENT$ , INV, inel, ref, $X_1$ , U
214	M	Dacă $opd_1 = X_1$ , este permis numai $opd_2 = ENT$ , INV, inel, ref, FORMAL, SUIVANT, $X_1$ , U
215	M	Dacă $opd_2 = funcția D$ , este permis numai $opd_2 = ENT$ , INV, inel
216	M	Dacă $opd_2 = funcția NUMDE$ , este permis numai $opd_2 = ENT$ , INV, inel, ref, $X_1$
217	G	Dacă $opd_2$ lipsește, este permis numai $opd_1 = ENT$
218	G	Dacă $opd_2$ este prezent, sînt permisi numai $opd_1 = INV$ și $opd_2 = ENT$ , INV, inel, ref, $X_1$
220	G	Interzis ca $opd_1$ să conțină filtru sau acces prin cheie
221	I	Dacă $A = ct. Y$ , este permis numai $A \leq 132$ și $A \neq 0$
222	I	Dacă $B = ct. Y$ , este permis numai $-30 \leq B \leq 30$
223	condiție	Dacă unul din operanzii = $X_1$ sau U, sînt permisi numai operatorii = și $\uparrow$
224	condiție	Caract. de tip bloc interzisă în condiții
225	condiție	Dacă $opd_1 = MOT$ , l. val, MOT-for, $Z_1$ , este permis numai $opd_2 = MOT$ , l. val, MOT-for, $Z_1$ , ct. Z, U
226	condiție	Dacă $opd_1 = v.$ num; v. num. for., $Y_1$ , $W_1$ , este permis numai $opd_2 = v.$ num, v. num. for, $Y_1$ , $W_1$ , ct. num, U
227	condiție	Dacă $opd_1 = ref$ , $X_1$ , este permis numai $opd_2 = X_1$ , U
228	condiție	ENT, INV, inel interzise în condiții simple
229	SE	Permis numai $opd = INV$ , inel
230	S	Permis numai $opd = ENT$ , INV, inel, ref, $X_1$
231	REFAIRE	În afara unei bucle FAIRE
232	SORTIE	În afara unei bucle FAIRE sau POUR
233	BANDE n	$n > 10$
234	BUFFER n	$n > 10$
235	POUR	Permis numai $opd = ENT$ , INV, inel, ref, bloc, $X_1$

237	M	Indici eronați la M $Z_i(p_1) = Z_j(p)$ lg <sub>2</sub>
238	M	Interzis opd <sub>1</sub> = ENT, INV, inel, bloc
239	condiție	Dacă opd <sub>1</sub> = cit. com, interzis opd <sub>2</sub> = X <sub>1</sub>
240	condiție	Dacă opd <sub>1</sub> = cit: com, interzis opd <sub>2</sub> = cit. com
241	condiție	Dacă opd <sub>1</sub> = X <sub>1</sub> , interzis opd <sub>2</sub> = cit. com
242	ct. num	Lungime > 18
243	ct. Z sau identificator	Lungime > 30
244	X <sub>1</sub>	i > 15
245	Y <sub>1</sub>	i = 0 sau > 30
246	W <sub>1</sub>	i = 0 sau > 20
247	Z <sub>1</sub>	i = 0 sau > 10
248	ct. num	Nenumerică
249	EXEC	Listă cu mai mult de 15 X <sub>1</sub>
250	cit. com	Caracteristică negăsită în structură
251	ECRIRE n	n în afara intervalelor 0, 7 și 10-17.
252	cit. com	Caracteristică folosită pt. acces cu AVEC sau PAR ne-declarată cheie în structură
253	apel macro	Separator de apel eronat
254	expr. aritm	Depășire a 13 operatori aritmetici, inclusiv ..("
255	APPEL	Nume task-fiu eronat
256	cit. com	La acces prin AVEC, interzis opd <sub>2</sub> = U
257	apel macro	..?" în cadrul unui parametru sau separator de apel
258	M, G	Actualizare referință sau inversă cu o entitate diferită de cea prevăzută în structură.

## Erori detectate de programul bibliotecar

### Erori detectate în analiza lexicală

\*ER.M00 NL yy ...<atom> ...ÎN LOC DE <atom>[<atom> ...]  
 [...REPARAT CU : ...<atom>]  
 [... → IN NL yy ... <atom>]

### Erori detectate în execuție

ERR. MACRO 1 NL : nr. linie ... <nume>  
 Nume macroinstrucțiune sau precompilat > 30. caractere.

ERR. MACRO 2 NL : nr. linie ... <nume>  
 Nume macroinstrucțiune sau precompilat este cuvânt cheie.

ERR. MACRO 3 NL : nr. linie ... <nume>  
 Nume macrodefiniție sau precompilat există deja.

ERR. MACRO 4 NL : nr. linie ... <separator>  
 Separator de apel > 30. caractere.

ERR. MACRO 5 NL : nr. linie ...

Număr de parametri ai unui macro > 30.

ERR. MACRO 6 NL : nr. linie ... <nr. param>

Parametrul nr. param apare doar în modelul de expansiune.

ERR. MACRO 7 NL : nr. linie ...

Lungime macrodefiniție > 65535 octeți.

ERR. MACRO 8 NL : nr. linie ...

Spațiul virtual în PRG saturat.

ERR. MACRO 9 NL : nr. linie ... <nr. param>

Parametrul nr. param utilizat ambiguu (:n: și ':n:').

ERR. MACRO 10 NL : nr. linie ... <nume>

Numele macrodefiniției sau precompilatorului inexistent în domeniul DIC.

ERR. MACRO 11 NL : nr. linie ...

Domeniul DIC formatat.

ERR. MACRO 12 NL : nr. linie ...

Domeniul PRG formatat.

ERR. MACRO 13 NL : nr. linie ...

Modelul de expansiune a macrodefiniției este vid.

ERR. MACRO 14 NL : nr. linie ...

Numărul de parametri declarați în modelul de apel este mai mare decât numărul de parametri utilizați la apelul macroinstrucțiunii.

## ANEXA B

### CUVINTE CHEIE

#### Cuvinte cheie ale limbajului de descriere a structurii

A	D	FIN	REFERE
ANNEAU	DE	FORMAL	SIMPLE
FRIERE	DEBUT	INVERSE	TOUT
AVANT	DECIMAL	MOT	TOUTE
AVEC	DILATE	NON	UN
BINAIRE	DOUBLE	ORDONNE	UNE
CHAINE	ENTITE	PACKE	UNIQUE
CLE	FILLER	RAD50	V



## Cuvinte cheie ale limbajului de manipulare

A	DE	I	SE
ALORS	DETACHE	LIBERER	SI
APPEL	ECRIRE	LIRE	SINON
ATTACHE	ERREUR	M	SORTIE
AVEC	ET	NUMDE	SUIVANT
AYANT	EXT	OU	TOUT
BANDE	EXEC	PAS	TOUTE
BLOQUER	EXISTE	PAR	U
BUFFER	FIN	POUR	UN
CKPT	FAIRE	PAUSE	UNE
D	FORMAL	REFAIRE	
DANS	G	S	

## Cuvinte cheie ale programului bibliotecar

:DEFMAC	:FDEF	:LISALL
:DEFPRO	:LISMAC	:SUPMAC
:EXP	:LISPRO	:SUPPRO

## ANEXA C

### SINTAXA LIMBAJULUI DE MANIPULARE

<program>	::= <secvență> [?]
<secvență>	::= <instrucțiune><secvență>
	::= <instrucțiune>
<instrucțiune>	::= <instr-APPEL>
	::= <instr-ATTACHE>
	::= BLOQUER
	::= <instr-CKPT>
	::= <instr-D>
	::= <instr-DETACHE>
	::= <instr-ECRIRE>
	::= <instr-EXEC>
	::= <instr-repetitivă-FAIRE>
	::= <instr-G>
	::= <instr-I>
	::= LIBERER
	::= <instr-LIRE>

	::= <instr-M>
	::= <instr-PAUSE>
	::= <instr-repetitivă-POUR>
	::= REFAIRE
	::= <instr-S>
	::= <instr-SE>
	::= instr-condiț-SI>
	::= SORTIE
	::= SUIVANT
<instr-APPEL>	::= APPEL 'MCR' AVEC <param-APPEL>
	::= APPEL id-task [AVEC <param-APPEL>]
<param-APPEL>	::= 'sir'
	::= $Z_1$
	::= BUFFER n
<instr-CKPT>	::= CKPT $Y_1$
	::= CKPT
<instr-D>	::= D $X_1 =$ <citare-realizare>
	::= D $X_1 =$ FORMAL id-formal-1 DANS BUFFER n
	::= D $X_1 =$ FORMAL id-formal-rep <nr-formal-rep>
	DE $X_j$
	::= D $W_1 =$ n
<nr-formal-rep>	::= nr-întreg-pozitiv
	::= $Y_1$
<instr-ATTACHE>	::= ATTACHE BANDE n A <specificator-fișier>
	AVEC <parametrii-fișier>
<specificator-fișier>	::= specificator-fișier-RSX
	::= $Z_1$
<parametrii-fișier>	::= '<prm>, <bfs>, <rcs>, <rcf>'
	::= $Z_1$
<prm>	::= I
	::= O
<bfs>	::= nr-întreg-pozitiv
<rcs>	::= nr-întreg-pozitiv
<rcf>	::= F
	::= V
<instr-DETACHE>	::= DETACHE BANDE n -
<instr-ECRIRE>	::= ECRIRE BUFFER $n_1$ DANS BANDE $n_2$
	::= ECRIRE [n]
<instr-EXEC>	::= EXEC id-precompilat [[(<lista- $X_1$ >)]
<listă- $X_1$ >	::= $X_1$ <lista- $X_1$ >
	::= $X_1$

<instr-repetitivă-FAIRE>	::= FAIRE <secvența FIN>
<instr-G>	::= G <un>ent-niv-1[ X <sub>1</sub> ] [ <nr-realiz> ]
	::= G <un>ent-niv-2[ X <sub>1</sub> ] [ <nr-realiz> ] [ <calificare> ]
	::= G <tout> ent-niv-1
	::= G <tout> ent-niv-2 [ <calificare> ]
	::= G <cuantificator>inv-niv-1 = X <sub>1</sub>
	::= G <cuantificator>inv-niv-1 = <citare>
	::= G <cuantificator>inv-niv-2[ <calif-real> ] = <citare>
	::= G <cuantificator>inv-niv-2[ <calif-real> ] = X <sub>1</sub>
<instr-l>	::= l <citare>
	::= l <car-simplă> [ <calificare> ]
	::= l <format> <car-simplă> [ <calif-real> ]
	::= l [ <format> ] <formal-mot> DE X <sub>1</sub>
	::= l <format> ] <formal-nr> DE X <sub>1</sub>
	::= l [ <format> ] <variab-nr>
	::= l [ <format> ] Z <sub>1</sub>
	::= l [ <format> ] 'sir'
<format>	::= ([+] <poziție> [ <lg-format> ])
<poziție>	::= nr-întreg-pozitiv
	::= Y <sub>1</sub>
<lg-format>	::= nr.—întreg
	::= Y <sub>1</sub>
<instr-LIRE>	::= LIRE BANDE n <sub>1</sub> DANS BUFFER n <sub>2</sub>
<instr-M>	::= M X <sub>1</sub> = FORMAL id-formal-1 DANS BUFFER r
	::= M X <sub>1</sub> = FORMAL id-formal-rep <nr-formal-rep>
	DE X <sub>1</sub>
	::= M X <sub>1</sub> = <citare-realizare>
	::= M X <sub>1</sub> = SUIVANT DE <un> <element-entitate>
	::= M X <sub>1</sub> = SUIVANT DE <un> <element-invers>
	::= M X <sub>1</sub> = SUIVANT DE <un> <element-inel>
	::= M X <sub>1</sub> = X <sub>j</sub>
	::= M X <sub>1</sub> = U
	::= M Y <sub>1</sub> = D <citare>
	::= M Y <sub>1</sub> = NUMDE <citare-realizare>
	::= M Y <sub>1</sub> = NUMDE X <sub>j</sub>
	::= M <variab-nr> = U
	::= M <variab-nr> = <expr-aritm>
	::= M <variab-nr> = <formal-nr> DE X <sub>j</sub>
	::= M <variab-nr> = Z <sub>j</sub>
	::= M <variab-nr> = <ext>
	::= M <variab-nr> = val-nr <calif-real>
	::= M Z <sub>1</sub> = Z <sub>j</sub>
	::= M Z <sub>1</sub> [ [ <poz> ] ] = Z <sub>j</sub> ( <poz> <lg-sir> )
	::= M Z <sub>1</sub> = U
	::= M Z <sub>1</sub> = <variab-nr>
	::= M Z <sub>1</sub> = formal-mot DE X <sub>j</sub>

	::= M $Z_1 = \text{'sir'}$
	::= M $Z_1 = \langle \text{ext} \rangle$
	::= M $Z_1 = \langle \text{car-alfa} \rangle \langle \text{calif-real} \rangle$
	::= M formal-mot DE $X_1 = \text{'sir'}$
	::= M formal-mot DE $X_1 = Z_j$
	::= M formal-mot DE $X_1 = \text{formal-mot DE } X_j$
	::= M formal-mot DE $X_1 = \langle \text{ext} \rangle$
	::= M formal-mot DE $X_1 = U$
	::= M formal-mot DE $X_1 = \langle \text{car-alfa} \rangle \langle \text{calif-real} \rangle$
	::= M $\langle \text{formal-nr} \rangle$ DE $X_1 = \langle \text{expr-aritm} \rangle$
	::= M $\langle \text{formal-nr} \rangle$ DE $X_1 = \langle \text{formal-nr} \rangle$ DE $X_j$
	::= M $\langle \text{formal-nr} \rangle$ DE $X_1 = U$
	::= M $\langle \text{formal-nr} \rangle$ DE $X_1 = \langle \text{ext} \rangle$
	::= M $\langle \text{formal-nr} \rangle$ DE $X_1 = \text{val-nr} \langle \text{calif-real} \rangle$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = \text{'sir'}$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = Z_1$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = \text{formal-mot DE } X_j$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = U$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = \langle \text{ext} \rangle$
	::= M $\langle \text{car-alfa} \rangle [\langle \text{calificare} \rangle] = \langle \text{car-alfa} \rangle \langle \text{calif-real} \rangle$
	::= M val-nr $[\langle \text{calificare} \rangle] = \langle \text{expr-aritm} \rangle$
	::= M val-nr $[\langle \text{calificare} \rangle] = U$
	::= M val-nr $[\langle \text{calificare} \rangle] = \langle \text{formal-nr} \rangle$ DE $X_j$
	::= M val-nr $[\langle \text{calificare} \rangle] = \langle \text{val-nr} \rangle \langle \text{calif-real} \rangle$
	::= M val-nr $[\langle \text{calificare} \rangle] = \langle \text{ext} \rangle$
	::= M referință $[\langle \text{calificare} \rangle] = U$
	::= M referință $[\langle \text{calificare} \rangle] = \langle \text{citare-realizare} \rangle$
$\langle \text{instr-PAUSE} \rangle$	::= PAUSE $Y_1$
	::= PAUSE
$\langle \text{instr-repetitivă-POUR} \rangle$	::= POUR $\langle \text{citare} \rangle \langle \text{secvență} \rangle$ FIN
	::= POUR $X_1 \langle \text{secvență} \rangle$ FIN
$\langle \text{instr-S} \rangle$	::= S $\langle \text{citare} \rangle$
	::= S $X_1$
$\langle \text{instr-SE} \rangle$	::= SE $\langle \text{cuantificator} \rangle$ invers $[\langle \text{citare} \rangle]$
	::= SE $\langle \text{cuantificator} \rangle$ inel $\langle \text{citare} \rangle$
$\langle \text{instr-condiț-SI} \rangle$	::= SI $\langle \text{condiție-compusă} \rangle$ ALORS $\langle \text{secvență} \rangle$ [SINON $\langle \text{secvență} \rangle$ ] FIN
$\langle \text{condiție-compusă} \rangle$	::= $\langle \text{secvență-ET} \rangle$ OU $\langle \text{condiție-compusă} \rangle$
	::= $\langle \text{secvență-ET} \rangle$
$\langle \text{secvență-ET} \rangle$	::= $\langle \text{condiție} \rangle$ ET $\langle \text{secvență-ET} \rangle$
	::= $\langle \text{condiție} \rangle$
$\langle \text{condiție} \rangle$	::= $\langle \text{condiție-simplă} \rangle$
	::= $\langle \text{condiție-existentă} \rangle$
$\langle \text{condiție-simplă} \rangle$	::= $Z_1 \langle \text{op-alfanr} \rangle Z_j$
	::= $Z_1 \langle \text{op-alfanr} \rangle \text{'sir'}$
	::= $Z_1 \langle \text{op-alfanr} \rangle$ formal-mot DE $X_j$
	::= $Z_1 \langle \text{op-alfanr} \rangle \langle \text{car-alfa} \rangle \langle \text{calif-real} \rangle$

	::= $Z_1 \langle \text{op} \rangle U$
	::= $\langle \text{variab-nr} \rangle \langle \text{op-nr} \rangle \langle \text{expr-aritm} \rangle$
	::= $\langle \text{variab-nr} \rangle \langle \text{op-nr} \rangle \langle \text{formal-nr} \rangle \text{DE } X_i$
	::= $\langle \text{variab-nr} \rangle \langle \text{op-nr} \rangle \text{val-nr} \langle \text{calif-real} \rangle$
	::= $\langle \text{variab-nr} \rangle \langle \text{op} \rangle U$
	::= $X_i \langle \text{op} \rangle X_j$
	::= $X_i \langle \text{op} \rangle U$
	::= $\text{formal-mot DE } X_i \langle \text{op-alfanr} \rangle Z_j$
	::= $\text{formal-mot DE } X_i \langle \text{op-alfanr} \rangle \text{'sir'}$
	::= $\text{formal-mot DE } X_i \langle \text{op-alfanr} \rangle U$
	::= $\text{formal-nr DE } X_i \langle \text{op-nr} \rangle \langle \text{expr-aritm} \rangle$
	::= $\text{formal-nr DE } X_i \langle \text{op} \rangle U$
	::= $\text{val-nr } [\langle \text{calif-real} \rangle] \langle \text{op-nr} \rangle \langle \text{expr-aritm} \rangle$
	::= $\langle \text{car-simplă} \rangle [\langle \text{calif-real} \rangle] \langle \text{op} \rangle U$
$\langle \text{op-alfanr} \rangle$	::= $\langle \text{op-nr} \rangle$
$\langle \text{op-nr} \rangle$	::= =
	::= >
	::= >=
	::= <
	::= <=
$\langle \text{op} \rangle$	::= $\langle \text{op} \rangle$
	::= =
	::= ↑=
$\langle \text{condiție-exist} \rangle$	::= $\langle \text{tip-exist} \rangle \langle \text{citare-realizare} \rangle$
	::= $\langle \text{tip-exist} \rangle \langle \text{car-simplă} \rangle [\langle \text{calif-real} \rangle]$
$\langle \text{tip-exist} \rangle$	::= EXISTE
	::= PAS
$\langle \text{citare-realizare} \rangle$	::= $\langle \text{un} \rangle \langle \text{element-entitate} \rangle$
	::= $\langle \text{un} \rangle \langle \text{element-invers} \rangle$
	::= $\langle \text{un} \rangle \langle \text{element-inel} \rangle$
	::= $\langle \text{un} \rangle \langle \text{element-referință} \rangle$
	::= $X_i$
$\langle \text{element-entitate} \rangle$	::= $\text{ent-niv-1 } [X_i] [\langle \text{selecție-ent-niv-1} \rangle]$
	::= $\text{ent-niv-2 } [X_i] [\langle \text{selecție-ent-niv-2} \rangle] [\langle \text{calif-real} \rangle]$
$\langle \text{element-invers} \rangle$	::= $\text{inv-niv-1 } [X_i] [\langle \text{filtru-secv} \rangle]$
	::= $\text{inv-niv-2 } [X_i] [\langle \text{filtru-secv} \rangle] [\langle \text{calif-real} \rangle]$
$\langle \text{element-inel} \rangle$	::= $\text{inel } \langle X_i \rangle [\langle \text{filtru-secv} \rangle] [\langle \text{calif-real} \rangle]$
$\langle \text{element-referință} \rangle$	::= $\text{referință } [X_i] [\langle \text{calif-real} \rangle]$
$\langle \text{selecție-ent-niv-1} \rangle$	::= AVEC $\langle \text{cond-simplă} \rangle ; [\langle \text{filtru-secv} \rangle]$
	::= PAR $\text{id-car-cheie} ; [\langle \text{filtru-secv} \rangle]$
	::= $\langle \text{nr-realiz} \rangle$
	::= $\langle \text{filtru-secv} \rangle$
$\langle \text{selecție-ent-niv-2} \rangle$	::= $\langle \text{nr-realiz} \rangle$
	::= $\langle \text{filtru-secv} \rangle$
$\langle \text{filtru-secv} \rangle$	::= AYANT $\langle \text{condiție-compusă} \rangle ;$

⟨calif-real⟩	::= DE ⟨citare-realizare⟩ ::= DE bloc-niv-1
⟨citare⟩	::= DE bloc-niv-2 ⟨calif-real⟩ ::= ⟨cuantificator⟩ ⟨entitate⟩ ::= ⟨cuantificator⟩ ⟨invers⟩ ::= ⟨cuantificator⟩ ⟨inel⟩ ::= ⟨referință⟩ ::= $X_1$
⟨entitate⟩	::= ent-niv-1 [ $X_1$ ] [⟨selecție-ent-niv-1⟩] ::= ent-niv-2 [ $X_1$ ] [⟨selecție-ent-niv-2⟩] [⟨calificare⟩]
⟨referință⟩	::= referință [ $X_1$ ] [⟨calificare⟩]
⟨calificare⟩	::= DE ⟨citare⟩ ::= DE bloc-niv-1 ::= DE bloc-niv-2 [⟨calificare⟩]
⟨cuantificator⟩	::= ⟨un⟩ ::= ⟨tout⟩
⟨un⟩	::= UN ::= UNE
⟨tout⟩	::= TOUT ::= TOUTE
⟨nr-realiz⟩	::= nr-intreg-pozitiv ::= $Y_1$
⟨car-simplă⟩	::= ⟨car-alfa⟩ ::= val-nr
⟨car-alfa⟩	::= mot ::= lista-de-valori
⟨expr-aritm⟩	::= ⟨termen⟩ + ⟨expr-aritm⟩ ::= ⟨termen⟩ - ⟨expr-aritm⟩ ::= ⟨termen⟩
⟨termen⟩	::= ⟨factor⟩ * ⟨termen⟩ ::= ⟨factor⟩ / ⟨termen⟩ ::= ⟨factor⟩
⟨factor⟩	::= ⟨variab-nr⟩ ::= număr ::= (⟨expr-aritm⟩)
⟨variab-nr⟩	::= $Y_1$ ::= $W_1$
⟨ext⟩	::= EXT [⟨element-ext⟩]
⟨element-ext⟩	::= 'sir' ::= $Z_1$
⟨poziție⟩	::= nr-întreg-pozitiv
⟨îg-șir⟩	::= nr-întreg-pozitiv
⟨formal-nr⟩	::= formal-dilate ::= formal-binar

## ANEXA D

## DIFERENȚE FAȚĂ DE SOCRATE VI.5 FELIX C-256

<i>Tip</i>	<i>Text explicativ</i>
Noutate	Partajare bază de date în domenii.
Modificare	Interfață limbaje evaluate — SOCRATE-MINI
Noutate	Posibilitate aliniere caracteristici la subpagină. Vezi și descriere comutator /SA în manualul de operare.
Noutate	Caracteristică de tip DECIMAL.
Noutate	Caracteristică de tip FILLER.
Noutate	Opțiune CHAINE AYANT/ARRIERE la inel.
Modificare	Dicționar organizat după metoda arborilor de tip B. Chei totdeauna ordonate ; sensul de ordonare indicat de opțiunea AVANT/ARRIERE.
Noutate	Caracteristici numerice fracționare în FORMAL.
Noutate	Caracteristică de tip BINAIRE în FORMAL.
Absență	Netratată caracteristica de tip TEXTE.
Noutate	Variabile $W_1$ .
Noutate	Interfața SOCRATE-MINI — Limbaje evaluate. Cerere APPEL
Noutate	Interfață SOCRATE-MINI — MCR
Modificare	Fișier jurnal rapid pe disc. Vezi și utilitar RST în Manualul de operare.
Modificare	Fișiere utilizator pe orice suport. Cerere ATTACHE pentru definire fișier și OPEN.
Noutate	Cerere DETACHE pentru CLOSE fișiere.
Noutate	Introducerea operatorului relațional == (egal parțial) în condiția simplă
Noutate	Utilizare operatori relaționali >, >=, <, <=, == la AVEC
Noutate	Cererea EXEC permite salvarea contextului (variabilele $X_i$ )
Noutate	Calificarea cu o variabilă $X_1$ poate avea și forma $X_1$ . identificator
Noutate	Tratarea erorilor recuperabile. Variabila ERREUR
Absență	Netratată cerere DEPUIS.
Absență	Netratată cerere C.
Absență	Netratată opțiune TELQUE.
Modificare	La SOCRATE-MINI, operatorii aritmetici +, *, / sînt și separatori ai limbajului de manipulare ; în schimb, semnele ; și ' nu sînt.
Modificare	Numărul de imbricări pt. entități poate fi max 4, iar la SOCRATE Felix max 7
Modificare	În structură, un bloc nu poate conține decît caracteristici simple
Modificare	La redefinirea de cîmpuri în formale, poziția în formal pornește aici de la 0, iar la SOCRATE Felix de la 1.

- Modificare Comentariile nu pot ocupa decît linii sursă în întregime. Ele nu se pot insera în liniile cu instrucțiuni, cu delimitatori /\* și \*/
- Modificare Cererea D nu modifică variabila  $X_1$ , ci are rol numai în analiză
- Modificare La sintaxa  $M Z_1 (a) = M Z_j (b, c)$ , parametrii b și c nu sînt implicați, ca la SOCRATE Felix.
- Modificare La SOCRATE-MINI, BUFFER 0 semnifică zona articol pentru fișierul de ieșire ; ea poate fi scrisă în orice fișier utilizator deschis în prealabil cu ATTACHE și i se poate asocia un formal de nivel 1.
- Modificare La SOCRATE-MINI, cererile SUIVANT și REFAIRE au roluri identice : reluarea forțată a unei bucle POUR sau FAIRE.
- Modificare La SOCRATE-Felix, transferurile parțiale succesive în  $Z_1$  trebuiesc efectuate de la stînga spre dreapta, deoarece sînt însoțite de golirea părții de sfîrșit a variabilei Z. La SOCRATE-MINI pot fi efectuate în orice ordine, cu condiția să fi fost precedate de o cerere  $M Z_1 = U$ .

## ANEXA E

## EXEMPLU DE STRUCTURĂ

```

DEBUT
ENTITE 1000 angajat
DEBUT
    nume MOT 20 AVEC CLE FIN
    funcție MOT 15
    sex (2 1) (B F)
    stare-civilă (4 11) (căsătorit necăsătorit divorțat văduv)
    stare-politică (2 3) (PCR UTC)
    talie DECIMAL (1 V 2) DE 1.50 A 2.10
    salariu DE 1500 A 6000
    marca DE 1 A 1000 AVEC CLE UNIQUE FIN
    apartenența-atelier REFERE personal atelier DE UN atelier DE UNE secție
    naștere
    DEBUT
        data MOT 8
        loc MOT 20
    FIN
    ENTITE 10 copil
    DEBUT
        an-de-naștere DE 1950 A 2000
        ocupație (4 9) (preșcolar elev student incadrat)
    FIN
FIN
ENTITE 10 secție
DEBUT
    nume MOT 25
    șef-secție REFERE UN angajat
    ENTITE 15 atelier
    DEBUT

```



nume MOT 25  
 șef-atelier REFERE UN angajat  
 personal-atelier ANNEAU

FIN

FIN

tesa INVERSE TOUT angajat

FORMAL f-angajat

DEBUT

nume	MOT	20
marca	DILATE	4
funcție	MOT	10
sex	MOT	1
stare-politică	MOT	1
stare-civilă	MOT	1
talie	DILATE	2 V 2
salariu	DILATE	4
secție	DILATE	2
atelier	DILATE	2
data-naștere	MOT	8
loc-naștere	MOT	18

FIN

FORMAL f-copil-angajat

DEBUT

marca	DILATE	4
eof	MOT	0 3
FORMAL	10 f-copil	
DEBUT		
	ocupație	MOT 1
	an-de-naștere	DILATE 4

FIN

FIN

FORMAL f-secție-atelier

DEBUT

tip	MOT	1
eof	MOT	0 3
nume	MOT	9 15

FIN

FIN

## Manual de operare SOCRATE-MINI

## 1. INTRODUCERE

## Componentele SOCRATE-MINI

Componentele SOCRATE-MINI sînt :

*GFM* -- *generatorul schemei de memorare* -- realizează :

– generarea bazei de date — constă în reținerea schemei de memorare (ce conține informații despre domeniile și fișierele componente ale bazei de date) într-un fișier director al bazei de date ;

– scrierea inițială (formatarea) domeniilor.

– *DFS* — *procesorul pentru definirea structurii* — realizează :

– catalogarea inițială a unei structuri ;

– adăugarea de noi caracteristici la structură ;

– editarea unei liste care ilustrează modul de implantare al caracteristicilor structurii în spațiul virtual.

– *SO* — *procesorul limbajului de manipulare* — realizează accesul la baza de date :

– încărcarea inițială a bazei de date ;

– actualizarea bazei de date ;

– regăsirea unor informații din baza de date.

În particular, prin utilizarea macroinstrucțiunilor și a dialogului la terminal, SOC permite crearea unui limbaj apropiat de cel uman, pentru accesul la bază al utilizatorilor nespecialiști.

– *MGS* — *programul bibliotecar* — realizează gestiunea domeniului de programe din baza de date :

– catalogarea, ștergerea și listarea macroinstrucțiunilor ;

– catalogarea, ștergerea și listarea programelor precompilate.

– *Programele utilitare pentru controlul sesiunii SO .RATE-MINI* realizează :

– *LGI* — deschiderea unei sesiuni ;

– *LGO* — închiderea unei sesiuni ;

– *ABS* — abortarea procesoarelor SOCRATE-MINI.

– *Programele utilitare pentru întreținerea bazei de date* :

– *RSS* — restaurarea bazei de date după producerea unui incident, utilizînd fișierul de securitate rapid ;

– *STS* — editarea statisticii bazei de date și, la cerere, a schemei de memorare ;

– *SAR* — salvarea — restaurarea bazei de date în vederea schimbării spațiului real.

- LES.TSK, LESO.OBJ, LESO.SKL — produse pentru interfața Limbaje evaluate — SOCRATE MINI.
- SOLE.OBJ — modul obiect pentru interfața SOCRATE-MINI — Limbaje evaluate.
- RES.TSK, LERESO.OBJ, LERESO.SKL — produse pentru interfața de lucru într-o rețea de bază de date.
- SOMIFE.TSK — produs pentru interfața SOCRATE-MINI — SOCRATE Felix.

## Domenii. Schema de memorare

Baza de date poate fi partajată logic în mai multe spații virtuale distincte, numite *domenii*. Spațiul real, corespunzător unui astfel de spațiu virtual, se poate compune fizic din unul sau mai multe fișiere disc. Partajarea în domenii permite montajul parțial al bazei de date pe durata unei sesiuni, completarea dinamică a bazei de date cu noi caracteristici, o mai bună securitate și confidențialitate.

Informațiile despre domeniile componente ale bazei de date și despre fișierele disc corespunzătoare formează *schema de memorare*. Aceste informații sînt reținute prin intermediul programului GFM într-un fișier de disc (separat de baza de date) numit fișier director al bazei de date.

Fișierul director se crează la generarea bazei de date (apelul programului GFM cu comutator /GN). Ulterior, schema de memorare poate fi modificată cu programul GFM (comutator /GP) în sensul adăugării de noi domenii sau al schimbării spațiului real aferent domeniilor vechi.

În schema de memorare, la generarea bazei de date, sînt obligatorii patru domenii, cu numele STR, PRG, DIC și FIS. Ele vor conține următoarele informații :

- STR -- structura bazei de date ;
- PRG -- macroinstrucțiuni și programe precompilate, în forma intermediară ;
- DIC — un repertoriu al macroinstrucțiunilor și programelor ;  
-- tabelele de indecși pentru caracteristicile declarate chei ;
- FIS și domeniile utilizator — informațiile propriu-zise din baza de date.

În urma generării, fișierele componente ale bazei de date nu există fizic pe disc. Fișierele disc aferente unui domeniu sînt create abia la formatarea domeniului, prin intermediul programului GFM, comutator /FM sau /FP.

## Sesiune SOCRATE-MINI

Pentru a optimiza accesul la baza de date, pe de o parte, și pentru a permite accesul mai multor utilizatori simultan, pe de altă parte, operațiile de intrare-ieșire cu baza de date se execută utilizînd 31 buffere comune de lucru numite *cadre*. Acestea se găsesc într-o partiție de tip COM, cu numele „CADRE”.

Procesorul SOC, cel mai susceptibil de a fi apelat simultan de mai mulți utilizatori, este reentrant și are modulele înglobate într-o *bibliotecă partajabilă rezidentă*, bibliotecă ce trebuie instalată într-o partiție de tip COM, cu numele „SOCLIB”. În partiția de lucru, fiecare apel al procesorului SOC dintr-un terminal necesită o zonă de reentrare de 12—16 ko (în funcție de formatele utilizate).

O dată instalate cadrele și biblioteca partajabilă în partiții, se poate lansa o sesiune SOCRATE-MINI. O sesiune de lucru SOCRATE-MINI este un interval de timp cuprins între un apel al programului LGI și un apel al programului LGO, în care mai mulți utilizatori au acces simultan la o aceeași bază de date de la mai multe terminale. La un moment dat, pe sistem nu se poate lucra decât cu o singură bază de date.

La lansarea sesiunii, programul LGI realizează următoarele lucruri :

- stabilește prin dialog la terminal, domeniile cu care se va lucra pe parcursul sesiunii. Domeniile obligatorii în diversele cazuri sînt :

- domeniul STR la catalogarea structurii ;

- domeniile PRG și DIC la catalogarea macroinstrucțiunilor ;

- domeniile PRG, DIC și STR la catalogarea programelor precompilate.

- citește în memorie, în partiția CADRE, datele din schema de memorare despre domeniile active, din fișierul director.

- verifică dacă sînt montate fișierele aferente domeniilor active și reține în partiția CADRE poziția lor pe disc. Din acest motiv este important de știut că aceste fișiere trebuie să fie prezente pe toată durata sesiunii în aceeași poziție pe disc.

- opțional, stabilește prin dialog, condițiile în care se va lucra cu fișierul de securitate.

Programul LGO realizează :

- interzicerea apelului procesoarelor din acel moment ;

- scrierea ultimelor cadre din memorie în baza de date.

Fără apelul final al programului LGO, baza de date va rămîne incompletă și, foarte posibil, incoerentă.

Pe durata unei sesiuni, din motive de sincronizare a accesului la baza de date, un procesor SOCRATE nu va fi niciodată abortat cu ABO, ci cu programul ABS.

La un apel, un procesor SOCRATE (DFS, SOC sau MGS) utilizează spațiu disc sub formă de fișiere de manevră temporare, pe perifericul SY :. Spațiul necesar este de cca 3 ori mai mare decât dimensiunea fișierului sursă, în care toate macro-urile au fost expandate.

## Convenții de apel generale

Componentele S.G.B.D. SOCRATE-MINI respectă în general regulile celorlalte programe existente pe sistemele de operare din clasa MIX, MINOS, AMS, RSX-11M/PLUS etc. Deci ele pot fi apelate prin una din cele 6 metode cunoscute de la celelalte programe, avînd structura liniei de comandă construită conform regulilor stabilite pentru aceste clase de sisteme de operare.

- Printr-un specificator de fișier se înțelege următoarea structură :

per : [uic] nume-fișier.tip;ver

unde :

per :

este un periferic pe al cărui volum se află fișierul. Numele este dat de 2 caractere ASCII urmate opțional de 1 sau 2 cifre (octal) reprezentînd numărul unității urmate de „:”.

[uic] :

este codul de identificare al utilizatorului avînd următoarea formă :

[gru, nru]

unde : gru -- numărul grupului utilizatorului

nru -- numărul utilizatorului în cadrul grupului

Amîndouă sînt numere octale cuprinse între 0–256.

*nume-fișier* :

numele fișierului are lungimea pînă la 9 caractere alfanumerice, numele și tipul fișierului sînt separate de „.”.

*tip* :

tipul fișierului este una din componentele ce pot identifica un fișier. Tipul fișierului poate avea cel mult 3 caractere, el este separat de versiune prin „:”.

*ver* :

este un număr octal utilizat pentru deosebirea diferitelor versiuni. Versiunea are valori între 1 și 7777(8).

– În apelarea componentelor S.G.B.D. SOCRATE-MINI nu pot fi folosite fișiere de comenzi indirecte.

– În toate cazurile în care utilizatorii folosesc liste de specificatori de fișiere, listele au proprietatea : perifericul, uic-ul și tipul fișierului se propagă în josul listei pentru a furniza valori implicate pentru cîmpurile din specificatorii de fișiere.

– În cadrul unui specificator de fișiere ale utilizatorilor S.G.B.D. SOCRATE nu poate fi folosit „\*”.

– Regulile privind comutatorii specificatorilor de fișiere sînt cele general valabile în clasele de sisteme de operare amintite mai sus.

## 2. DFS – PROCESORUL DE DEFINIRE A STRUCTURII

### Introducere în DFS

*DFS* – procesorul limbajului de definiere a structurii – are ca scop catalogarea structurii în domeniul STR al bazei de date.

Funcțiile acestui procesor sînt :

– catalogarea inițială a unei structurii ; în acest caz domeniul STR trebuie să fi fost generat și formatat cu utilitarul GFM înainte deschiderii sesiunii în curs ;

– adăugarea unor noi caracteristici la o structură existentă ;

– editarea unei liste cu „geografia datelor” care să ilustreze implementarea caracteristicilor în spațiul virtual.

Înainte de apelul lui DFS, sesiunea SOCRATE trebuie deschisă și domeniul STR activat cu LGI.

## Comutatori

Comutator	Fiș la care se atașează	Poate fi negat	Valoare implicită	Funcție
/SA	intrare	DA	/SA	Solicită alinierea fiecărei caracteristici la subpagină nouă, dacă nu încapă în întregime în subpagina ce conține caracteristica precedentă. Dacă este negat, nu se realizează aliniere.
/CVF	intrare	NU	—	Arată că din fiecare articol al fișierului de intrare sînt luate în considerare doar primele 60 de caractere. Restul sînt considerate comentarii. În absența comutatorului, articolele fișierului de intrare sînt luate în considerare în întregime.
/MAP	listing	NU	--	Solicită editarea unei „geografii a datelor”, cu modul de implementare a caracteristicilor în spațiul virtual.
/SP	listing	DA	/SP	Arată că se dorește ca fișierul listing la care este atașat să fie listat la LP:.

## Apel

### Linia de comandă

>DFS fișier-listing/SW<sub>1</sub> = fișier-intrare 1/SW<sub>2</sub>, fișier-intrare2/SW<sub>2</sub>...

unde :

- „fișier-listing” este fișierul pe care sînt listate fișierele sursă de intrare, eventualele erori de sintaxă și eventuala geografie a datelor solicitată prin comutatorul /MAP ; tipul implicit al fișierului este LST ;
- „fișier-intrare i” sînt fișiere ce conțin sub formă de linii-sursă, caracteristicile ce se doresc catalogate ; tipul implicit al fișierelor este DEF ;
- SW<sub>1</sub> sînt comutatorii opționali /SP și /MAP ;
- SW<sub>2</sub> sînt comutatorii opționali /SA și CVF.

### Convenții de apel

- Fișierele de intrare nu sînt concatenate, ci sînt analizate și catalogate pe rînd ; de aceea, cu excepția primului, celelalte fișiere de intrare din linia de comandă trebuie să aibă forma cerută la adăugări și structură (să înceapă cu cuvîntul cheie „D” și nu „DEBUT”).
- Se presupune că se dorește catalogarea caracteristicilor în ordinea din linia de comandă ; de aceea, dacă în analiza unui fișier de intrare sînt detectate erori de sintaxă, fișierele care urmează sînt analizate dar nu catalogate.

- Fișierul-listing poate lipsi. În acest caz linia de comandă are forma :  
>DFS = fișier-intrare<sub>1</sub>/SW<sub>2</sub>, fișier-intrare<sub>2</sub>/SW<sub>2</sub>...
- Se poate cere editarea unei geografii a datelor pentru o structură catalogată anterior. În acest caz, linia de comandă are forma :  
>DFS fișier-listing/MAP
- Apelul prin intermediul unui fișier indirect „>DFS fișier“ nu este permis.

## Catalogarea inițială a unei structuri

În acest caz domeniul STR trebuie să fi fost generat și formatat cu utilitarul GFM, apoi sesiunea SOCRATE să fie deschisă și domeniul STR activat cu LGI.

Structura fișierului de intrare în acest caz este :

```
[ $FIS ]
DEBUT
.
.      caracteristici bază de date
.
.
[ .      caracteristici formale ]
.
FIN
[ *FIS ]
?
```

Etichetele virtuale \$FIS și \*FIS sînt opționale și au rol de comentariu. Și în mod implicit și în mod explicit, domeniul de bază al caracteristicilor bazei de date, la o catalogare inițială, este domeniul FIS. În interior se poate solicita ca anumite caracteristici ale bazei de date să aparțină la alte domenii, utilizînd etichete virtuale.

## Adăugarea de caracteristici la structură

În acest caz, domeniul STR conține deja o structură catalogată anterior.

Sesiunea SOCRATE trebuie să fie deschisă și domeniul STR activat cu LGI.

Structura fișierului de intrare în acest caz este :

```
[ $dom ]
D
.
.      caracteristici bază de date]
.
[ .      caracteristici formale]
.
FIN
[ *dom ]
?      Cuvîntul cheie „D“ arată că e vorba de o adăugare de structură.
```

Etichetele virtuale „\$dom” și „\*dom” definesc domeniul de bază al caracteristicilor. În mod implicit, acesta este domeniul FIS. Și într-un caz și în celălalt, în interior se poate solicita ca anumite caracteristici ale bazei de date să aparțină la alte domenii, utilizând etichete virtuale.

## **Alinierea caracteristicilor la subpagină. Comutatorul /SA**

Comutatorul /SA (Subpage Alignement), atașat unui fișier de intrare, arată că se dorește pentru toate caracteristicile bazei de date definite în acel fișier o astfel de implementare în spațiul virtual, încât să se evite încălecarea unei caracteristici simple pe două subpagini diferite.

Pentru aceasta, o caracteristică simplă care nu încapă în întregime în subpagina curentă, este aliniată la subpagină nouă. Fiecare realizare a unei entități este aliniată la subpagină. Blocurile sînt aliniate la subpagină.

Scopul urmărit este creșterea vitezei de lucru, deoarece se evită cazul cînd pentru citirea sau scrierea unei caracteristici sînt necesare două accese disc.

Prețul plătit este un consum suplimentar de spațiu virtual și, uneori, și de spațiu fizic. Pentru a minimiza și chiar evita complet acest consum suplimentar, pentru o structură dată, se poate solicita o „geografie a datelor” prin comutatorul /MAP și, după analiza informațiilor de acolo, se pot aranja caracteristicile simple în cadrul entităților într-o astfel de ordine încît să umple cît mai bine subpaginile.

Trebuie ținut cont că, la generarea unui domeniu utilizator (corespunzător unui spațiu virtual) cu utilitarul GFM, numărul implicit de subpagini într-o pagină este 14, ceea ce corespunde unei dimensiuni a subpaginii de 32 octeți. Dar dimensiunea subpaginii poate avea, pentru oricare din domeniile utilizator, 14 valori punctuale distincte, între 32 și 500 octeți (vezi utilitarul GFM).

## **Geografia datelor. Comutatorul /MAP**

Comutatorul MAP asociat fișierului listing are ca efect editarea pe acest fișier a unui tabel care ilustrează modul de implementare a caracteristicilor în spațiul virtual. Tabelul reflectă situația întregii structuri, chiar dacă ultima operație efectuată a fost numai o adăugare de caracteristici.

Dacă, utilizînd etichete virtuale, caracteristicile au fost definite pe mai multe domenii, atunci va fi editat cîte un tabel pentru fiecare domeniu. În fiecare tabel vor fi descrise numai caracteristicile implementate în spațiul virtual al domeniului respectiv.

Tabelul are mai multe coloane, cu următoarele semnificații :

*L.STR* (linie structură)

Structura este împărțită în linii de cîte 32 octeți. Liniile 0 și 1 sînt rezervate. Începînd cu linia 2, caracteristicile ocupă una, două sau mai multe astfel de linii. În această coloană este dat numărul primei linii ocupată de caracteristica respectivă, pentru a se putea face referiri la ea în coloanele „Frate-tată”, „Fii” și „Diverse”.



**IDENTIFICATOR**

Conține primele 18 caractere din denumirea caracteristicii.

**FRA-TA** „frate-tată)

Cînd conține valori pozitive, punctează către următorul „frate“, deci următoarea caracteristică de același nivel.

Cînd conține valori negative, înseamnă că e ultimul „frate“ pe nivelul respectiv. În acest caz valoarea punctează către „tată“, deci către caracteristica de nivel superior căreia caracteristica curentă îi este subordonată.

**FII**

Coloana este completată numai pentru caracteristici compuse : entități, blocuri și formale. Ea punctează către primul „fiu“, deci către prima caracteristică de nivel inferior.

**TIF**

Coloana prezintă codificat tipul caracteristicii. Codurile au următoarea semnificație :

- 1 -- entitate definită la primul nivel
- 2 -- entitate inclusă
- 3 -- INVERSE
- 4 -- inel ; tipul inelului reiese din coloana „DIMENS“ :
  - inelul „chaine simple“ ocupă 4 octeți.
  - inelul „chaine double“ ocupă 8 octeți.
- 5 -- referință ; tipul referinței reiese din coloana „DIMENS“ :
  - referința simplă ocupă 4 octeți.
  - referința către un inel „chaine simple“ ocupă 8 octeți.
  - referința către un inel „chaine double“ ocupă 12 octeți.
- 6 -- MOT
- 7 -- Valori numerice bornate : BINAIRE și DÉCIMAL. Caracteristica BINAIRE este tratată ca un caz particular al caracteristicii DÉCIMAL, cu un număr de poziții zecimale 0.
- 8 -- Lista de valori
- 9 -- Bloc
- 10 -- Formal
- 11 -- MOT formal
- 12 -- Valoare numerică DILATE în formal
- 13 -- Valoare numerică PACKE în formal
- 14 -- Valoare numerică BINAIRE în formal

**ORIGINE**

Coloana prezintă adresa de implantare, în octeți, a caracteristicii. La caracteristicile definite la primul nivel, este chiar adresa în spațiul virtual corespunzător domeniului. La caracteristicile subordonate, este adresa relativă față de caracteristica de nivel imediat superior. La entități este adresa de implantare a primei realizări. Adresa lanțului de biți de prezență, care precede realizările, este dată la coloana „DIVERSE“.

**DIMENS** (Dimensiune)

Coloana prezintă dimensiunea ocupată, în octeți, de caracteristica respectivă.

La entități, coloana prezintă dimensiunea totală ocupată astfel :

Dimens. lanț biți + Nr. realizări \* Dimens. realizare

CH (tip CHAINE la ANNEAU, sau tip cheie)

Coloana prezintă :

- pentru ANNEAU :
  - AV – Avant
  - AR – Arriere
- pentru chei :
  - U – Unique
  - D – Double
  - AV – Avant
  - AR – Arriere

#### OC. DIC (Ocupare DIC)

Pentru chei, prezintă necesarul de spațiu virtual, în domeniul DIC, în octeți.

#### DIVERSE

Coloana prezintă :

- pentru entități : – Nr. realizări
  - Adresa lanțului de biți (cînd lanțul de biți este prezent în acest domeniu).
- pentru inverse, inel, referință : numărul liniei de structură a caracteristicii asociate : entitatea inversată, entitatea referită (la referințe simple), referința, respectiv inelul corespunzător.
- pentru formale repetitive : numărul de repetiții
- pentru celelalte caracteristici : informații specifice, respectînd aproximativ sintaxa în care au fost definite.

## Mesaje de eroare apărute la terminal

Mesajele de eroare posibile la execuția procesorului DFS sînt de 2 tipuri :

– mesaje de eroare detectate în analiza liniilor sursă ale unui fișier de intrare ; în general sînt erori de sintaxă. Ele sînt scrise pe fișierul listing, iar la terminal apare în mod normal doar un rezumat al lor sub forma :

### \*\* ERORI DETECTATE : n

Aceste erori apar totuși la terminal dacă fișierul listing este absent în linia de comandă.

– mesaje de eroare apărute datorită unei execuții anormale a programului DFS. Mai jos vor fi detaliate numai acest tip de erori :

#### 1. ER.D EROARE SINTAXĂ ÎN LINIA DE COMANDĂ

Linia de comandă, la apelul procesorului DFS, nu este conformă cu sintaxa.  
 Acțiune : Reintroducerea liniei de comandă cu sintaxa corectă.

#### 2. ER.D APEL ÎN AFARA SESIUNIII

Sesiunea SOCRATE nu a fost deschisă cu LGI.  
 Acțiune : Se deschide sesiunea.

#### 3. ER.D DFS DEJA APELAT

Procesorul DFS este deja în execuție, apelat probabil din alt terminal.  
 Acțiune : se așteaptă terminarea execuției.

## 4. ER.D EROARE SGF PE FIȘIERUL DE INTRARE COD : --x

La citirea fișierului de intrare a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

## 5. ER.D EROARE SGF PE FIȘIERUL DE LISTARE COD : - x

La scrierea pe fișierul listing a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

## 6. ER.D EROARE SGF PE FIȘIER DE MANEVRĂ TEMPORAR COD : -x

Pe durata execuției, procesorul DFS utilizează un fișier de manevră temporar pe perifericul SY. În timpul lucrului cu acest fișier, a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

## 7. ER.D DOMENIU STR INACTIV

La deschiderea sesiunii SOCRATE cu LGI, domeniul STR nu a fost activat.

*Ațiune* : Se închide sesiunea cu LGO și se redeschide cu LGI, activîndu-se domeniul STR.

## 8. ER.D SPAȚIU REAL SATURAT PENTRU DOMENIUL STR

Spațiul fizic atașat domeniului STR la generarea bazei de date cu GFM este saturat.

*Ațiune* : se închide sesiunea cu LGO, se generează încă o dată domeniul STR cu un număr mai mare de blocuri, se formatează cu GFM, și se deschide sesiunea cu LGI, activîndu-se domeniul STR nou generat.

## 9. ER.D EROARE ÎN CITIRE-SCRIERE PE DOMENIUL STR COD : --x

La citirea sau scrierea în domeniul STR a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii. Dacă este o cauză hardware, se închide sesiunea, se generează și se formatează încă o dată domeniul STR cu GFM și se redeschide sesiunea. Dacă este o altă cauză, se înlătură.

## 10. ER.D „DEBUT“ INTERZIS PESTE O STRUCTURĂ EXISTENTĂ

Peste o structură deja catalogată se dorește o catalogare inițială, utilizînd cuvîntul cheie „DEBUT“ în fișierul de intrare.

*Ațiune* : Dacă structura veche nu mai e necesară, se formatează domeniul STR. Dacă se dorește păstrată, noua structură se poate adăuga înlocuind în fișierul de intrare cuvîntul cheie „DEBUT“ cu „D“.

## 11. ER.D „DEBUT“ SAU „D“ ABSENT

Fișierul de intrare nu începe cu cuvîntul-cheie „DEBUT“ sau „D“, și deci nu se poate alege funcția dorită : catalogare inițială sau adăugare de structură.

*Ațiune* : Corecți fișierul sursă.

## 12. ER.D ABSENȚĂ „?“

În mod normal, dacă după cuvîntul cheie „FIN“ de la sfîrșitul structurii lipsește „?“ , acesta este simulat. Dacă totuși este detectat sfîrșitul fișierului de intrare în timpul tratării liniilor sursă, este afișat acest mesaj de eroare.

*Ațiune* : Se analizează și se înlătură toate erorile de sintaxă de pe fișierul listing și se reia execuția programului.

### 13. ER.D STRUCTURĂ INCOERENTĂ

La editarea tabelii solicitată prin comutatorul /MAP se constată că într-o sesiune anterioară a fost catalogată incomplet structura, datorită neînchiderii sesiunii cu LGO.

*Ațiune* : Se închide sesiunea, se formatează domeniul STR și se recataloghează întreaga structură.

### 14. ER.D UIC FĂRĂ DREPT DE ACCES LA BAZĂ

Procesorul DFS a fost apelat de un utilizator fără drept de acces la bază.

*Ațiune* : Discuțați cu administratorul bazei de date.

### 15. ER.D UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI

Procesorul DFS a fost apelat pentru o catalogare de structură de către un utilizator care are numai drept de consultare a bazei.

*Ațiune* : Discuțați cu administratorul bazei de date.

## 3. SOC – PROCESORUL LIMBAJULUI DE MANIPULARE

### Introducere în SOC

SOC – procesorul Limbajului de manipulare – permite efectuarea de preluări asupra datelor conținute în bază. Toate tipurile de acces sînt posibile la baza de date :

- încărcarea bazei de date ;
- actualizarea bazei de date ;
- regăsirea unor informații din baza de date.

Înainte de apelul lui SOC, biblioteca partajabilă SOCLIB trebuie încărcată într-o partiție de tip COM și sesiunea deschisă cu LGL, activîndu-se domeniile necesare execuției.

### Comutatori

Comu- tator	Fiș la care se atașează	Poate fi negat	Valoare implicită	Funcție
1	2	3	4	5
/CVF/	intrare	NU	—	Arată că din fiecare articol al fișierului de intrare sînt luate în considerare doar primele 60 de caractere. Restul sînt considerate comentarii. În absența comutatorului, articolele fișierului de intrare sînt luate în considerare în întregime.

1	2	3	4	6
/DSB:n	listing	NU	/DSB:0	Definește, prin valoarea sa, maximul sumelor dimensiunilor bufferelor FCS necesare la un moment dat pentru lucrul cu fișiere externe bazei de date.
/DSR:n	listing	NU	/DSR:0	Definește, prin valoarea sa, maximul sumei lungimilor înregistrărilor logice (FORMALE) folosite la un moment dat.
/SL	listing	DA	/SL	Solicită listarea programului sursă pe fișierul de ieșire.
/ME	listing	DA	/-ME	Solicită listarea pe fișierul listing a textului expandat la toate apelurile de macro-uri.
/SP	listing	DA	/SP	Arată că se dorește ca fișierul listing la care este atașat să fie listat la LP:.

## Apel

### Linia de comandă

>SOC fișier-listing/SW<sub>1</sub> == fișier-intrare/SW<sub>2</sub>

unde :

- *fișier-listing* este fișierul pe care sînt listate fișierul sursă de intrare, eventualele erori de sintaxă și datele tipărite prin cererile de listare din program ; tipul implicit al fișierului este LST ;
- *fișier-intrare* este fișierul ce conține sub formă de linii-sursă, programul de cereri, scris în limbajul de manipulare a datelor ; tipul implicit al fișierului este SOC.
- SW<sub>1</sub> sînt comutatorii opționali /DSB, /DSR, /SL, /ME și /SP ;
- SW<sub>2</sub> este comutatorul opțional /CVF.

### Convenții de apel

- *Fișierul-listing* nu poate lipsi, deoarece pe acest fișier sînt afișate și informațiile dorite prin programul de cereri. Dacă se dorește absența listingului sursă, aceasta se poate obține utilizînd comutatorul /-SL.
- Apelul prin intermediul unui fișier indirect „>SOC fișier“ nu este permis.

## Dimensionarea zonei de memorie necesară. Comutatorii /DSB și /DSR

În partiția de lucru, fiecare apel al procesorului SOC necesită o zonă de reen-tranță de 12 ko. În funcție de necesitățile programului (lucru cu fișiere, FORMALE) zona aceasta poate crește pînă la 16 ko.

Dacă se lucrează cu fișiere externe bazei de date, trebuie indicate două lucruri :

– prin comutatorul /DSB (Define Storage Buffer) : dimensiunea bufferelor FCS ale fișierelor utilizate simultan ; se va indica valoarea maximă necesară la un moment dat, și nu valoarea maximă totală, deoarece un buffer este alocat în memorie numai între cererile ATTACHE și DETACHE. La dimensiunea bufferelor se vor adăuga câte 16 octeți pentru fiecare fișier, zona necesară procesorului. De exemplu, pentru un buffer uzual de 512 octeți, se va solicita o zonă de 528 octeți.

– prin comutatorul /DSR (Define Storage Record) : dimensiunea zonelor articol (deci a FORMALELOR) fișierelor utilizate simultan.

Dacă sînt utilizate zone de memorie de manevră, definite prin FORMALE, acestea se adaugă la cele necesare pentru zonele articol ale fișierelor (comutator /DSR).

Valorile definite prin comutatorii /DSB și /DSR sînt valori zecimale întregi. Suma valorilor indicate prin comutatorii /DSB și /DSR nu poate depăși 4 ko.

## Formatul fișierului de intrare. Comutatorul /CVF

Fișierul de intrare poate avea două formate posibile :

– format liber, în care liniile sursă pot fi de dimensiune variabilă (pînă la 120 caractere) ;

– format cartelă, în care liniile sursă au dimensiuni de 80 caractere, din care coloanele 61—80 sînt considerate comentarii. Acest lucru permite programatorului ca, în afara liniilor comentariu (indicate prin caracterele /\*), să își comenteze programul pe margine.

Programele sursă scrise în al doilea format vor fi anunțate prin comutatorul /CVF (ConVentional Format).

## Opțiuni de listare. Comutatorii /SL, /ME și /SP

Pentru toate produsele care analizează un fișier sursă de intrare, metoda standard de a renunța la un listing sursă care nu interesează este de a nu defini fișierul de listare. Aici metoda nu poate fi aplicată, deoarece acest fișier conține nu numai listarea programului sursă, ci și rezultatele execuției programului.

Evitarea listingului sursă, atunci cînd acesta nu interesează, se poate realiza utilizînd comutatorul /-SL sau /NOSL (NO Source Listing).

Comutatorul /ME (Macro Expansion) permite ca, la apelul unei macroinstrucțiuni, să fie afișat textul expandat al acesteia. Sînt listate expandările tuturor macroinstrucțiunilor apelate în program, inclusiv a celor apelate imbricat.

Comutatorul /SP (SPool) aplicat unui fișier de listare definit pe disc are ca efect activarea spooler-ului la sfîrșitul execuției programului, în vederea listării fișierului de ieșire la imprimantă LP :.

## Mesaje de eroare apărute la terminal

Mesajele de eroare posibile la execuția procesorului SOC sînt de 3 tipuri :

— mesaje de eroare detectate în analiza liniilor sursă ale unui fișier de intrare ; în general sînt erori de sintaxă. Ele sînt scrise pe fișierul listing, iar la terminal apare în mod normal doar un rezumat al lor sub forma :

### •• ERORI DETECTATE : n

— mesaje de eroare apărute în timpul execuției programului solicitat, de exemplu : avertisări date de rutinele de validare automată la introducerea caracteristicilor în baza de date. Aceste mesaje sînt afișate numai pe fișierul de listare și execuția programului continuă normal.

— mesaje de eroare apărute datorită unei execuții anormale a procesorului SOC. După afișarea acestor erori, procesorul își încetează execuția. Mai jos vor fi detaliate numai acest tip de erori :

#### 1. ER.R EROARE SINTAXA ÎN LINIA DE COMANDĂ

Linia de comandă, la apelul procesorului SOC, nu este conformă cu sintaxa.

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

#### 2. ER.R APEL ÎN AFARA SESIUNII

Sesiunea SOCRATE nu a fost deschisă în LGI.

*Ațiune* : Se deschide sesiunea.

#### 3. ER.R EROARE SGF PE FIȘIERUL DE INTRARE COD : -x

La citirea fișierului de intrare a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

#### 4. ER.R EROARE SGF PE FIȘIERUL DE IEȘIRE COD : -x

La scrierea pe fișierul listing a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

#### 5. ER.R EROARE SGF PE FIȘIER DE MANEVRĂ TEMPORAR COD : -x

Pe durata execuției, procesorul SOC utilizează 3 fișiere de manevră temporare pe perifericul SY :. În timpul lucrului cu unul din aceste fișiere, a fost detectată o eroare, cu codul x afișat, se caută cauza erorii și se înlătură.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

#### 6. ER.R EROARE SGF PE FIȘIER UTILIZATOR COD: -x

La citirea sau scrierea unui fișier utilizator (cu ATTACHE, LIRE, ECRIRE sau DETACHE), a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Ațiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

#### 7. ER.R EROARE PE FIȘIERUL DE SECURITATE

La scrierea pe blocul de control (primul bloc) al fișierului de securitate, a fost detectată o eroare ireparabilă.

*Ațiune* : Se închide sesiunea și se redeschide cu alt fișier de securitate.

## 8. ER.R DOMENIU INACTIV DOM :xxx

La deschiderea sesiunii SOCRATE cu LGI, domeniul xxx, necesar execuției, nu a fost activat.

*Acțiune* : Se închide sesiunea cu LGO și se redeschide cu LGI, activându-se domeniul respectiv.

## 9. ER.R SPAȚIU REAL SATURAT DOM :xxx

Spațiul fizic atașat domeniului xxx la generarea bazei de date cu GFM este saturat.

*Acțiune* : se închide sesiunea cu LGO, se generează încă o dată domeniul xxx cu un număr mai mare de blocuri, se formatează cu GFM, și se deschide sesiunea cu LGI, activându-se domeniul respectiv nou generat.

## 10. ER.R EROARE ÎN CITIRE DOMENIU DOM:xx FIS:yy COD:-zz

## 11. ER.R EROARE ÎN SCRIERE DOMENIU DOM:xx FIS:yy COD:-zz

La citirea sau scrierea în domeniul xx, în fișierul nr. yy, a fost detectată o eroare, cu codul zz afișat în corpul erorii.

*Acțiune* : Utilizând codul afișat, se caută cauza erorii. Dacă este o cauză hardware, se închide sesiunea, se generează și se formatează încă o dată domeniul respectiv cu GFM și se redeschide sesiunea. Dacă este o alta cauză, se înlătură.

## 12. ER.R NR.PLIU &gt; 127 DOM:xx

Pentru domeniul xx, raportul dintre dimensiunea spațiului virtual și a spațiului real este > 127.

*Acțiune* : Se închide sesiunea, se generează domeniul cu un spațiu real mărit, se formatează, se deschide sesiunea și se reia programul.

## 13. ER.R DOMENIUL STR VID

*Acțiune* : Se cataloghează mai întâi structura.

## 14. ER.R SAT.SP.VIRT.ALOCAT CHEII IN DIC

Zona de spațiu virtual alocată unei anumite chei în domeniul DIC este saturată, datorită prea multor actualizări pe cheia respectivă.

*Acțiune* : Se compactează domeniul DIC cu utilitarul SAR.

## 15. ER.R UIC FĂRĂ DREPT DE ACCES LA BAZĂ

Procesorul SOC a fost apelat de un utilizator fără drept de acces la bază.

*Acțiune* : Discutați cu administratorul bazei de date.

## 16. ER.R UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI

Procesorul SOC a fost apelat pentru o actualizare a bazei de date de către un utilizator care are numai drepturi de consultare.

*Acțiune* : Discutați cu administratorul bazei de date.

## 17. ER.R EXTENSIE DE MEMORIE IMPOSIBILĂ

Necesarul de memorie solicitat prin comutatorii /DSB și /DSR depășește 4,5 ko.

*Acțiune* : Se relansează procesorul SOC cu valorile /DSB și /DSR micșorate corespunzător.



**18. ER.R VALOARE /DSR INSUFICIENTĂ**

Necesarul de memorie pentru formalele utilizate simultan în program este mai mare decât valoarea solicitată prin comutatorul /DSR.

*Ațiune* : Se relansează procesorul SOC cu valoarea /DSR mărită corespunzător.

**19. ER.R VALOARE /DSB INSUFICIENTĂ**

Necesarul de memorie pentru bufferele fișierelor utilizate simultan este mai mare decât valoarea solicitată prin comutatorul /DSB.

*Ațiune* : Se relansează procesorul SOC cu valoarea /DSB mărită corespunzător.

**20. ER.R nume-task-fiu NŪ POATE FI APELAT COD: —x**

Un subprogram scris într-un limbaj evoluat (task-fiu), solicitat printr-o cerere APPEL, nu poate fi lansat în execuție.

*Ațiune* : Utilizând codul x, afișat în corpul erorii, se caută cauza erorii și se înlătură.

**21. ER.R nume-task-fiu TERMINAT ANORMAL**

Un subprogram scris într-un limbaj evoluat (task-fiu), solicitat printr-o cerere APPEL, s-a terminat anormal.

*Ațiune* : Se caută cauza terminării anormale a task-ului cu numele afișat în corpul erorii și, dacă e posibil, se înlătură.

## 4. MGS – PROGRAMUL BIBLIOTECAR

### Introducere în MGS

MGS — Programul bibliotecar — are ca scop întreținerea bibliotecii de macro-instrucțiuni și programe precompilate ce se găsește pe domeniul PRG.

Funcțiunile programului sînt :

- *inserarea* macroinstrucțiunilor ;
- *inserarea* programelor precompilate ;
- *ștergerea* macroinstrucțiunilor ;
- *ștergerea* programelor precompilate ;
- *listarea* macroinstrucțiunilor ;
- *listarea* programelor precompilate ;
- *listarea* completă ;

Înainte de apelul lui, sesiunea SOCRATE trebuie deschisă și domeniile DIC PRG și STR activate cu LGI. Biblioteca partajabilă SOCLIB trebuie să fie deja încărcată, într-o partiție de tip COM.

## Comutatori

Comutator	Fiș la care se atașează	Poate fi negat	Valoare implicită	Funcție
/CVF	intrare	NU	—	Arată că din fiecare articol al fișierului de intrare sînt luate în considerare doar primele 60 de caractere. Restul sînt considerate comentarii. În absența comutatorului, articolele fișierului de intrare sînt luate în considerare în întregime.
/ME	listing	NU	—	Solicită listarea pe fișierul de ieșire a expansiunii macroinstrucțiunilor, inclusiv a celor apelate imbricat.
/SP	listing	DA	/SP	Arată că se dorește ca fișierul listing la care este atașat să fie listat la LP:.

## Apel

### Linia de comandă

>MGS fișier-listing/SW<sub>1</sub> = fișier-intrare<sub>1</sub>/SW<sub>2</sub>, fișier-intrare<sub>2</sub>/SW<sub>2</sub>...  
unde :

- „fișier-listing“ este fișierul pe care sînt listate fișierele sursă de intrare, eventualele erori de sintaxă iar în cazul terminării corecte mesaje ce confirmă efectuarea operației cerute (inserare, ștergere de macroinstrucțiuni și/sau programe precompilate); tipul implicit al fișierului este LST;
- „fișier-intrare<sub>1</sub>“ sînt fișiere ce conțin sub formă de linii-sursă, macroinstrucțiuni, programe precompilate și comenzi de ștergere sau listare; tipul implicit al fișierului este SOC;
- SW<sub>1</sub> sînt comutatorii opționali /SP și /ME;
- SW<sub>2</sub> este comutatorul opțional /CVF.

### Convenții de apel

— Fișierele de intrare nu sînt concatenate, ci sînt analizate pe rînd; de aceea erorile apărute în execuția unei funcțiuni dintr-un fișier nu vor afecta analiza și execuția funcțiunilor din celelalte fișiere.

— Fișierul-listing nu poate lipsi. Dacă se dorește execuția procesorului MGS fără obținerea unui fișier de ieșire, se poate apela astfel:

>MGS NL: = fișier-intrare<sub>1</sub>/SW<sub>2</sub>, fișier-intrare<sub>2</sub>/SW<sub>2</sub>...

- Apelul prin intermediul unui fișier indirect „>MGS @ fișier“ nu este permis.

## Mesaje de eroare apărute la terminal

Mesajele de eroare posibile la execuția procesorului MGS sînt de 2 tipuri :

— mesaje de eroare detectate în analiza liniilor sursă ale unui fișier de intrare ; în general sînt erori de sintaxă. Ele sînt scrise pe fișierul listing, iar la terminal apare în mod normal doar un rezumat al lor sub forma :

**\*\* ERORI DETECTATE : n**

— mesaje de eroare apărute datorită unei execuții anormale a programului MGS. Mai jos vor fi detaliate numai acest tip de erori :

### 1. EROARE SINTAXĂ ÎN LINIA DE COMANDĂ

Linia de comandă, la apelul procesorului MGS, nu este conformă cu sintaxa.  
*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 2. EROARE : APEL ÎN AFARA SESIUNII

Sesiunea SOCRATE nu a fost deschisă cu LGI.  
*Acțiune* : Se deschide sesiunea.

### 3. EROARE : FIȘIER DE INTRARE NEGĂSIT

La OPEN pentru fișierul de intrare definit în linia de comandă a fost detectat codul de eroare IE.NSF (No Such File).

*Acțiune* : Se definește în linia de comandă un fișier de intrare corect.

### 4. EROARE : ARTICOL FIȘIER INTRARE > 120.

La citirea fișierului de intrare a fost detectat un cod de eroare IE.RBG, datorită unui articol mai mare de 120 octeți.

*Acțiune* : Se caută și se scurtează articolul în cauză, consultînd fișierul listing.

### 5. EROARE PE FIȘIERUL DE INTRARE : — x

La citirea fișierului de intrare a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Acțiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

### 6. EROARE PE FIȘIERUL DE LISTARE : —x

La scrierea pe fișierul listing a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Acțiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

### 7. EROARE PE FIȘIER DE MANEVRĂ : —x

Pe durata execuției, procesorul MGS utilizează 3 fișiere de manevră temporare pe perifericul SY:. În timpul lucrului cu aceste fișiere, a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Acțiune* : Utilizînd codul afișat, se caută cauza erorii și se înlătură.

### 8. EROARE — DOMENIU PRG NEACTIVAT CU LGI

La deschiderea sesiunii SOCRATE cu LGI, domeniul PRG nu a fost activat.  
*Acțiune* : se închide sesiunea cu LGO și se redeschide cu LGI, activîndu-se domeniul PRG.

### 9. EROARE : DOMENIUL PRG SATURAT.

Spațiul fizic atașat domeniului PRG la generarea bazei de date cu GFM este saturat.

**Acțiune :** se închide sesiunea cu LGO, se generează încă o dată domeniul PRG cu un număr mai mare de blocuri, se formatează cu GFM și se deschide sesiunea cu LGI, activându-se domeniul PRG nou generat.

#### 10. EROARE PE DOMENIUL PRG : —x

La citirea sau scrierea în domeniul PRG a fost detectată o eroare, cu codul x afișat în corpul erorii.

**Acțiune :** Utilizînd codul afișat, se caută cauza erorii. Dacă este o cauză hardware, se închide sesiunea, se generează și se formatează încă o dată domeniul PRG cu GFM și se redeschide sesiunea. Dacă este o altă cauză, se înlătură.

#### 11. EROARE : DOMENIUL DIC NEACTIVAT CU LGI

La deschiderea sesiunii SOCRATE cu LGI, domeniul DIC nu a fost activat.

**Acțiune :** se închide sesiunea cu LGO și se redeschide cu LGI, activându-se domeniul DIC.

#### 12. EROARE : DOMENIUL DIC SATURAT

Spațiul fizic atașat domeniului DIC la generarea bazei de date cu GFM este saturat.

**Acțiune :** Se închide sesiunea cu LGO, se generează încă o dată domeniul DIC cu un număr mai mare de blocuri, se formatează cu GFM și se redeschide sesiunea cu LGI, activându-se domeniul DIC generat.

#### 13. EROARE PE DOMENIUL DIC : —x

La citirea sau scrierea în domeniul DIC a fost detectată o eroare, cu codul x afișat în corpul erorii.

**Acțiune :** utilizînd codul afișat, se caută cauza erorii. Dacă este o eroare cauzată hardware, se închide sesiunea și se formatează domeniul DIC, după care se redeschide sesiunea.

#### 14. EROARE : DOMENIUL PRG NEFORMATAT

S-a executat formatarea domeniului DIC fără a formata și domeniul PRG.

**Acțiune :** se închide sesiunea, se formatează domeniul PRG și se deschide o nouă sesiune.

#### 15. EROARE : DOMENIUL DIC NEFORMATAT

S-a executat formatarea domeniului PRG fără a se formata și domeniul DIC.

**Acțiune :** se închide sesiunea, se formatează domeniul DIC și se deschide o nouă sesiune.

#### 16. UIC FĂRĂ DREPT DE ACCES LA BAZĂ

Procesorul MGS a fost apelat de un utilizator fără drept de acces la bază.

**Acțiune :** Discuțați cu administratorul bazei de date.

#### 17. UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI

Procesorul MGS a fost apelat pentru o catalogare de programe sau macro-uri de către un utilizator care are numai drepturi de consultare a bazei.

**Acțiune :** Discuțați cu administratorul bazei de date.

## 5. PROGRAME UTILITARE

### GFM – Program de generare și formatare a bazei de date

#### Introducere în GFM

Programul GFM (Generare ForMatare) este un program utilitar al S.G.B.D. SOCRATE-MINI care generează și/sau formatează Baza de date (total/parțial), descriindu-i 'Schema de memorare'.

Funcțiunile principale ale lui GFM sînt :

- *generare totală* a bazei de date ;
- *generare parțială* a bazei de date (generarea numai a anumitor domenii din baza de date) ;
- *formatare totală* a bazei de date ;
- *formatare parțială* a bazei de date (formatarea numai a anumitor domenii din baza de date).

#### Comutatori GFM

Comutatorii GFM și funcțiunile lor sînt prezentate în tabelul următor :

Comutator	Poate fi negat	Valoare implicită	Funcție
/GN[:n]	DA	/-GN	Indică cerere de generare totală n — reprezintă numărul de subpagini al paginii domeniilor utilizator. Dacă este negat nu se dorește generare. n are valoare implicită 14.
/GP	NU	---	Indică cerere de generare parțială. Nu poate apare simultan cu comutatorul /GN sau /-GN.
/FM	DA	/-FM	Indică cerere de formatare totală. Dacă este negat nu se dorește formatarea.
/FP	NU	---	Indică cerere de formatare parțială. Nu poate apare simultan cu comutatorul /FM sau /-FM.
/CO:m	NU	---	Indică numărul de blocuri contiguu atribuite fișierului la care este atașat comutatorul
/RW:g <sub>1</sub>	NU	---	Indică grupul de utilizatori care vor avea drept de actualizare a bazei.
/RO:g <sub>2</sub>	NU	---	Indică grupul de utilizatori care vor avea doar drept de interogare a bazei.

### Linia de comandă inițială

Comanda inițială are următorul format :

>GFM fisdir/SW<sub>1</sub>/SW<sub>2</sub>

- fisdir este specificatorul fișierului director al bazei de date de formă :  
per :[uic] nume-fișier.tip ; ver
- SW<sub>1</sub>, SW<sub>2</sub> sînt comutatori care pot lua următoarele valori :  
GN[:n], -GN sau NOGN, GP, FM, -FM sau NOFM, FP

### Generarea

#### Generarea totală

Dacă se dorește generarea tuturor domeniilor bazei de date sau dacă generarea se face pentru prima dată se folosește comutatorul /GN(:n) unde n reprezintă numărul de subpagini al paginii domeniilor utilizator : 1 = <n<=14.

Funcție de acest n subpaginile domeniilor utilizator vor avea următoarele dimensiuni :

n	Dimensiunea utilă a subpaginilor paginii domeniilor utilizator.
1	500 octeți
2	248 ..
3	164 ..
4	122 ..
5	96 ..
6	80 ..
7	68 ..
8	58 ..
9	52 ..
10	46 ..
11	40 ..
12	38 ..
13	34 ..
14	32 ..

Valoarea implicită pentru n este 14.

Dacă s-a solicitat generare totală se afișează la TI:

\*\* GENERARE TOTALĂ \*\*

și apoi apare :

>GFM>

solicitînd transmiterea liniilor de descriere a domeniilor.

#### Liniile de descriere a domeniilor

Linia de comandă pentru descrierea domeniilor are următoarea structură :

GFM>nume-domeniu = fis<sub>1</sub>/CO:m<sub>1</sub>...fis<sub>p</sub>/CO:m<sub>p</sub>

unde :

nume-domeniu — reprezintă numele domeniului care trebuie să fie format din 3 caractere RADIX-50.

fisk — descriptor de fișier component al domeniului

$m_k$  — numărul de blocuri contiguu al fișierului.

Dacă descrierea fișierelor componente ale unui domeniu nu se poate face pe o linie se poate continua cu descrierea și pe linia următoare astfel :

GFM>nume-domeniu = fis<sub>1</sub>/CO : m<sub>1</sub>, ..., fis<sub>d</sub>/CO : m<sub>p</sub>

GFM>nume-domeniu = fis<sub>p+1</sub>/CO : m<sub>p+1</sub>, ..., fis<sub>d</sub>/CO : m<sub>q</sub>

Terminarea descrierii domeniilor se face prin // (două slash-uri)

GFM>//

În urma analizării liniilor de descriere a domeniilor se atașează fiecărui domeniu

numărul de blocuri  $nbd = \sum_{i=1}^q m_i$  unde  $q$  este numărul de fișiere atașate domeniului

și  $m_i$  — numărul de blocuri atașat fiecărui fișier component.

Se calculează apoi numărul de blocuri dicționar  $nd$ .

$nd = nbd/4\ 000 + s$

unde  $s = 0$  dacă împărțirea s-a făcut exact și  $s = 1$  dacă împărțirea s-a făcut cu rest.

Numărul de blocuri utile al domeniului  $nbu$  se calculează astfel :

$nbu = nbd - nd$

Acest număr trebuie să fie număr prim. Dacă nu este atunci se calculează primul număr prim mai mic decât  $nbu$  și se atribuie lui  $nbu$  valoarea aceasta scăzându-se din dimensiunile fișierelor diferența rezultată și trecînd ca prim fișier al domeniului fișierul cu dimensiunea cea mai mare. O bază trebuie să conțină neapărat domeniile STR, PRG, DIC și FIS.

### Generare: parțială

Dacă se dorește să se genereze domenii suplimentare pentru o bază de date care are deja generate o parte din domenii sau se dorește regenerarea unor domenii deja existente se folosește c mutatorul /GP.

Acesta nu poate fi folosit simultan cu /GN(:n), /-GN, NOGN. Dacă s-a solicitat generarea parțială se afișează la TI :

\*\* GENERARE PARȚIALĂ \*\*

apoi apare

GFM>

solicitînd transmiterea liniilor de descriere a domeniilor.

### Condiții restrictive la generare

- Nu pot exista mai mult de 15 domenii componente ale unei baze de date.
- La generare, primul domeniu nu poate fi decât STR, DIC sau PRG.
- Indiferent de numărul de domenii numărul total al fișierelor componente ale unei baze de date nu poate fi mai mare de 15.
- Nu pot exista două fișiere componente ale bazei de date cu același nume chiar dacă celelalte atribute ale specificatorilor diferă.

- Fişierele trebuie să se afle pe periferice adresabile.
- Nu se admit nume logice de periferice decât SY: şi LB:
- Dacă unui domeniu i se atribuie un fişier cu un singur bloc, se consideră acest domeniu fără nici un bloc util, deci domeniul are spaţiul real ataşat lui saturat.
- Un domeniu nu poate avea mai mult de 500 000 blocuri.

### Terminarea fazelor de generare

La terminarea fazelor de generare (fie generare parţială fie generare totală) se afişează la TI: **\*\* GENERARE TERMINATĂ BINE \*\*** şi apoi lista componenţei bazei de date unde sînt descrise domeniile şi fişierele componente ale bazei de date.

### Comutatorii /-GN sau /NOGN

Dacă nu se doreşte execuţia fazei de generare ci trecerea direct la faza de formatare se completează /-GN sau /NOGN.

### Formatarea

#### Formatarea generală

Dacă se doreşte formatarea sau reformatarea întregii baze de date se completează /FM.

Acest comutator nu poate apare simultan cu /FP sau /-FM sau /NOFM.

Se afişează la TI :

**\*\* FORMATARE GENERALĂ \*\***

şi apoi se trece la executarea formatării.

#### Formatarea parţială

Dacă se doreşte formatarea anumitor domenii ale bazei de date se completează /FP. Acest comutator nu poate apare simultan cu /FM sau /-FM sau /NOFM. Se afişează la TI :

**\*\* FORMATARE PARŢIALĂ \*\***

şi apoi se solicită numele domeniilor de formatat tipărindu-se :

GFM>

Se completează pe cîte o astfel de linie numele unui domeniu dorit a fi formatat urmînd ca sfîrşitul listei de domenii de formatat să se comunice tastînd // (două slash-uri) :

GFM>//

### Terminarea fazelor de formatare

La terminarea fazelor de formatare se afişează TI :

**\*\* FORMATARE TERMINATĂ BINE \*\*** şi apoi lista componenţei bazei de date.



**Comutatori /-FM sau /NOFM**

Dacă nu se dorește execuția fazei de formatare se completează comutatorul /-FM sau NOFM.

**Drepturi de acces la bază. Comutatorii /RW și RO**

Dacă se dorește protejarea bazei de date, se pot acorda drepturi de acces în scriere și citire unui anumit grup de utilizatori, comunicînd prin intermediul comutatorului /RW :  $g_1$  numărul grupului din UIC-ul lor [ $g_1$ , \*].

În același timp se pot acorda drepturi numai de citire din bază unui alt grup de utilizatori, prin intermediul comutatorului /RO :  $g_2$ ,  $g_2$  reprezentînd grupul din UIC-ul lor [ $g_2$ , \*].  
 $g_1$  și  $g_2$  iau valori cuprinse între 0–376 (8).

Dacă  $g_1$  și  $g_2$  au valoarea 0, se indică acordarea drepturilor respective tuturor utilizatorilor.

Aceste drepturi se acordă inițial la generarea totală a bazei.

Ulterior, valorile  $g_1$  și  $g_2$  se pot modifica, inclusiv cu valoarea 0, pentru a acorda la toți utilizatorii drepturi de acces.

Comutatorii /RW și /RO nu pot fi utilizați decît împreună cu unul din comutatorii funcțiilor de bază : /GN, /GP, /FM, /FP. Dacă nici una din aceste funcții nu este necesară, modificarea valorilor  $g_1$  și/sau  $g_2$  se va solicita împreună cu o formatare fictivă.

**Exemple**

```
>GFM FD.DAT/GN/RW:10/RO:12
```

Se acordă drept de acces în scriere și citire utilizatorilor cu UIC [10,\*] și drept de acces în citire utilizatorilor cu UIC [12,\*].

```
>GFM FD.DAT/GN
```

Se acordă drept de acces în scriere și citire tuturor utilizatorilor.

```
>GFM FD.DAT/-GN/FP/RW:17
```

Se modifică drepturile de acces în scriere, acordîndu-se aceste drepturi utilizatorilor cu UIC [17,\*], iar drepturile de acces în citire rămîn neschimbate.

```
>GFM FD.DAT/-GN/FP/RW:0/RO:0
```

Se anulează protecția bazei ; din acest moment, toți utilizatorii au drepturi de citire sau scriere în bază.

```
>GFM FD.DAT/GP/FP
```

Nu se dorește modificarea drepturilor de acces acordate anterior.

```
>GFM FD.DAT/-GN/FP/RW:10/RO:100
```

```
** FORMATARE PARȚIALĂ **
```

```
GFM>||
```

Se dorește numai modificarea drepturilor de acces, fără alte funcții ale utilitarului GFM.

**Mesaje de eroare**

Mesajele de eroare GFM sînt :

**1. EROARE SINTAXĂ**

Comanda nu este conformă cu sintaxa.

Acțiune : Reintroducerea liniei de comandă cu sintaxa corectă.

## 2. MAI MULT DE UN SPEC.FIȘIER sau CONȚINE \*

Linia de comandă inițială conține mai mult de un specificator de fișier sau în cadrul specificatorului apare caracterul \*

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

## 3. COMBINARE SWITCH-URI INCORECTĂ

La completarea comutatorilor nu s-a ținut seama de restricțiile menționate mai sus.

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

## 4. DEVICE IMPROPRIU

Perifericul specificat nu îndeplinește condiția cerută (nu este direct adresabil sau are nume logic de periferic diferit de SY: și LB: ).

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

## 5. OPEN ERONAT PENTRU FIȘIERUL DIRECTOR

Nu poate fi deschis fișierul director, din una din următoarele cauze :

- nu există spațiu pentru el pe periferic
- fișierul este protejat împotriva accesului
- există o problemă de periferic
- volumul este nemontat
- catalogul de fișier nu există
- numele fișierului nu există în catalog.

*Acțiune* : Determină care situație a cauzat eroarea și o corectează. Reia rularea programului GFM.

## 6. READ ERONAT SAU FIȘIER DIRECTOR IMPROPRIU

Nu s-a putut executa citirea articolului fișierului director sau acesta nu întrunește condițiile necesare unui fișier director.

*Acțiune* : Trebuie regenerată total baza de date sau găsită o copie corectă a fișierului director.

## 7. LIPSEȘTE DOMENIU

Pentru linia de descriere a domeniilor nu s-a completat numele domeniului.

*Acțiune* : Reintroducerea liniei de descriere cu sintaxa corectă.

## 8. EROARE SINTAXĂ DOMENIU

Numele domeniului din linia de descriere a domeniului nu s-a completat corect.

*Acțiune* : Reintroducerea liniei de descriere cu sintaxa corectă.

## 9. MAI MULT DE 15 FIȘIERE

S-au transmis mai mult de 15 fișiere componente.

*Acțiune* : Se generează din nou baza de date avînd maxim 15 fișiere componente.

## 10. MAI MULT DE 15 DOMENII

S-au transmis mai mult de 15 domenii componente ale bazei de date.

*Acțiune* : Se generează din nou baza de date avînd maxim 15 domenii componente.

## 11. SCRIERE IMPROPRIE PE FIS.DIR.

Din motive independente de program nu s-a putut efectua scrierea articolului fișierului director.

*Acțiune* : Se reia generarea bazei de date.

12. UN DOMENIU ARE SPAȚIU REAL > 500.000 BL RELUAȚI GENERAREA  
Un domeniu are în componența sa mai mult de 500.000 blocuri.

*Acțiune* : Trebuie reluată generarea de la început fără a mai avea domenii cu mai mult de 500.000 de blocuri.

13. RELUAȚI FORMATAREA

O anumită cauză al cărei mesaj apare anterior nu permite terminarea corectă a formătărilor.

*Acțiune* : Se elimină cauza și se execută programul numai pentru faza de formare.

14. NU EXISTĂ DOMENIUL CERUT

În timpul fazei de formare parțială s-a solicitat formatarea unui domeniu inexistent.

*Acțiune* : Se tastează numele corect al domeniului.

15. OPEN IMPROPRIU PENTRU

Nu poate fi deschis în OUTPUT unul din fișierele componente ale bazei de date din una din următoarele cauze :

- nu există suficient spațiu contiguu pentru el
- există o problemă pe periferic
- volumul este nemontat
- catalogul de fișiere nu există.

*Acțiune* : Se elimină cauza și se reia programul.

16. EROARE SCRIERE

Nu poate fi făcută formatarea unui fișier component datorită uneia din cauzele enunțate la eroarea 15.

*Acțiune* : Se înlătură cauza și se reia programul.

17. CLOSE IMPROPRIU PENTRU

Nu poate fi închis unul din fișierele componente.

*Acțiune* : Se elimină cauza și se reia programul.

18. LIPSA DOMENIU STR

Lipsește domeniu STR (de structură).

*Acțiune* : Se reia programul introducând în componența bazei și domeniul STR.

19. LIPSA DOMENIU PRG

Lipsește domeniu PRG (de programe).

*Acțiune* : Se reia programul introducând în componența bazei și domeniul PRG.

20. LIPSĂ DOMENIU DIC

Lipsește domeniu DIC (de dicționare).

*Acțiune* : Se reia programul introducând în componența bazei și domeniul DIC.

21. LIPSĂ DOMENIU FIS

Lipsește domeniul FIS (domeniul implicit utilizator).

*Acțiune* : Se reia programul introducând în componența bazei și domeniul FIS.

**22. NU S-A GENERAT NICI UN DOMENIU**

Nu s-a generat nici un domeniu.

*Acțiune* : Se reia programul introducând în componența bazei domeniile dorite.

**23. MAI MULTE FIȘIERE CU ACELAȘI NUME**

Baza de date are în componență mai multe fișiere cu același nume.

*Acțiune* : Se reia generarea fără a mai introduce mai multe fișiere cu același nume.

**24. NR. SUBPAGINI PE PAGINA ERONAT**

S-a cerut un număr de subpagini pe pagina domeniilor utilizator în afara intervalului 1 –14.

*Acțiune* : Se reia generarea indicându-se care e numărul de subpagini dorit.

**25. SESIUNE DESCHISĂ**

S-a apelat GFM în timpul unei sesiuni SOCRATE-MINI, lucru nepermis.

*Acțiune* : Se închide sesiunea SOCRATE-MINI și se reapelează GFM.

**26. UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI**

A fost apelat utilitarul GFM de un utilizator fără drepturi de actualizare a bazei de date.

*Acțiune* : Obțineți de la administratorul bazei includerea în grupul celor ce au dreptul să folosească GFM.

**27. PRIMUL DOMENIU AL BAZEI POATE FI NUMAI STR, DIC SAU PRG**

A fost declarat ca prim domeniu al bazei un domeniu de date.

*Acțiune* : Se redescrive baza, declarându-se ca prim domeniu unul din domeniile STR, DIC sau PRG.

## **LGI – Program de deschidere a unei sesiuni SOCRATE-MINI**

### **Introducere în LGI**

Programul LGI (LoGIn) este un program utilitar al S.G.B.D. SOCRATE-MINI care deschide o sesiune de lucru SOCRATE-MINI,

O astfel de sesiune trebuie deschisă o singură dată de la unul din terminalele de la care se va lucra în timpul sesiunii (terminalul respectiv se numește terminalul administratorului sesiunii de lucru).

Funcțiunile principale a lui LGI sînt :

– *deschiderea unei sesiuni* de lucru SOCRATE-MINI verificînd existența fișierelor componente ale domeniilor cu care se dorește să se lucreze și drepturile de acces la ele ale administratorului sesiunii de lucru ;

– *crează fișierul jurnal rapid* dacă se solicită sesiune cu fișier de securitate ;

– *activează task-ul de marcare a timpului* dacă această sesiune se dorește a fi cu puncte de reluare la intervale de timp.

## Comutatorii LGI

Comutatorii LGI și funcțiunile lor sînt prezentate în tabelul următor :

Comutator	Poate fi negat	Valoare implicită	Funcțiune
/AT	NU	/AT	Indică activarea tuturor domeniilor bazei de date. Nu poate apare simultan cu comutatorul /AP[:x]. Indică activarea numai a anumitor domenii ale bazei de date. x poate lua valorile P sau N. Dacă x = P se dorește ca în continuare să se comunice domeniile ce urmează a fi activate, iar dacă x = N se dorește comunicarea domeniilor ce urmează a fi inhibitate. Valoarea implicită a lui x este N. Nu poate apare simultan cu comutatorul /AT.
/AP[:x]	NU	----	
/CO:n <sub>1</sub>	NU	----	Atribuire fișierului jurnal rapid n <sub>1</sub> blocuri contiguu. Cere ca crearea punctelor de reluare să se facă după n <sub>3</sub> scrieri în bază. Nu poate apare simultan cu comutatorul /TI:n <sub>3</sub> . Cere ca crearea punctelor de reluare să se facă la intervalul de n <sub>3</sub> minute. Nu poate apare simultan cu comutatorul /WR:n <sub>2</sub> .
/WR:n <sub>2</sub>	NU	----	
/TI:n <sub>3</sub>	NU	----	

## Linia de comandă inițială

Comanda inițială are următorul format :

>LGI fisdir/SW<sub>1</sub> [,fissec/SW<sub>2</sub>/SW<sub>3</sub>]

unde :

— fisdir este specificatorul fișierului director al bazei de date

— fissec este specificatorul fișierului jurnal rapid care nu se completează decât dacă sesiunea se dorește cu fișier de securitate.

Dacă se completează este obligatorie prezența unuia din comutatorii WR : n<sub>2</sub> sau /TI : n<sub>3</sub> și a comutatorului /CO : n<sub>1</sub>

## Activarea

## Activarea totală

Dacă se dorește activarea tuturor domeniilor bazei de date se folosește comutatorul /AT. Acest comutator este implicit.

Se tipărește la TI :

\*\* ACTIVARE TOTALĂ\*\*

## Activarea parțială

Dacă se dorește activarea numai anumitor domenii ale azei de date se folosește comutatorul /AP(:x) unde x poate lua valorile P sau N avînd valoarea implicită N. Dacă x = P se va tipări la TI :

\* DOMENII DE ACTIVAT \* și în continuare se vor solicita domeniile ce se doresc a fi activate. Dacă  $x = N$  se va tipări la TI :

\* DOMENII DE INHIBAT \* și în continuare se vor solicita domeniile ce nu se doresc activate. Numele domeniilor se completează unul câte unul pe câte o linie de comandă.

LGI>nmd

unde : nmd -- nume domeniu.

Sfârșitul listei domeniilor de activat sau de inhibat se comunică cu // (două slash-uri).  
LGI>//

#### Activarea cu fișier jurnal rapid

Activarea unei sesiuni SOCRATE-MINI cu fișier jurnal rapid se face completând și fissec din linia de comandă inițială împreună cu unul din cei doi comutatori specifici lui /WR :  $n_2$  sau /TI :  $n_3$  funcție de principiul după care dorim să fie create punctele de reluare și în mod obligatoriu de /CO :  $n_1$  unde  $n_1$  reprezintă numărul de blocuri contiguu atribuite fișierului de securitate,  $n_1 > 20$ .

##### *Punctele de reluare create la număr de scrieri*

Dacă se dorește crearea punctelor de reluare după un anumit număr de scrieri în baza de date se completează comutatorul WR :  $n_2$ , unde  $n_2$  reprezintă numărul de scrieri dorit după care să se declanșeze crearea punctelor de reluare,  $n_1 > n_2$ .

Comutatorul /WR :  $n_2$  nu poate apare simultan cu comutatorul /TI :  $n_3$ .

##### *Punctele de reluare create la intervale de timp*

Dacă se dorește crearea punctelor de reluare după un anumit număr de minute se completează comutatorul /TI :  $n_3$ , unde  $n_3$  reprezintă numărul de minute dorit, după care să se declanșeze crearea punctelor de reluare  $1 \leq n_3 \leq 1440$ .

#### Restricții de activare

Activarea cu fișier jurnal rapid declanșat la un anumit interval de timp se poate face numai dacă task-ul MARKTM este instalat.

#### Deschiderea sesiunii SOCRATE-MINI

După verificările necesare și după eventuala activare a task-ului MARKTM dacă toate condițiile sînt îndeplinite se tipărește la TI :

**\*\* SOCRATE LA DISPOZIȚIA DUMNEAVOASTRĂ \*\***

Abia după apariția acestui mesaj se poate considera sesiunea SOCRATE-MINI deschisă.

Observație : Pe timpul unei sesiuni SOCRATE-MINI, adică între execuția programelor LGI și LGO, este interzisă orice operație asupra fișierelor componente ale bazei de date (șter-

geri, schimbări de poziție). Nerespectarea acestei condiții poate avea repercusiuni grave, datorită faptului că pe parcursul unei sesiuni SOCRATE-MINI operațiile de I/O se fac la nivel fizic.

Este interzis de asemenea ca pe parcursul unei sesiuni SOCRATE-MINI să se opereze asupra partiției de tip COM „CADRE”.

## Mesaje de eroare

Mesajele de eroare ale programului LGI sînt următoarele :

### 1. EROARE SINTAXĂ FIS DIRECTOR

Există o eroare privitoare la fișierul director.

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 2. EROARE SWITCH FIS. DIRECTOR

S-a completat eronat unul din comutatorii fișierului director.

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 3. DEVICE IMPROPRIU FIS. DIRECTOR

Perifericul specificat nu îndeplinește condițiile cerute (nu este direct adresabil sau este un nume logic de periferic diferit SY : sau LB :).

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 4. OPEN ERONAT PENTRU FIȘIERUL DIRECTOR

Nu poate fi deschis fișierul director, una din următoarele situații poate exista :

— fișierul este protejat împotriva accesului

— există o problemă de periferic

-- volumul este nemontat

-- catalogul de fișiere nu există

— numele fișierului nu există în catalog.

*Ațiune* : Determină care situație a provocat eroarea și o corectează.

### 5. READ ERONAT SAU FIȘIER DIRECTOR IMPROPRIU

Nu se poate executa citirea articolului fișierului director, acesta nu întrunește condițiile necesare unui fișier director sau eroare de memorie în partiția „CADRE”.

*Ațiune* : Trebuie regenerată total baza de date sau găsită o copie proprie a fișierului director sau înlocuit blocul de memorie defect.

### 6. EROARE SINTAXA FISIERULUI SECURITATE

Există o eroare privitoare la fișierul de securitate.

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 7. DEVICE IMPROPRIU FIȘIERULUI SECURITATE

Perifericul specificat nu îndeplinește condițiile cerute (nu este direct adresabil sau este un nume logic de periferic diferit de SY : și LB :).

*Ațiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 8. EROARE SWITCH FIS. SECURITATE

Nu s-au completat corect comutatorii fișierului de securitate.

*Ațiune* ; Reintroducerea liniei de comandă cu sintaxa corectă.

### 9. OPEN IMPROPRIU PENTRU

Unul din fișierele componente ale bazei de date nu poate fi deschis datorită uneia din situațiile prezentate la eroarea nr. 4.

*Acțiune* : Determină care situație a cauzat eroarea și o corectează. Reia rularea programului LGI.

#### 10. FIȘIER NĒCONTIGUU

Unul din fișierele componente ale bazei de date nu este contiguu.

*Acțiune* : Se fac contiguu toate fișierele bazei de date și se reia rularea programului LGI.

#### 11. FIȘIER IMPROPRIU PENTRU LUCRU

Unul din fișierele componente ale bazei de date este impropriu pentru lucru datorită unor manevre greșite cu acest fișier.

*Acțiune* : Se reia generarea bazei de date sau se aduce o copie corectă a acestui fișier și se reia rularea programului LGI.

#### 12. EROARE SINTAXA

Eroare de sintaxă în linia de comandă.

*Acțiune* : Se reintroduce linia de comandă cu sintaxa corectă.

#### 13. DOMENIU INEXISTENT

În timpul activării parțiale se solicită activarea sau inhibarea unui domeniu inexistent.

*Acțiune* : Se reintroduce linia de comandă cu numele corect al domeniului.

#### 14. DOMENIU NEFORMATAT

Se dorește activitatea unui domeniu neformatat.

*Acțiune* : Se formatează domeniul sau se activează parțial baza de date inhibând acest domeniu.

#### 15. CLOSE IMPROPRIU PENTRU

Nu poate fi închis fișierul datorită unei cauze independente de program.

*Acțiune* : Se elimină cauza și se reia programul LGI.

#### 16. OPEN IMPROPRIU PENTRU FIS. SECURITATE

Nu poate fi deschis fișierul de securitate, fie datorită cauzelor descrise la eroarea 4, fie datorită faptului că nu există spațiu suficient pentru el.

*Acțiune* : Se elimină cauza ce a provocat eroarea și se reia apoi rularea programului LGI.

#### 17. EROARE SCRIERE FIS. SECURITATE

Eroare detectată în timpul scrierii fișierului de securitate datorită unor cauze independente de program.

*Acțiune* : Se elimină cauza ce a provocat eroarea și se reia apoi rularea programului LGI.

#### 18. CLOSE IMPROPRIU PENTRU FIS. SECURITATE

Nu poate fi închis fișierul de securitate datorită unor cauze independente de program.

*Acțiune* : Se elimină cauza ce a provocat eroarea și se reia apoi rularea programului LGI.

#### 19. NU SE POATE LANSA TASK-UL DE MARCARE A TIMPULUI

Nu poate fi activat task-ul de marcarea timpului : fie că task-ul MARKTM nu a fost instalat anterior, fie că nu există memorie suficientă pentru a putea fi activat.

*Acțiune* : Se elimină cauza care a provocat eroarea și se reia programul LGI.



**20. SESIUNEA DEJA DESCHISĂ**

Sesiunea a fost deja deschisă și nu este permisă o redeschidere a ei.

*Ațiune* : Se termină sesiunea curentă și a poi se redeschide alta.

**21. UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI**

A fost apelat utilitarul LGI de un utilizator fără drepturi de actualizare a bazei de date.

*Ațiune* : Obțineți de la administratorul bazei includerea în grupul celor ce au aceste drepturi.

**22. LGI DEJA ACTIV**

Programul LGI este deja în curs de execuție, apelat de un alt utilizator.

*Ațiune* : Discutați cu administratorul bazei de date.

## **LGO – Program de închidere a unei sesiuni SOCRATE-MINI**

### **Introducere în LGO**

Programul LGO (LoGOut) este un program utilitar al S.G.B.D. SOCRATE-MINI care închide o sesiune de lucru SOCRATE-MINI.

O astfel de sesiune trebuie închisă o singură dată de la terminalul de unde s-a deschis sesiunea SOCRATE-MINI (terminalul respectiv se numește terminalul administratorului sesiunii de lucru).

Funcțiunile principale ale lui LGO sînt :

– *Închiderea unei sesiuni* de lucru SOCRATE-MINI copiind cadrele existente în memorie în fișierele componente ale bazei de date, verificînd anterior dacă nu s-a declanșat procesul de creare al unui punct de reluare.

– *Închiderea unei sesiuni* de lucru SOCRATE-MINI copiind cadrele existente în memorie în fișierele componente ale bazei de date, fără verificarea declanșării procesului de creare al unui punct de reluare. În acest caz există posibilitatea ca baza de date să rămînă incoerentă logic.

### **Comutatorii LGO**

Comutatorii LGO (și funcțiunile lor) sînt prezentați în tabelul următor :

Comutator	Poate fi negat	Valoare implicită	Funcție
/TR	NU	---	Închide o sesiune de lucru SOCRATE-MINI copiind cadrele în fișierele bazei de date, verificînd dacă s-a declanșat procesul de creare al unui punct de reluare.
/AB	NU	---	Închide o sesiune de lucru SOCRATE-MINI copiind cadrele în fișierele componente ale bazei de date fără verificarea declanșării procesului de creare a unui punct de securitate.

### Linia de comandă inițială

Linia de comandă inițială are următoarea formă :

>LGO/SW

unde : /SW este unul din comutatorii /TR sau /AB pentru terminarea normală respectiv pentru abordarea unei sesiuni de lucru SOCRATE-MINI.

### Terminarea normală

Dacă se dorește terminarea normală a unei sesiuni SOCRATE-MINI (cu verificarea declanșării procesului de creare al unui punct de reluare) se folosește comutatorul /TR. În acest caz după copierea cadrelor se afișează la TI :

**\*\* SESIUNE TERMINATĂ NORMAL \*\***  
**LA REVEDERE**

La apariția acestor mesaje se poate considera închisă o sesiune SOCRATE-MINI.

OBS : La închiderea sesiunii se creează punct de securitate dacă sesiunea a fost deschisă cu fișier jurnal.

### Abortarea unei sesiuni SOCRATE-MINI

Dacă din diferite motive ne interesează la un moment dat abortarea unei sesiuni SOCRATE-MINI (buclă făcută de un program etc.), se folosește comutatorul /AB care permite abortarea unei sesiuni SOCRATE-MINI, menținând coerența structurală a bazei de date, dar negarantînd coerența logică. După copierea cadrelor în fișierele bazei de date se tipărește la TI :

**\* S-A ABORTAT SESIUNEA (NU SE GARANTEAZĂ COERENȚA LOGICĂ) \***

După apariția unui astfel de mesaj se poate considera sesiunea închisă.

OBS : La închiderea sesiunii nu se creează punct de securitate pentru a putea permite re poziționarea bazei, cu utilitarul RSS, în ultimul punct de securitate anterior închiderii sesiunii. Datorită riscurilor de incoerență logică a bazei aceasta este singura modalitate de lucru corectă.

### Mesaje de eroare

Mesajele de eroare ale programului LGO sînt următoarele :

#### 1. EROARE SINTAXĂ

Există o eroare de sintaxă în linia de comandă.

Acțiune ; Reintroducerea liniei de comandă cu sintaxa corectă.

#### 2. SESIUNE DEJA ÎNCHISĂ

Se dorește închiderea unei sesiuni SOCRATE-MINI care a fost anterior închisă.

### 3. CERERE DE CKPT ÎN CURS DE REZOLVARE SAU BAZA BLOCATĂ

S-a solicitat terminarea normală a unei sesiuni SOCRATE-MINI când există declanșat procesul de creare al unui punct de reluare, sau când accesul la baza de date a fost blocat cu o cerere BLOQUER.

*Acțiune* : Se așteaptă terminarea execuției tuturor programelor și apoi se reia apelarea programului LGO sau se reia apelarea programului LGO cu comutatorul /AB (asumându-și riscurile folosirii lui).

### 4. SESIUNEA POATE FI TERMINATĂ NUMAI CU ABORT

Indică imposibilitatea în acel moment de terminare normală a sesiunii de lucru SOCRATE-MINI.

*Acțiune* : Analog ca la eroarea 3.

### 5. EROARE M<sub>x</sub> PENTRU FIȘIERUL NR. z (ER : y).

Eroare la citirea sau scrierea unui bloc din fișierele bazei de date (x=4 citire, x=5, scriere, x=6 nu se poate scrie în primul articol al fis. de securitate).

y — reprezintă codul erorii corespunzătoare din I/O STATUS CODE.

z — reprezintă numărul fișierului din baza la care s-a produs eroarea.

*Acțiune* : Deoarece este o eroare hardware, singura soluție este restaurarea unei copii a bazei de date anterioară începerii sesiunii și reluarea sesiunii.

### 6. NU SE POATE ATRIBUI LUN (ER : x) PENTRU FIȘIERUL NR. y.

Nu poate să fie atribuit LUN pentru fișierul din baza de date cu numărul y

x — reprezintă codul erorii corespunzător din I/O STATUS CODE.

*Acțiune* : Se elimină eroarea și se reia sesiunea după restaurarea unei copii a bazei de date anterioară începerii sesiunii.

### 7. UIC FĂRĂ DREPT DE ÎNCHIDERE A SESIUNII

A fost apelat utilitarul LGO de un alt utilizator decât cel care a deschis sesiunea SOCRATE-MINI cu LGI.

### 8. SINCRONIZAREA ACCESELOR LA BAZA DE DATE NU SE MAI DESFĂȘORĂ NORMAL.

S-a solicitat terminarea normală a unei sesiuni când mecanismul de sincronizare a acceselor la baza de date este dereglat.

*Acțiune* : Se reia apelarea utilitarului LGO cu comutatorul /AB (>ALGO/AB).

## **ABS — Program de abortare a procesoarelor SOCRATE-MINI**

### **Introducere în ABS**

Programul ABS (ABort Socrate) este un program utilitar S.B.G.D. SOCRATE-MINI care permite abortarea unuia din procesoarele SOCRATE-MINI, SOC, DFS sau MGS. Datorită specificului S.G.B.D. SOCRATE-MINI este interzisă abortarea procesoarelor prin comanda ABO, abortarea unui procesor putându-se doar face cu utilitarul ABS.

Această restricție este valabilă și pentru programele în alte limbaje care apelează subprograme SOCRATE-MINI.

Nerespectarea acestei cerințe implică riscul ca întreaga sesiune să nu se poată continua.

Utilitarul emite analog lui ABO un mesaj de abortare la consola ce a lansat task-ul abortat.

Pentru utilizatorii privilegiați utilitarul se comportă aidoma cu comanda ABO a sistemului de operare.

Utilizatorii neprivilegiați pot aborta numai task-urile lansate de la terminalul lor (TI:).

### Linia de comandă

Linia de comandă are următoarea formă :

**>ABS [nume-task]**

unde *nume-task* este numele procesorului ce se dorește abordat și este format din 3—6 caractere. Dacă se dorește abortarea procesorului lansat de la TI : se transmite :

**>ABS DFS** sau **>ABS SOC**

Dacă se dorește abortarea unui procesor lansat de la o altă consolă din sistem se transmite :

**>ABS DFSTnn** sau **>ABS SOCTnn**

unde *nn* — reprezintă numărul consolei care a lansat procesorul.

Dacă *nume-task* lipsește, se consideră că se dorește abortarea task-ului cu numele *TTnn*, unde *nn* are aceeași semnificație ca mai sus.

Sînt respectate toate convențiile existente pentru comanda ABO a sistemului de operare.

### Mesaje de eroare

#### 1. APELARE ERONATĂ

A fost apelat utilitarul ABS sub forma **>RUN ABS**

*Ațiune* : Se instalează și se apelează sub forma **>ABS...**

#### 2. APEL ÎN AFARA UNEI SESIUNI SOCRATE

A fost apelat ABS în afara unei sesiuni SOCRATE-MINI.

#### 3. UIC FĂRĂ DREPT DE ACCES LA BAZĂ

A fost apelat programul ABS de un utilizator fără drept de acces la bază.

#### 4. CERERE DE CKPT ÎN CURS DE REZOLVARE SAU BAZA BLOCATĂ

A fost apelat utilitarul ABS cînd este în curs de creare un punct de securitate sau baza a fost blocată sau este în lucru utilitarul SAR.

*Ațiune* : Se așteaptă terminarea procesului ce a generat mesajul de eroare și se reia apelarea utilitarului sau se termină sesiunea cu comanda **>LGO /AB**.

#### 5. SINCRONIZAREA ACCESELOR LA BAZA DE DATE NU SE MAI DESFĂȘORĂ NORMAL

Mecanismul de sincronizare al acceselor la baza de date este dereglat.

*Ațiune* : Se închide sesiunea SOCRATE-MINI cu **>LGO /AB**.

**6. NUME TASK ERONAT**

Numele task-ului conține caractere non Radix-50.

*Acțiune* : Se tastează corect numele task-ului.

**7. TASK INACTIV**

Task-ul ce se dorește abortat este inactiv.

*Acțiune* : Se tastează corect numele task-ului.

**8. TASK NEINSTALAT**

Task-ul ce se dorește abortat nu este instalat.

*Acțiune* : Se tastează corect numele task-ului.

**9. VIOLARE DE PRIVILEGIU**

Task-ul ce se dorește abortat a fost lansat de la alt terminal, iar utilizatorul este neprivilegiat.

## **RSS – Program de restaurare a bazei de date după producerea unui incident**

### **Introducere în RSS**

Programul RSS (ReStore Socrate) este un program utilitar S.G.B.D. SOCRATE-MINI care permite restaurarea într-un punct de reluare, a bazei de date după producerea unui incident, dacă sesiunea s-a derulat cu fișier jurnal rapid.

Funcțiunea principală a lui RSS este :

*Restaurarea coerenței bazei de date după producerea unui incident prin aducerea ei la stadiul în care se afla atunci când s-a creat ultimul punct de reluare anterior producerii incidentului.*

Urmează ca din acel punct sesiunea să fie reluată normal prin LGI și apoi prin executarea programului utilizator.

Reluarea din acel punct pentru programele utilizator trebuie rezolvată la nivelul limbajului de manipulare de către fiecare utilizator în parte, funcție de specificul programului.

### **Linia de comandă inițială**

Comanda are următoarea structură :

>RSS fisdir, fissec

unde :

fisdir — este specificatorul fișierului director al bazei de date

fissec — este specificatorul fișierului de securitate creat în sesiunea în care s-a produs incidentul.

## Restaurarea într-un punct de reluare

După producerea incidentului care a dus la abandonarea unei sesiuni SOCRATE-MINI cu fișier jurnal rapid se poate restaura baza de date în punctul de reluare imediat anterior producerii incidentului.

Dacă sesiunea nu s-a desfășurat cu fișier jurnal rapid (sesiune de interogare) deci fără alterarea caracteristicilor bazei de date, nu este nevoie de executarea programului RSS, baza rămânând coerentă. Dacă s-a desfășurat o sesiune de lucru SOCRATE-MINI în care s-au modificat caracteristici din baza de date fără a se folosi fișierul jurnal rapid sau dacă la nivelul limbajului de manipulare programele nu au cerut crearea de puncte de reluare, sau au folosit în mod neadecvat instrucțiunile de creare a punctelor de reluare, există riscul ca baza să fie incoerentă structural sau logic, fără posibilitatea ca RSS să poată reface coerența bazei de date.

Este necesară dimensionarea corectă a fișierului jurnal rapid deoarece dacă acesta s-a umplut, cu toate că sesiunea se desfășoară normal în continuare, RSS nu poate readuce baza într-un punct de reluare.

Abia crearea unui nou punct de reluare permite S.G.B.D. SOCRATE-MINI să facă din nou posibilă folosirea RSS. Deci între momentul umplerii fișierului jurnal rapid și momentul creării unui nou punct de reluare orice incident duce la necesitatea reluării întregii sesiuni de lucru.

La terminarea fazei de reluare se tipărește la TI :

**\*\* RESTAURAREA TERMINALĂ NORMAL \*\***

În acest moment se poate considera că baza de date a fost adusă în punctul de reluare anterior producerii incidentului.

## Mesaje de eroare

Mesaje de eroare ale programului RSS sînt următoarele :

### 1. EROARE SINTAXA FIS. DIRECTOR

Există o eroare privitoare la fișierul director.

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

### 2. OPEN ERONAT PENTRU FIȘIERUL DIRECTOR

Nu poate fi deschis fișierul director, una din următoarele situații poate exista :

- fișierul este protejat împotriva accesului
- există o problemă de periferic
- volumul este nemontat
- catalogul de fișiere nu există
- numele fișierului nu există în catalog.

*Acțiune* : Determină care situație a cauzat eroarea și o corectează. Reintroduce linia de comandă.

### 3. READ ERONAT SAU FIȘIER DIRECTOR IMPROPRIU

Nu poate executa citirea articolului fișierului director, acesta nu întrunește condițiile necesare unui fișier director, sau eroare memorie în partiția „CADRE”.

*Acțiune* : Trebuie regenerată total baza de date sau găsită o copie corectă a fișierului director și apoi se introduce linia de comandă, sau se înlocuiește blocul de memorie defect.

## 4. LIPSEȘTE NUMELE FIȘIERULUI DIRECTOR

Fișierul director nu are nume.

*Acțiune* : Se reintroduce linia de comandă cu sintaxa corectă.

## 5. LIPSEȘTE NUMELE FIȘIERULUI DE SECURITATE

Lipsește numele fișierului de securitate.

*Acțiune* : Se reintroduce linia de comandă cu sintaxa corectă.

## 6. OPEN ERONAT PENTRU FIȘIERUL DE SECURITATE

Nu poate fi deschis fișierul de securitate datorită uneia din cauzele prezentate la eroarea 2.

*Acțiune* : Determină care situație a cauzat eroarea și o corectează. Reintroduce linia de comandă.

## 7. DOMENIUL xxx NU ESTE ACTIV

Unul din fișierele domeniilor xxx nu există sau este impropriu.

*Acțiune* : Se reia întreaga sesiune după refacerea bazei de date folosind o copie anterioară sesiunii.

## 8. EROARE SINTAXA

Există o eroare de sintaxă în linia de comandă.

*Acțiune* : Reintroduce linia de comandă cu sintaxa corectă.

## 9. LIPSEȘTE FIȘIERUL DE SECURITATE

În linia de comandă nu s-a specificat fișierul de securitate.

*Acțiune* : Reintroduce linia de comandă cu sintaxa corectă.

## 10. EROARE SINTAXA FIȘIER SECURITATE

Există o eroare de sintaxă privitoare la fișierul de securitate.

*Acțiune* : Reintroduce linia de comandă cu sintaxa corectă.

## 11. FIȘIERUL DE SECURITATE IMPROPRIU

Fișierul de securitate s-a indicat greșit.

*Acțiune* : Reintroduce linia de comandă cu fișierul de securitate corect.

## 12. BAZA DE DATE NU POATE FI FĂCUTĂ COERENTĂ

Incidentul s-a produs într-un moment din care baza de date nu mai poate fi făcută coerentă.

*Acțiune* : Se reia întreaga sesiune de lucru după restaurarea unei copii a bazei de date anterioară sesiunii.

## 13. SE ABANDONEAZĂ RESTAURAREA

Se semnalează că dintr-o cauză al cărei mesaj a apărut anterior se abandonează restaurarea.

*Acțiune* : Se execută operațiile conform indicațiilor de la mesajul de eroare respectiv.

## 14. APEL ÎN TIMPUL UNEI SESIUNI

Se încearcă restaurarea bazei într-un punct de reluare în timpul desfășurării unei sesiuni de lucru SOCRATE-MINI.

## 15. EROARE Mx PENTRU FIȘIERUL NR. z (ER : y)

Eroare la citirea sau scrierea fizică a unui bloc din fișierele bazei de date (x=4 citire, x=5 scriere).

y — reprezintă codul erorii corespunzătoare din I/O STATUS CODE.

z — reprezintă numărul fișierului din bază pe care s-a produs eroarea.

**Acțiune :** Deoarece este o eroare hardware singura soluție este restaurarea unei copii a bazei de date anterioare întregii sesiuni și reluarea în întregime a sesiunii.

16. NU SE POATE ATRIBUI LUN (ER : x) PENTRU FIȘIERUL NR. : y

Nu poate fi atribuit LUN pentru fișierul din baza de date cu numărul y.

x — reprezintă codul erorii corespunzătoare din I/O STATUS CODE.

**Acțiune :** Se elimină eroarea și se reia sesiunea după restaurarea unei copii a bazei de date anterioare începerii sesiunii.

17. UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI

A fost apelat utilitarul RSS de un utilizator fără drept de actualizare a bazei de date.

**Acțiune :** Obțineți de la administratorul bazei includerea în grupul celor ce au aceste drepturi.

## STS — Program de editare a statisticii bazei de date și a componenței ei

### Introducere în STS

Programul STS(Stati Stic) este un program utilitar S.G.B.D. SOCRATE MINI care emite statistica bazei de date.

Funcțiunile principale de STS sînt :

- emiterea statisticii bazei de date ;
- emiterea listei componenței bazei de date.

### Comutatorii STS

Comutatorii STS și funcțiunile sînt prezentate în tabelul următor :

Comu- tator	Poate fi negat	Valoare implicită	Funcțiune
/BA	NU	/BA	Indică emiterea statisticii pentru întreaga bază de date. Nu poate apare simultan cu comutatorul /DO.
/DO	NU	---	Indică emiterea statisticii pentru anumite domenii ale bazei de date. Nu poate apare simultan cu comutatorul /BA.
/SP	DA	/-SP	Indică acțiunea ca fișierul la care este atașat să fie listat la L/P:



### Linia de comandă inițială

Comanda inițială are următorul format :

>STS [fissts/SW<sub>1</sub>] [, fiscomp/SW<sub>1</sub>]=fisdir/SW<sub>2</sub>

unde :

- fissts — este specificatorul fișierului statistic avînd ca valoare implicită a tipului LST.
- fiscomp — este specificatorul fișierului componentei bazei de date avînd ca valoare implicită a tipului LST.
- SW<sub>1</sub> — este comutatorul /SP.
- SW<sub>2</sub> — este unul din comutatoarii /BA sau /DO.

### Editarea statisticii

#### Editarea statisticii întregii baze de date

Dacă se dorește editarea statisticii întregii baze de date se completează în linia de comandă inițială fișierul statistic și se atașează fișierului director comutatorului /BA sau /DO și se atașează nici un comutator. Se emite la TI : următorul mesaj :

**\*\* STATISTIC INTEGRAL \*\***

editîndu-se apoi fișierul statistic.

#### Editarea statisticii parțiale a bazei de date]

Dacă se dorește editarea statisticii numai anumitor domenii ale bazei de date se completează în linia de comandă inițială fișierul statistic și se atașează fișierului director comutatorului /DO. Se emite la TI : următorul mesaj :

**\*\* STATISTIC PARȚIAL \*\***

solicitîndu-se apoi lista domeniilor pentru care se dorește statistica. Se comunică domeniile pentru care se dorește statistica tastînd numele cîte unui domeniu pe o linie :

STS> xxx

Sfîrșitul listei domeniilor ce se doresc a fi editate se comunică tastînd //(două slash-uri) :

STS>//

Dacă se dorește ca fișierul statistic să fie listat la LP : se atașează comutatorul /SP.

#### Editarea componentei

Dacă se dorește editarea componentei bazei de date se completează în linia de comandă specificatorul fișierului de componentă, tastînd comutatorul /SP dacă se dorește listarea lui la LP

## Mesaje de eroare

### 1. EROARE SINTAXĂ

Există o eroare de sintaxă în linia de comandă.

*Ațiune* : Reintroduce linia de comandă cu sintaxa corectă.

### 2. OPEN ERONAT PENTRU FIȘIERUL DIRECTOR

Nu poate fi deschis fișierul director, una din următoarele situații poate exista :

- fișierul este protejat împotriva accesului
- există o problemă de periferic
- volumul este nemontat
- catalogul de fișiere nu există
- numele fișierului nu există în catalog.

*Ațiune* : Determină care situație a cauzat eroarea și o corectează. Reintroduce linia de comandă.

### 3. READ ERONAT SAU FIȘIERUL DIRECTOR IMPROPRIU

Nu s-a putut citi fișierul director sau acesta nu întrunește condițiile unui fișier director.

*Ațiune* : Se reface fișierul director. Reintroduce linia de comandă.

### 4. OPEN ERONAT PENTRU FIȘIERUL STATISTIC

Nu poate fi deschis fișierul statistic, una din următoarele situații poate exista :

- există o problemă de periferic
- volumul este nemontat
- catalogul de fișiere nu există
- nu există spațiu suficient pentru fișier.

*Ațiune* : Se elimină cauza care a provocat eroarea și se reintroduce linia de comandă.

### 5. OPEN ERONAT PENTRU FIȘIERUL COMPONENTA

Nu poate fi deschis fișierul de componentă al bazei de date din una din cauzele prezentate la eroarea 4.

*Ațiune* : Se elimină cauza care a provocat eroarea și se reintroduce linia de comandă.

### 6. EROARE SCRIERE (ER : x) PENTRU FIȘIERUL STATISTIC

Eroare detectată de SGF în scrierea fișierului statistic.

x — reprezintă codul erorii din I/O STATUS CODE.

*Ațiune* : Se elimină eroarea și se reintroduce linia de comandă.

### 7. EROARE SCRIERE (ER : x) PENTRU FIȘIERUL COMPONENTĂ

Eroare detectată de SGF, în scrierea fișierului de componentă

x — reprezintă codul erorii din I/O STATUS CODE.

*Ațiune* : Se elimină eroarea și se reintroduce linia de comandă.

### 8. NU SE POATE ATRIBUI LUN (ER : x) PENTRU FIȘIERUL NR. y DIN BAZĂ

Nu se poate atribui LUN pentru un fișier din componentă bazei de date.

x — reprezintă codul erorii din I/O STATUS CODE

y — reprezintă nr. fișierului din baza de date.

*Ațiune* : Se elimină eroarea și se reintroduce linia de comandă.

**9. EROARE M4 PENTRU FIȘIERUL NR. x (ER : y)**

Nu se poate citi din fișierul cu numărul x din baza de date.

y — reprezintă codul erorii din I/O STATUS CODE.

*Acțiune* : Se elimină eroarea și se reintroduce linia de comandă.

**10. OPEN ERONAT PENTRU**

Nu poate fi deschis unul din fișierele componente ale bazei de date datorită unuia din motivele amintite la eroarea 4 sau a lipsei numelui fișierului din catalogul de fișiere.

*Acțiune* : Se elimină eroarea și se reintroduce linia de comandă.

**11. FIȘIER NECONTIGUU**

Unul din fișierele componente ale bazei de date nu este contiguu.

*Acțiune* : Se face contiguu fișierul vizat și se reintroduce linia de comandă.

**12. FIȘIER IMPROPRIU**

Unul din fișierele componene ale bazei de date este impropriu pentru lucru.

*Acțiune* : Se aduce o copie a bazei de date cu fișierele componente corecte și se reintroduce linia de comandă.

**13. DOMENIUL xxx NEFORMATAT**

Se solicită statistica pentru un domeniu neformatat.

*Acțiune* : Se reintroduce linia de comandă fără a mai solicita statistica pentru domeniile neformate.

**14. DOMENIU INEXISTENT**

S-a solicitat statistica pentru un domeniu care nu există în componența bazei de date.

*Acțiune* : Se reintroduce nume domeniu corect.

**15. UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI**

A.fost apelat utilitarul STS de un utilizator fără drept de actualizare a bazei de date.

*Acțiune* : Obțineți de la administratorul bazei de date includerea în grupul celor ce au aceste drepturi.

## **SAR — Program de salvare — restaurare a bazei de date**

### **Introducere in SAR**

Programul SAR (SAlvare-Restaurare) este un program utilitar S.G.B.D. SOCRATE-MINI care are drept funcții *salvarea* (extragerea) informațiilor din baza de date într-un fișier secvențial și *restaurarea* acestora după o nouă formare.

Prin operațiile de salvare-restaurare se poate urmări unul din următoarele scopuri :

- schimbarea dimensiunii spațiului real al unui domeniu ;
- schimbarea dimensiunii subpaginii pentru domeniile utilizator ;
- compactarea domeniului DIC ;
- compactarea domeniului PRG.

Programul SAR se apelează numai după ce a fost deschisă sesiunea cu LGI și activate domeniile ce se doresc salvate sau restaurate. Dacă se dorește restaurarea unui domeniu, acesta trebuie să fi fost formatat înainte de deschiderea sesiunii.

### Comutatorii SAR

Comutator	Poate fi negat	Valoare implicită	Funcție
/SV	NU	----	Indică funcția de salvare.
/RS	NU	----	Indică funcția de restaurare.
/ALL	NU	----	Împreună cu comutatorul /SV are ca efect salvarea tuturor domeniilor active în sesiunea respectivă. Împreună cu comutatorul /RS are ca efect restaurarea tuturor domeniilor din fișierul secvențial.

### Apel

#### Linia de comandă inițială

```
>SAR spec-fis { /SV } [ /ALL ]
                { /RS }
```

unde :

- spec-fis este specificatorul fișierului secvențial în care se vor salva sau din care se vor restaura informațiile din baza de date ;
- /SV și /RS sînt comutatori ce indică funcția dorită. Trebuie indicat unul și numai unul din acești doi comutatori. Ei nu pot fi negați.
- /ALL este un subcomutator ce poate însoți comutatorul ce indică funcția (/SV sau /RS). Subcomutatorul /ALL este opțional ; nu poate fi negat și nu are valoare implicită.

Utilizarea comutatorului /SV cu /ALL are ca efect afișarea mesajului

**\*\* SALVARE TOTALĂ \*\***

urmată de salvarea tuturor domeniilor active în sesiunea respectivă.

Utilizarea comutatorului /RS cu /ALL are ca efect afișarea mesajului

**\*\* RESTAURARE TOTALĂ \*\***

urmată de restaurarea tuturor domeniilor ce au fost salvate în fișierul indicat în linia de comandă.

#### Linii de comandă pentru selectarea domeniilor

În absența subcomutatorului /ALL, este afișat unul din mesajele

**\*\* SALVARE PARȚIALĂ \*\***

sau

**\*\* RESTAURARE PARȚIALĂ \*\***

după care sînt cerute numele domeniilor dorite (trei caractere RADIX-50), cîte unul pe o linie, tipărindu-se :

SAR>

Nu se pot solicita pentru salvare sau restaurare decît domenii active.

Sfîrșitul listei domeniilor se comunică tastînd două slash-uri :

SAR>||

### Schimbarea spațiului real

Schimbarea spațiului real alocat unui domeniu la generare se poate realiza în ambele sensuri :

-- creșterea dimensiunii spațiului real, dacă acesta este pe cale de a fi saturat ;  
 -- micșorarea dimensiunii spațiului real, dacă acesta a fost supradimensionat la generare.

Operațiile care trebuie realizate pentru schimbarea spațiului real sînt :

- salvarea domeniului respectiv cu SAR, comutator /SV ;
- închiderea sesiunii cu LGO ;
- generarea domeniului cu GFM, comutator /GP, pentru a se indica noul spațiu real ;
- formatarea domeniului cu GFM, comutator /FP ; --
- deschiderea sesiunii cu LGI, pentru ca domeniul respectiv să devină activ ;
- restaurarea domeniului cu SAR, comutator /RS.

*Observație* —

La estimarea dimensiunii spațiului real, trebuie avut grijă ca raportul între spațiul virtual și spațiul real să nu fie mai mare de 128 (acesta este numărul maxim de plieri posibile). La catalogarea structurii este verificat implicit acest raport față de spațiul real generat în momentul respectiv pentru domeniile DIC, FIS și cele definite de utilizator prin etichete virtuale ; dar dacă ulterior dorim să micșorăm spațiul real al unui domeniu, raportul trebuie verificat de fiecare dată manual.

### Schimbarea dimensiunii subpaginii

Operațiile necesare în acest scop sînt următoarele :

- salvarea tuturor domeniilor cu SAR, comutator /SV cu /ALL ;
- închiderea sesiunii cu LGO ;
- ștergerea de pe disc a fișierului director al bazei de date, precum și a tuturor fișierelor ce compun domeniile acestea ;
- generarea bazei de date cu GFM, comutator /GN:n, unde n este noua valoare dorită (număr de subpagini în pagină).
- formatarea tuturor domeniilor cu GFM, comutator /FM ;
- deschiderea sesiunii cu LGI, cu activarea tuturor domeniilor ;
- restaurarea tuturor domeniilor cu SAR, comutator /RS cu /ALL.

*Observații*

1. Dacă inițial structura a fost catalogată cu comutator /SA (pentru alinierea caracteristicilor la subpagină), după schimbarea dimensiunii subpaginii și restaurare, caracteristicile în mare parte vor fi încălcate pe cite două subpagini ; ba mai mult, pierderile de spațiu virtual

realizate ca urmare a alinierii inițiale, se mențin și acum, deși sînt complet inutile. Din acest motiv, este bine ca estimarea dimensiunii subpaginii să fie făcută corect de la început și să nu fie modificată ulterior.

2. În general, o dată informațiile încărcate în domeniile utilizator, se poate realiza ulterior recatalogarea structurii, cu condiția să se păstreze modul de catalogare (/SA sau /-SA). În schimb, din momentul schimbării dimensiunii subpaginii, recatalogarea structurii cu comutator /SA este complet interzisă, deoarece în structură caracteristicile vor fi aliniate la noua valoare de subpagină, iar în bază informațiile rămîn aliniate la vechea valoare de subpagină, așa cum au fost încărcate.

### Compactarea domeniilor DIC și PRG

Operațiile necesare sînt cele prezentate. Diferența față de domeniul STR și cele utilizator este că salvarea-restaurarea domeniilor DIC și PRG are sens, și chiar este recomandabilă după un anumit interval de timp, fără o schimbare obligatorie a dimensiunii spațiului real.

Pentru domeniul DIC, compactarea este recomandabilă după încărcarea masivă a informațiilor în baza de date, în cazul în care cheile nu au fost încărcate secvențial, în ordinea valorilor (crescătoare sau descrescătoare) declarată în structură. După compactare, tabelele de indecși sînt umplute în mod optim, ceea ce duce la o ocupare mai bună a spațiului real și, în anumite cazuri, la un spor de viteză.

Pentru domeniul PRG, compactarea se recomandă la sfîrșitul etapei de programare a unei aplicații. În acest moment, de obicei spațiul virtual al domeniului PRG are multe goluri, ca urmare a introducerii intercalate de macroinstrucțiuni și programe precompilate, cît și a ștergerii diferitelor variante de programe. Aceste goluri în spațiul virtual au ca efect o ocupare proastă a spațiului real, iar în unele cazuri, pierderi de viteză.

Prin salvare-restaurare, sînt reintroduse în domeniul PRG nou formatat mai întîi toate macroinstrucțiunile, apoi toate programele precompilate, ceea ce duce la o ocupare optimă a spațiului virtual.

#### Observații

1. La salvarea și la restaurarea domeniului DIC, trebuie să fie activ și domeniul STR.
2. La salvarea și la restaurarea domeniului PRG, trebuie să fie activ și domeniul DIC.
3. Dacă se dorește compactarea simultană a domeniilor DIC și PRG, aceasta trebuie realizată printr-un singur apel al programului SAR. În caz contrar, se poate realiza o desincronizare între programele din domeniul PRG și dicționarul cu denumiri de programe din domeniul DIC.

### Mesaje de eroare

Mesajele de eroare SAR sînt :

#### 1. EROARE DE SINTAXĂ ÎN LINIA DE COMANDĂ

Linia de comandă, la apelul programului SAR, nu este conformă cu sintaxa.

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

#### 2. COMBINARE SWITCH-URI INCORECTĂ

Comutatorii /SV sau /RS sînt fie ambii prezenți, fie ambii absenți.

*Acțiune* : Reintroducerea liniei de comandă cu sintaxă corectă.

## 3. EROARE : DEVICE IMPROPRIU

Perifericul specificat în linia de comandă nu este bandă sau disc magnetic.

*Acțiune* : Reintroducerea liniei de comandă cu sintaxa corectă.

## 4. EROARE : APEL ÎN AFARA SESIUNII

Programul SAR a fost apelat în afara sesiunii SOCRATE-MINI.

*Acțiune* : Se deschide sesiunea cu LGL.

## 5. EROARE : FIȘIER NEGĂSIT

La restaurare, a fost detectat codul de eroare IE.NSF (No Such File) la OPEN pentru fișierul secvențial definit în linia de comandă.

La salvare, UFD-ul fișierului de salvare nu există.

*Acțiune* : Se definește în linia de comandă un specificator de fișier corect.

## 6. EROARE : FIȘIER DE RESTAURARE NECONFORM

La restaurare, fișierul definit în linia de comandă nu a fost obținut cu funcția de salvare a utilitarului SAR.

*Acțiune* : Se definește în linia de comandă un fișier de restaurare corect.

## 7. EROARE : FIȘIERUL DE RESTAURARE NU CONȚINE : xxx

Unul sau mai multe domenii solicitate spre restaurare nu au fost salvate în fișierul indicat în linia de comandă.

## 8. EROARE I/O PE FIȘIERUL DE SALVARE-RESTAURARE : -x

În crearea fișierului (la salvare) sau citirea acestuia (la restaurare) a fost detectată o eroare, cu codul x afișat în corpul erorii.

*Acțiune* : Utilizând codul afișat, se caută cauza erorii și se înlătură.

## 9. EROARE : DOMENIUL xxx INACTIV

Sesiunea este deschisă, dar domeniul xxx afișat în corpul erorii, este inactiv.

*Acțiune* : Se închide, apoi se redeschide sesiunea, activându-se și domeniul în cauză.

## 10. EROARE : DOMENIUL xxx SATURAT

Spațiul real este insuficient pentru restaurarea tuturor informațiilor salvate din domeniul respectiv.

*Acțiune* : Se închide sesiunea, se generează domeniul cu un spațiu real mărit (GFM, comutator /GP), se formatează (GFM, comutator /FP) și se redeschide sesiunea.

## 11. DOMENIUL xxx \*\* EROARE : NUMĂR PLIU &gt; 127.

Raportul dintre dimensiunea spațiului virtual și a spațiului real pentru domeniul respectiv este > 127.

*Acțiune* : Aceeași ca și cum domeniul ar fi saturat.

## 12. EROARE : DOMENIUL xxx NEFORMATAT

Domeniul xxx nu a fost formatat înainte de restaurare (nu este vid).

*Acțiune* : Se închide sesiunea, se formatează domeniul și se redeschide sesiunea.

## 13. EROARE : DOMENIUL xxx VID

La salvare, domeniul xxx este vid (formatat).

*Acțiune* : Nici una.

## 14. EROARE I/O PE DOMENIUL xxx : -y

La salvarea sau restaurarea domeniului xxx, a fost detectată o eroare cu codul y afișat în corpul erorii.

*Acțiune* : Utilizând codul afișat, se caută cauza erorii și se înlătură, după caz.

15. EROARE : UIC FĂRĂ DREPT DE ACTUALIZARE A BAZEI  
Utilitarul SAR a fost apelat de un utilizator care nu are drepturi de actualizare a bazei de date.

Acțiune : Obțineți de la administratorul bazei includerea în grupul celor care au aceste drepturi.

## 6. INTERFEȚE DE COMUNICARE

### Interfața Limbaje evaluate – SOCRATE-MINI

Compilarea sau asamblarea programelor apelante de subprograme SOCRATE se efectuează normal.

Editarea de legături (TKB) se face adăugînd la modulul obiect utilizator modulul obiect LESO.OBJ din catalogul SOCRATE [1,55].

Exemplu de catalogare program COBOL :

```
>TKB PROG=PROG, LB:[1,55]LESO,[1, 1]COBLIB/LB,RMSLIB/LB
```

Pentru COBOL există suplimentar un fișier LESO.SKL, în același catalog, care poate fi comunicat ca un al doilea SKL la execuția lui CBLMRG.

Exemplu de catalogare program FORTRAN :

```
>TKB PROG=PROG, LB:[1, 55]LESO,[1, 1]FOROTS/LB
```

*Observații*

— Din necesități de lucru au fost rezervate două *EVENT FLAG* locale și anume 15. și 16. care nu mai pot fi deci folosite de utilizatori în scopuri proprii.

— Dimensiunea lui LESO.OBJ este de circa 1,5 K0 și este obligatorie aranjarea sa în rădăcina programului.

### Interfața SOCRATE-MINI – Limbaje evaluate

Compilarea sau asamblarea subprogramelor apelate de SOCRATE se face normal.

Editarea de legături (TKB) se face adăugînd modulul obiect obținut după asamblare sau compilare la modulul obiect SOLE.OBJ aflat în catalogul SOCRATE [1,55].

Numele subprogramului se declară cu opțiunea TASK= a TKB-ului.

Trebuie comunicată și dimensiunea zonei de parametri de apel ai subprogramului prin opțiunea EXTSCCT = BUFSOC : nnnn (8).



Pentru subprogramele FORTRAN care folosesc *COMMON* pentru transmiterea parametrilor, sau pentru limbaje care au opțiuni similare nu mai este nevoie de transmis dimensiunea zonei de parametri de apel.

Exemplu de catalogare subprogram COBOL :

```
>TKB
TKB>PROG=LB:[1,55]SOLE, SY:[1, 100]PROG, LB:[1,1]COBLIB/LB
TKB>/
ENTER OPTIONS:
TKB>TASK=EXTRAD
TKB>EXTSCT=BUFSOC:4
TKB>||
```

Exemplu de catalogare subprogram FORTRAN :

```
>TKB
TKB>PROG=LB:[1,55]SOLE, SY:[1, 100]PROG, LB:[1,1]FOROTS/LB
TKB>/
ENTER OPTIONS:
TKB>TASK=EXTRAD
TKB>||
```

Deoarece de fapt, datorită memoriei virtuale reduse (64KO), legătura dintre programul apelant SOCRATE și subprogramele scrise în alte limbaje se face sub forma legăturii task-părinte-task-fiu este necesară instalarea task-ului obținut în urma editării de legături înaintea apelării sale, lucru posibil de realizat atât independent de programul SOCRATE, cât și din interiorul programului SOCRATE cu instrucțiunea APPEL MCR.

#### Observație

— Pentru subprogramele a căror execuție durează puțin (cca 0.01 s) și se apelează foarte des, este indicată fixarea lor în memorie anterior intrării în bucla de execuție.

## 7. INTERFAȚĂ PENTRU LUCRUL ÎNTR-O REȚEA DE BAZE DATE

### Considerații generale

Interfața pentru regăsirea datelor din baze de date eterogene distribuite într-o rețea a fost proiectată pentru sisteme de operare compatibile RSX-11M V4.0 sau MIX V1.0 avînd un soft de rețea compatibil cu DECNET V3.0 sau MININET V1.0.

Înainte de deschiderea unei sesiuni de comunicație trebuie instalate task-urile de rețea prin apelarea fișierului de comenzi indirecte de instalare.

După instalare se vor da următoarele comenzi de rețea :

```
>NCP SET SYS
>NCP SET EXE STA ON
>NCP SET OBJ 128 NAME RES$$$ COP n USER LOG VER m
>NCP SET LINE DUP-1 STA ON
```

unde :

n reprezintă numărul maxim de copii care se permit pentru procesorul de interfață RES, acest număr indicînd din cîte noduri poate fi apelată simultan baza din acest nod ;

l — numărul liniei ;

m — poate lua valorile ON sau OFF.

Dacă m ia valoarea ON se va face verificarea drepturilor de acces pentru toți utilizatorii din alte noduri care doresc să se conecteze la baza din acest nod.

Dacă m ia valoarea OFF nu se mai face, de către softul de rețea, verificarea drepturilor de acces.

Se observă deci că prin acest mecanism cel care determină cine are acces la o bază este administratorul acesteia.

Mai mult decît atît administratorul unei baze poate adăuga la mecanismul de verificare al drepturilor de acces al softului de rețea, pe cel propriu SGBD-ului SOCRATE-MINI.

La terminarea unei sesiuni de comunicație pe rețea sînt necesare următoarele comenzi :

```
>NCP CLE LINE DUP-1 ALL
>DMO NS:/DEV
>NCP CLE SYS
```

*Observații*

— Procesorul de interfață RES este un task care are cod tip obiect (object type code) = 128.

— Indicațiile privitoare la lucru sub soft de rețea, existente în acest manual, nu înlocuiesc în nici un caz documentația de rețea, care se presupune cunoscută.

## Interfața apelată din limbajele evolute

Compilarea sau asamblarea programelor care apelează subprograme SOCRATE-MINI ale unei baze de date aflate în alt nod al unei rețele se efectuează normal.

Editarea de legături (TKB) se face adăugînd la modulul obiect utilizator modulul obiect LERESO.OBJ din catalogul SOCRATE-MINI [1,55].

Exemplu de catalogare program COBOL :

```
>TKB
TKB>PROG=PROG, LB:[1,55]LERESO,[1,1]COBLIB/LB, RMSLIB/LB
TKB>|
Enter Options :
TKB>UNITS=30
TKB>!!
```

Pentru programele scrise în COBOL există suplimentar un fișier LERESO-SKL, în același catalog, care poate fi comunicat ca al doilea fișier SKL la execuția lui CBLMRG.

Exemplu de catalogare program scris în FORTRAN :

```
>TKB
TKB>PROG=PROG, LB:[1,55]LERESO,[1,1]FOROTS/LB
TKB>/
Enter Options :
TKB>UNITS=30
TKB>///
```

#### *Observații*

- Din necesități de lucru au fost rezervate trei EVENT FLAG locale : 12, 13, 14 care nu pot fi folosite de utilizatori în scopuri proprii.
- Pentru rețea au fost rezervate LUN-urile 20—30 care deci nu mai sînt la dispoziția utilizatorului, în plus la TKB trebuie folosită opțiunea UNITS=30.
- Dimensiunea lui LERESO.OBJ este de circa 1,5 Ko și este obligatorie aranjarea sa în rădăcina programului.

## **Interfața apelată din limbajul de manipulare**

Față de cele prezentate anterior această interfață nu presupune și alte decizii de operare de luat.

## **8. INTERFAȚA SOCRATE-MINI – SOCRATE FELIX**

### **Resurse necesare**

Din punct de vedere hardware, resursele necesare sînt :

- la Felix : o cale în CTQM ; se va selecta prin placheta CD10 poziția 16A :
  - cod EBCDIC ;
  - verificare prin CRC.
- la mini : interfața sincronă de tip DUP-11 :
  - SI-40 CORAL ;
  - TTS 107 la I-100.
- cablu de legătură bifilar între cele două cuploare ;
- două modem-uri, dacă cele două calculatoare se găsesc la distanță.

Obs. : În cazul legăturii directe (fără modem-uri) sînt necesare 4 fire de legătură :

- emisie ;
- recepție ;
- masă ;
- ceas.

Legătura este de tip half-duplex. Este utilizat protocolul TMM-UC.

Din punct de vedere software, resursele necesare sînt :

- la Felix : S.G.T. (Sistemul de Gestiune a Transmisiilor) ;
- la mini : driverul pentru alinierea interfeței sincrone la caracteristicile CTQM-ului, numit XWDRV, aparținînd pachetului TELEMINI.

## Generarea procesorului de interfață de la Felix

SGBD-ul SOCRATE Felix utilizat poate fi V1.5 CII, V1.6.R ITC sau orice altă versiune care respectă normele de apel ale interfeței COBOL-SOCRATE. Din acest motiv, procesorul de interfață de la Felix nu este furnizat în format IMT, ci trebuie link-editat împreună cu modulele obiect specifice versiunii respective de SOCRATE.

Suplimentar, se oferă posibilitatea ca variabilele de lucru  $X_1$ ,  $Y_1$ ,  $Z_1$  (eventual  $W_1$ , dacă e cazul), să fie globale, total sau parțial, pe durata unei sesiuni de legătură mini-Felix ; aceasta oferă facilitatea ca, la începutul execuției unui program precompilat din baza de date de la Felix, să fie restaurate valorile variabilelor de lucru rămase de la precompilatul precedent. După dorință, se poate renunța la această facilitate, din motive de viteză.

Obs. : Nu este posibil transferul automat al variabilelor de lucru între programul apelant de la mini și precompilatul apelat de la Felix. Acest lucru se poate realiza numai utilizînd formatul de legătură.

Cartelele de comandă necesare pentru generare sînt următoarele :

```

. XPROC *, DVT : AD, VS : vs-disc, FN : 'SOCMINI SOU/GENESOMI',MOD
. MOD &VERSOC :v &
. MOD &NRX:m &
. MOD &NRY:n &
. MOD &NRW:p &
. MOD &NRZ:q &
. MOD &DV:DVT:AD, VS:vs-disc &
. ENDMOD
. XPROC *, DVT:AD, VS:vs-disc, FN: 'BIBSOUSOSOU/LINKCOBO'/MOD
. MOD...
. MOD &TREE:... &
. ENDMOD
. ASSIGN S, DVT:AD, VS:vs-disc
. ASSIGN Q, DVT:AD, VS:vs-disc
. ASSIGN T, DVT:AD, VS:vs-disc
. LABEL S, FN: 'BAZDBAZ', AM:OFL
. LABEL Q, FN: 'CADRE', AM:OFL
. LINK DVT:AD, VS:vs-disc, LN:SOCMINI, GN:1, VN:0, FN:SOMIFE

```

unde :

- v semnifică versiunea SGBD-ului SOCRATE utilizat. Poate avea valorile :
  - v = 0 pentru SOCRATE V1.5 CII sau altă versiune, care nu utilizează variabilele  $W_i$  ;
  - v = 1 pentru SOCRATE V1.6.R ITC.

- Valoarea implicită este 0.  
 –  $m$  este numărul primelor variabile  $X_1$  ce se doresc a fi globale. Poate avea valori cuprinse între 0–15.  
 Valoarea implicită este 9.  
 –  $n$  este numărul primelor variabile  $Y_1$  ce se doresc a fi globale. Poate avea valori cuprinse între 0–30.  
 Valoarea implicită este 10.  
 –  $p$  este numărul primelor variabile  $W_1$  ce se doresc a fi globale. Poate avea valori cuprinse între 0–30.  
 Valoarea implicită este 14.  
 Dacă  $v = 0$ , parametrul NRW nu are sens.  
 –  $q$  este numărul primelor variabile  $Z_1$  ce se doresc a fi globale. Poate avea valori cuprinse între 0–15.  
 Valoarea implicită este 5.  
 La apelul procedurii LINKCOBO, în poziționarea parametrului formal TREE se va ține cont de faptul că numele segmentului rădăcină este SOMIN%00.

## Indicații de operare

Operațiile necesare utilizării interfeței SOCRATE-MINI-SOCRATE Felix sînt în ordine :

- Încărcarea driverului pentru interfața sincronă :
- >LOA XW:
- Instalarea task-ului de interfață :
- >INS SOMIFE
- Lansarea în execuție, la Felix, a procesorului de interfață :
- într-o partiție serie, prin intermediul cartelelor de comandă :
- ..        FETCH DVT:AD, VS:vs-disc, LN:SOCMINI, GN:1, VN:O, FN:SO-
- MIFE
- .        OPTION CS 'CF:nr-cadre, DS:mem-formale, UN:1'
- ,        RUN
- într-o partiție paralelă, prin intermediul intervenției operator EXC, cu condiția ca procesorul să fi fost catalogat în biblioteca sistemului,
- Z%BUS:
- OPI EXC nn
- nn EXECUTE SOMIFE, AN:cont, OPT
- nn OPTIONS CF:nr-cadre, DS:mem-formale, UN:1
- Execuția programului apelant în limbajul de manipulare SOCRATE-MINI.

# Aplicație complexă cu SCBD SOCRATE-MINI pentru activitatea de exploatare a flotei maritime

## 1. ANALIZA ACTIVITĂȚII. „URMĂRIREA OPERATIVĂ A ACTIVITĂȚII FLOTEI ȘI PROGRAMAREA TRANSPORTULUI MARITIM“

### Caracteristicile activității de transport maritim

Prin natura activității sale, o întreprindere de transport maritim are trăsături specifice, care o detașează de majoritatea celorlalte întreprinderi de transport și cu atât mai mult de alte întreprinderi cu specific diferit.

Caracteristicile principale care definesc activitatea unei astfel de întreprinderi privesc :

- **Aria geografică mare în care își desfășoară activitatea**

Practic navele care aparțin întreprinderii se află răspândite pe întreg globul, în diverse ipostaze ale activității unele se află în marș, transportând mărfuri pentru un anumit port, altele descarcă sau încarcă marfă într-un port, sau tranzitează prin unui din canalele existente pe mapamond, sînt nave care se întorc spre portul de reședință după efectuarea unui transport, alte nave efectuează reparații accidentale sau planificate în șantierelor navale.

- **Caracterul deosebit al tarifelor de transport**

Datorită angrenării sale în comerțul internațional o întreprindere de transport maritim este obligată să adopte o politică de piață în stabilirea traficului. Tarifele nu fac obiectul unui act normativ și variază conform raportului cerere/ofertă existent în momentul derulării lor. Această trăsătură a tarifelor (navlurilor) de transport implică o cunoaștere foarte exactă a factorilor economici care vor interveni pe parcursul efectuării transportului : cheltuieli portuare, taxe de trecere prin canale, consum combustibili, preț combustibil, durata încărcării descărcării navei etc.

În afara acestora trebuie ținut seama și de anumiți factori socio-politici care pot afecta costul transportului : războaie în zona de transport, greve în porturile în care navele urmează să opereze, posibilitățile de obținere a unei alte oferte de transport în zona unde nava descarcă etc.

- **Timpul scurt în care trebuiesc luate deciziile economice**

Cu toată aria mare de dispersare a flotei și cu toate că factorii implicați în luarea unei decizii de angajare a unui transport sînt atît de diverși, unii dintre ei fiind chiar dificil de cuantificat, timpul rezervat luării unor astfel de decizii este de 1–2 zile și cîteodată se reduce la cîteva ore.

- **Valoarea economică ridicată a unei decizii de transport**

Prin costurile ridicate de transport, prin valoarea mare a mărfurilor și a navelor care urmează să le transporte, valori care sînt de obicei în valută liber convertibilă, luarea unei decizii de angajare a unei nave are implicații economice deosebite pentru întreprindere.

Toate aceste caracteristici au impus întreprinderii de transport maritim luarea unor măsuri organizatorice care să permită buna desfășurare a activității sale.

În primul rînd selecționarea unor buni specialiști care să întrețină și să conducă navele flotei, pregătiți în școli de diverse grade, în institute de învățămînt superior, specialiști controlați și verificați periodic prin examene de profil.

În același timp au fost selecționați specialiști cu înaltă calificare care să cunoască cît mai bine condițiile de transport specifice diverselor zone ale globului, care să fie la zi cu tendințele manifestate pe piața internațională, cu toate informațiile necesare luării unei decizii corecte de angajare a navelor.

În afara politicii de personal a fost necesară organizarea unui sistem de informare zilnică despre activitatea fiecărei nave, informare strict necesară în condițiile economice prezentate anterior. În fiecare noapte, între orele 22–24 navele flotei maritime aflate în marș transmit prin radio date privitoare la activitatea zilei anterioare, evenimente deosebite de navigație etc.

Pentru navele aflate sub operațiuni în porturile românești, pînă la ora 04.00 a zilei următoare se raportează cum a decurs încărcarea/descărcarea navei. Navele care datorită faptului că se află în porturi străine nu pot transmite prin radio raportează evenimentele importante prin agenția din portul respectiv, iar la ieșire din port transmit un raport complet al activității desfășurate în ultimul port.

Datorită creșterii numărului de nave din subordine și a faptului că între diversele tipuri de nave (petroliere, mineraliere, cargouri, feriboturi) există mari deosebiri atît cu privire la caracteristicile tehnice ale lor cît și cu privire la condițiile contractuale ale angajării lor, flota a fost împărțită în secții de transport, secții care au în subordine, în general, nave cu caracteristici tehnico-economice apropiate.

Pe măsura dezvoltării comerțului internațional, a creșterii activității de import-export, activitatea de urmărire operativă a flotei a căpătat din ce în ce mai mult un caracter complex impunîndu-se introducerea prelucrării automate a datelor care să permită eliberarea specialiștilor întreprinderii de activitățile de rutină și să-i asiste în luarea unor decizii cît mai bune, pentru creșterea eficienței de transport maritim.

## **Stabilirea cerințelor întreprinderii**

O dată luată decizia de introducere a mijloacelor de prelucrare automată a datelor în sprijinul activității de urmărire operativă a flotei maritime s-a pus problema evidențierii acelor domenii care ar necesita stringent o astfel de prelucrare. Pentru

aceasta echipa de proiectare a analizat activitățile serviciilor funcționale legate de urmărirea operativă a activității flotei, îndeosebi atribuțiunile lor în buna desfășurare a acesteia.

Pe măsura desfășurării muncii de analiză s-a discutat cu specialiștii implicați în activitățile analizate, cu factorii de conducere, toate aspectele atât tehnico-organizatorice cât și cele socio-economice care apar pe parcursul desfășurării activității de conducere operativă a flotei maritime.

În urma încheierii unei prime faze de analiză s-au desprins principalele activități care prin specificul lor necesitau prelucrarea unor volume mari de date și în același timp necesitau un timp de răspuns mic. Acestea au fost :

- **Urmărirea operativă a activității navelor:**

Aceasta se desfășoară pe baza datelor transmise prin radiograme de nave între orele 22—24 ale fiecărei zile. Prin volumul lor și prin caracterul zilnic aceste date necesită un mare volum de prelucrare. În plus analiza activității unei nave sau unor tipuri de nave pe o perioadă mai mare de timp devenea prin caracterul anevoios de obținere și prelucrare a datelor, de multe ori, prohibită.

- **Urmărirea operării navelor în porturile românești**

Activitate importantă, care permite evaluarea datelor de plecare a navelor în voiaj, programarea operațiilor de aprovizionare și întreținere a navelor, planificarea unor recuperări date echipajelor.

- **Urmărirea activității de reparații planificate ale navelor.**

În această activitate se permite planificarea corectă a transportului de mărfuri pe măsura terminării reparațiilor, precum și programarea periodică a navelor ce urmează să fie reparate conform normelor tehnice.

- **Urmărirea activității navelor în porturi**

Activitate de mare importanță în luarea unor decizii corecte de angajare a navelor, deoarece prin urmărirea și analiza activității navelor în diversele porturi ale lumii se pot estima duratele de staționare în aceste porturi, cheltuielile portuare și de operare necesitate de o eventuală escală, precum și cauzele de întrerupere a activității de operare specifice porturilor respective.

- **Urmărirea consumului de combustibili**

Prin procentul ridicat din totalul cheltuielilor de transport, cheltuielile legate de consumul de combustibil se impun ca un factor important de care trebuie ținut seama la angajarea unei nave. Eventualele parcurgeri de traseu fără încărcătură trebuie minimizate, după cum și viteza de navigație se impune a fi adaptată condițiilor



specifice atât de ordin meteorologic cât și celor legate de tipul motorului sau de zona geografică în care se desfășoară transporturile.

Abaterile de la consumul optim vor fi evidențiate pentru a fi analizate, caz cu caz și astfel să se elimine cauzele ce le-au produs.

#### ● Stabilirea disponibilităților de transport

Activitate foarte importantă în procesul de conducere operativă a flotei maritime care necesită cunoașterea exactă a contractelor de transport încheiate și care se derulează în prezent, poziția actuală a navei, datele estimative privitoare la durata marșului și a activităților de încărcare/descărcare pe care navele le mai au de efectuat pînă la derularea completă a contractelor existente.

În urma acestei analize se poate stabili data la care nava este disponibilă și zona în care aceasta va deveni aptă să transporte o nouă încărcătură de mărfuri, informație necesară pentru a putea obține o ofertă de transport care să îndeplinească condițiile tehnico-economice necesare.

Pe de altă parte atunci cînd întreprinderea primește o ofertă de transport, stabilirea disponibilităților servește la selecționarea navelor ce vor fi luate în calcul pentru alegerea navei ce urmează să efectueze transportul cu cheltuieli minime.

#### ● Alegerea navei ce urmează să efectueze un anumit transport

Este o activitate cu pondere foarte importantă în stabilirea unei bune politici economice a întreprinderii, activitate în care se interșes informații din toate celelalte activități operative precum și date specifice ei.

Prin natura activității sale specialistul este obligat să cunoască informații cu privire la caracteristicile tehnice ale navelor, caracteristicile fizico-chimice ale mărfurilor, condițiile de încărcare/descărcare din porturile ce urmează a fi atinse, cheltuielile portuare de operare, duratele de staționare, consumul de combustibil al navelor și prețul acestuia, cheltuielile fixe implicate de salariile echipajului și amortizarea navei, taxele de trecere prin canale și strîmtori, distanțele dintre porturile pe care urmează să le atingă, eventualele evenimente socio-politice care pot avea implicații în efectuarea transportului respectiv, datele de reparații planificate ale ravelor, curcașterea legislației internaționale etc.

Toate aceste informații împreună cu altele de o importanță mai scăzută concură la luarea unei decizii corecte pentru angajarea unei nave. Obținerea și prelucrarea lor în timp util necesită din partea specialistului însărcinat cu o astfel de activitate un volum de muncă ridicat și o responsabilitate deosebită. Cu toate că existența unor factori necuantificabili nu permite alegerea navei în mod automat, ca urmare a unui proces algoritmic programabil, totuși oferirea unei liste de priorități și asistarea specialistului pe parcursul luării deciziei prin oferirea de informații și prognoze cât și prin efectuarea unor calcule de rutină este posibilă și chiar de dorit, putînd duce în mod cert la îmbunătățirea acestei activități atât de complexe și cu implicații atât de mari în economia întreprinderii.

● **Planificarea transportului mărfurilor de masă**

Spre deosebire de alegerea unei nave care urmează să transporte o anumită partidă de mărfuri pentru mărfurile de masă cum sînt minereurile, produsele petroliere, cărbunile etc. se pune problema efectuării unei planificări pe o perioadă relativ lungă (de ordinul lunilor) a transportării cu un număr de nave a unei anumite cantități de marfă livrabile într-un anumit ritm. O astfel de activitate implică pe lîngă cunoașterea elementelor enumerate anterior și o muncă dificilă de întocmire a unor grafice de transport detaliate și care trebuiesc adaptate pe parcursul derulării lor, funcție de apariția de noi evenimente/cerințe.

Importanța pentru economia națională a bunei derulări a unor astfel de transporturi implică o acuratețe deosebită a rezultatelor cît și rezolvarea operativă, a oricărei probleme posibilă să apară fără impietarea desfășurării activității de transport.

## **2. PROIECTAREA SCHEMEI CONCEPTUALE A BAZEI DE DATE**

### **De ce bază de date ?**

Odată stabilite cerințele, preocuparea echipei de proiectare s-a canalizat în direcția determinării cărora dintre acestea li se poate găsi o rezolvare prin intermediul prelucrării automate a datelor, cît și asupra modului în care trebuie transpusă într-un limbaj informatic realitatea pe care o reprezintă activitatea de urmărire operativă a activității flotei maritime.

Datorită caracterului complex a acestei activități, a datelor de o mare diversitate pe care le manipulează, a conexiunilor multiple ce apar între aceste date, cît și datorită caracterului interactiv pe care îl are activitatea, rezolvarea printr-o aplicație construită pe bază de fișiere clasice a părut echipei de proiectare neadecvată.

Printre motivele cele mai importante care au decis abordarea aplicației cu o bază de date s-au aflat :

— prin natura lor diversele activități vehiculează date comune a căror redundanță necontrolată ar putea genera inconsistențe grave ale datelor cu urmări greu de prevăzut în viața aplicației ;

— datorită caracterului interactiv a acestor activități proiectarea cu ajutorul limbajelor clasice (COBOL, FORTRAN) este mai dificil de realizat chiar și pentru minicalculatoare, nemaivorbind de faptul că pentru calculatorul FELIX C-256/1024 gradul de dificultate crește ;

— stabilirea modelului datelor a tuturor conexiunilor necesare, a nivelelor de subordonare a diverselor tipuri de date este mult facilitată de caracteristicile limbajelor de descriere a datelor (LDD) specifice sistemelor de gestiune a bazelor de date ;

— caracterul interactiv al limbajului de manipulare a datelor (LMD) permite o mai bună adaptare a aplicației la modul conversațional în care trebuie să se desfășoare activitatea de urmărire a transportului maritim.

— independența relativă a datelor față de programe, caracteristică aplicațiilor cu baze de date, permite o mai mare flexibilitate în modificarea algoritmilor.

— caracterul neprocedural a anumitor instrucțiuni ale limbajului de manipulare a datelor permite obținerea de informații pentru care nu există programe de prelucrare prevăzute anterior.

## De ce SOCRATE ?

Alegerea SGBD-SOCRATE ca SGBD care să gestioneze baza de date a avut ca fundament următoarele considerente :

— SGBD SOCRATE este singurul SGBD operațional pentru gama de calculatoare FELIX-C-256/1024 ;

— față de SGBD-urile de tip CODASYL de pe minicalculatoare (ARGUS, LEDA) SOCRATE-MINI are următoarele argumente : viteză de prelucrare substanțial mai mare, tipuri de legături mai complexe, interfețe diversificate inclusiv de regăsire a datelor în baze de date distribuite într-o rețea, siguranță în folosire datorită numărului de implementări mult mai mare decât a celor de tip CODASYL ;

— inexistența unui SGBD relațional care să fie operațional și să satisfacă cerințele de viteză cerute de specificul aplicației ;

— caracterul interactiv specific limbajului SOCRATE mult mai bine adaptat caracteristicilor aplicației avute în vedere.

Chiar dacă toate aceste cerințe pot fi rezolvate într-un mod sau altul și cu ajutorul altor SGBD-uri și chiar cu ajutorul unor fișiere clasice, considerăm totuși că rezolvarea corectă a problemei impune o aplicație proiectată cu SGBD-SOCRATE.

Este drept că o astfel de bază de date necesită din partea utilizatorului specializarea unei echipe „administrator al bazei de date”, specializare ce necesită cunoștințe deosebite despre SGBD și aplicație, dar făcând raportul cerințe/performance rămânem la părerea că o astfel de soluție este de dorit pentru aplicații de un asemenea tip.

Odată definite cerințele întreprinderii și stabilit SGBD-ul care va gestiona baza de date al viitoarei aplicații s-a trecut la proiectarea schemei conceptuale a bazei de date.

În viața unei baze de date această etapă are importanță primordială, multe din calitățile (defectele) viitoarei aplicații fiind tributare ei.

Ținem să subliniem acest lucru deoarece, pentru aplicațiile avînd la bază organizarea clasică a datelor în fișiere, schema conceptuală cu toate că are o importanță mare nu ocupă totuși locul pe care-l are în cadrul proiectării unei baze de date.

În urma analizării cerințelor echipa de proiectare a stabilit Schema conceptuală de principiu a bazei de date (fig. VIII. 2.1).

Studiind această schemă se poate observa că baza de date are în componența sa următoarele entități :

- **NAVA** — este o entitate de nivel 0 cu 500 de realizări

Principalele caracteristici ale acestei entități sînt :

- cod navă-cheie unică ;
- denumire navă-cheie unică ;
- un grup de caracteristici tehnice ale navei : capacități de transport, pescaj, capacități tanc combustibil, consumuri în staționare, număr de hambare/tancuri etc. ;

— un grup de caracteristici economice : cheltuieli fixe pe zi, valoare fonduri fixe, indemnizație hrană, indemnizație străinătate, timp angajare navă etc. ;

-- un grup de caracteristici cu informații operative privitoare la navă : poziția navei (latitudine, longitudine), porturi de destinație, distanța pînă la destinație etc. ;

Subordonate entității NAVA se află entitățile :

● **CONSUM-COMBUSTIBIL** — avînd 10 realizări, 5 pentru mers în gol și 5 pentru mers încărcat cu caracteristicile : viteză, consum. Cu ajutorul celor 5 perechi viteză-consum se determină prin interpolare consumul de combustibil al navei pentru o anumită viteză.

● **TRASEUL NAVEI** — cu 30 realizări.

Această entitate evidențiază traseul ce-l are de parcurs nava conform contractelor încheiate. Ea conține codurile porturilor ce urmează a fi atinse sau care au fost deja atinse de navă, data intrării în aceste porturi, distanța dintre ele precum și o entitate subordonată ei.

● **MARFA-OPERATĂ** — cu 20 realizări ce ne furnizează date privitoare la mărfurile ce urmează sau au fost operate în aceste porturi.

● **PORT** — este o entitate de nivel 0 avînd 5 000 de realizări și conținînd realizări privitoare la toate porturile, țările și zonele geografice în care pot opera navele flotei maritime. Caracteristicile acestei entități se împart în :

- |  |  |
|--|--|
| -- caracteristici generale ca :                  | — caracteristici care permit prognozarea duratei de staționare în portul respectiv : |
| — cod-port-cheie unică ;                         | — zile-așteptare-radă ;  |
| — denumire-port-cheie unică ;                    | — zile-operare-petrolier ;   |
| — denumire-țară-cheie dublă ;                    | — zile-operare-mineralier etc.   |
| -- denumire-zonă-cheie dublă ;                   |  |
| — tip-port (port, canal, strîmtoare, țară, zonă) |  |
| — poziție geografică.                            |  |

Pentru prognozarea duratelor de staționare se folosesc datele culese de la ultimele 30 de nave de același tip care au trecut prin portul respectiv. Pentru cargouri datorită diferențelor mari care pot apare în operarea lor funcție de natura mărfii s-a ținut seama și de aceasta împărțind mărfurile în trei categorii : ușor de operat, mediu și greu de operat și făcîndu-se prelucrări statistice diferențiate pentru ele. Dacă pentru un port nu deținem date deoarece prin el nu a trecut încă nici o navă de tipul respectiv se preiau din realizarea corespunzătoare țării căruia îi aparține portul cele mai mari valori întîlnite în unul din porturile țării respective. Analog dacă nici în țară nu există un caz precedent se trece la zona geografică de care aparține portul și dacă nici în acel caz nu avem date se preia din realizarea cu denumire LUME date acoperitoare pentru prognoza respectivă.

Entitatea PORT are subordonată entitatea

● **CHELT-PORT-ÎNC-DESC** — cu 25 de realizări.

Cu ajutorul ei urmînd o cale similară celei prezentate la prognoza duratei de staționare se antecalculează costul escalei într-un port ce urmează a fi atins pe parcursul unui voiaj.

● **Act-operativă** — este o entitate de nivel 0 cu 50 000 de realizări, fiecare dintre ele reprezentînd activitatea zilnică a unei nave. Datele privitoare la acestea se rețin pe 6 luni în urmă și oricum pînă la terminarea unui voiaj, chiar dacă acesta depășește cele 6 luni amintite anterior.

Principalele caracteristici ale sale sînt :

- data activității ;
- cod-navă ;
- consum-combustibil ;

Ea are în subordine entitatea

● **SUBACTIVITATE** cu 6 realizări, deoarece pe parcursul unei zile o navă poate atinge mai multe porturi sau poate desfășura diverse activități : marș, operare, așteptare radă etc., fiecare dintre acestea corespunzînd unei realizări distincte a entității SUBACTIVITATE.

Caracteristicile principale ale unei realizări a entității SUBACTIVITATE sînt :

- cod-port ;
- cod-subactivitate ;
- durată-minute etc.

Ea are în subordine entitatea

- **ÎNTRERUPERI-ACTIVITATE** cu caracteristicile :
  - cod-întrerupere ;
  - durată-minute ;

care permit evidențierea diverselor întreruperi apărute pe parcursul unei activități.

● **Act-portuară – entitate de nivel 0 cu 15 000 realizări** în care se evidențiază activitatea flotei pe parcursul operării navelor într-un anumit port. Datele privitoare la această activitate se rețin pe 12 luni și privesc marfa operată, date importante privitoare la operare, durata operării etc.

Ea are subordonată entitatea

● **ÎNTRERUPERE** – cu 50 realizări ce evidențiază în mod detaliat toate întreruperile de pe parcursul operării unei nave într-un port, cauza și durata lor.

În afara acestor entități importante baze de data mai conține și entitățile nomenclator :

- |                                 |               |
|---------------------------------|---------------|
| ● SUBACT-INTR                   | ● L-NAVA      |
| ● GRUPA-MARFĂ                   | ● PORT-ESCALĂ |
| ● TIP-COMBUSTIBIL               | ● ANGAJĂRI    |
| ● ZILE-LUNI                     | ● DISPONIBIL  |
| precum și entitățile de lucru : | ● REALIZĂRI-1 |
| ● DATE-PAR                      | ● REALIZĂRI-2 |
| ● CAPACITĂȚI-DISPONIBILE        | ● PORT-NAVE   |
| ● PARAM                         | ● NAVE        |
| ● L-TRASEUL-NAVEI               |               |
| ● L-MARFA-OPERATĂ               |               |

Între entitățile NAVĂ respectiv ACT-OPERATIVĂ și ACT-PORTUARĂ sînt stabilite legături de tip ANNEAU-REFERE la fel ca și între entitățile PORT respectiv ACT-OPERATIVĂ și ACT-PORTUARĂ.

Prin aceste legături se atașează unei nave respectiv unui port toate activitățile operative (portuare) ale acesteia (respectiv desfășurate în acel port).

Ordinea din aceste inele este relevantă.

Entitățile ACT-OPERATIVĂ i se atașează o legătură de tip INVERSE prin care sînt grupate toate activitățile operative din ultima zi. De altfel în continuare va fi prezentată schema conceptuală în detaliu (fig. VIII. 2.2), pentru aceia care doresc o documentare mai profundă privitoare la baza de date.



Odată stabilită schema conceptuală în detaliu, s-a trecut la studierea ocupării subpaginilor din spațiu virtual și s-au aranjat de așa manieră caracteristicile componente ale bazei pentru a duce la o ocupare cât mai bună a acestora.

Caracteristicile cu frecvență mai mare au fost grupate față de cele care au o frecvență de umplere mai scăzută, pentru a nu ocupa inutil subpagini din spațiul real.

Procesul de stabilire în detaliu al structurii bazei de date, fiind un proces iterativ, studierea modului de ocupare a spațiului virtual și real se face prin rulări succesive cu ajutorul switch-ului /MAP de la procesorul DFS pentru minicalculatoare sau a ordinelor % ACTI TB:44 și % ACTI TB:64 pentru calculatorul FELIX C-256/1024.

### 3. ELABORAREA PROGRAMELOR ȘI A PROCEDURILOR AUTOMATE

Odată stabilită schema conceptuală a bazei de date s-a trecut la elaborarea procedurilor automate, proiectându-se două astfel de proceduri. Procedura de actualizare care va avea ca sarcină punerea la zi a bazei de date și care va fi apelată de personalul oficiului de calcul și procedura de interogare, prin care personalul compartimentelor funcționale va avea posibilitatea obținerii de informații necesare bunei desfășurări a procesului de producție.

*Procedura de actualizare este compusă din următoarele programe :*

- **PACTO** — prin care se face actualizarea bazei de date cu informații operative (poziția navei, mile parcurse, destinația, evenimente deosebite, consum combustibili etc.) furnizate zilnic prin radiograme de către navele flotei. Programul efectuează actualizarea în mod conversațional cu verificarea corelării datelor avînd prevăzut un mecanism de reluare în caz de incident, mecanism bazat pe fișierul de securitate de tip „Q” specific SOCRATE-MINI. Acest program va fi rulat în tura de noapte după ora 24,00 de către operatorul de servicii, creînd astfel posibilitatea atît compartimentelor funcționale cît și conducerii de a avea la dispoziție datele necesare încă de la prima oră a zilei.

- **MODACTO** — program care permite modificarea anumitor date operative introduse anterior prin intermediul programului PACTO și care, datorită unor cauze diverse (erori de transmisie prin radio, erori umane etc.) au fost incorecte. De asemenea, prin intermediul acestui program, se șterg datele care nu mai trebuiesc reținute deoarece și-au pierdut caracterul de actualitate. Actualizarea cu informații operative a fost ruptă în două programe, pe de o parte adăugarea de noi informații conform radiogramelor și pe de alta modificarea și ștergerea acestora funcție de diversele corecții furnizate de compartimentele funcționale, din cauza momentelor diferite la care sînt disponibile datele necesare celor două actualizări, lucru datorat surselelor diferite din care provin ele. Acest mod de abordare a problemei permite pe de o parte un control mai riguros al adăugării (se cer informații pentru fiecare navă rînd pe rînd, nepermițîndu-se astfel omiterea vreuneia) iar pe de altă parte, actualizarea imediat după ce datele primare sînt la dispoziția operatorului.

- **ACTTRASN** — prin intermediul căruia se actualizează baza de date cu toate informațiile privitoare la contractele de transport ce se derulează în acest moment sau care s-au perfectat și urmează să fie abordate ulterior. Aceste informații privesc porturile ce urmează să fie atinse, distanțele ce trebuiesc parcurse, mărfurile ce vor

fi operate, și provin de la compartimentele funcționale. Ele sînt validate și corelate cu informațiile operative furnizate de radiograme.

- **ACTNOM** — care permite punerea la zi a bazei de date cu informații de tip nomenclator necesare aplicației (cod întrerupere, denumire întrerupere, cod subactivitate, denumire subactivitate, cod combustibil, denumire combustibil, cod marfă, denumire marfă, tip operare marfă etc.).

- **ACTCHDG** — prin care se actualizează baza de date cu informații privitoare la : cursul leu/dolar, anumite valori ale cheltuielilor portuare și de treceri canale, precum și alte date cu caracter general care intervin în bunul mers al aplicației.

- **IFX1100** — prin intermediul căruia se face legătura cu aplicațiile financiar contabile care se exploatează pe sistemul FELIX C-256/1024. Cu ajutorul lui se prelucrează statistic informațiile existente în fișierele acestei aplicații obținîndu-se astfel datele necesare prognozelor privitoare la durata operării și la cheltuielile portuare, prognoze indispensabile în procesul de programare a transportului maritim de mărfuri.

Toate programele de actualizare sînt prevăzute cu mecanismul de reluare în caz de incident bazat pe lucru cu fișierul de securitate tip „Q” specific SOCRATE-MINI. Datorită acestui mecanism se permite reluarea actualizării cu pierderi minime de tip și de informație într-un mod automat care nu lasă operatorului posibilitatea săvîrșirii unor erori, inerente altfel. Pentru prevenirea urmărilor defectării suporturilor de informație, zilnic, se salvează printr-un program de aplicație SAVRESO informațiile actualizate în ultima zi, iar de două ori pe săptămîină se salvează cu utilitarul BRU toată baza de date.

În cazul apariției unor erori hardware pe disc, se formatează discul cu programele utilitare FMT și BAD, se restaurează baza de date cu programul utilitar BRU și eventual se reîncarcă informațiile utilitare cu programul de aplicație SAVRESO care aduce baza de date la zi.

Procedura de actualizare este de fapt un fișier de comenzi indirecte care începe cu programul utilitar LGI ce deschide sesiunea SOCRATE-MINI cu fișier de securitate și se termină automat cu programul utilitar LGO ce închide sesiunea de lucru SOCRATE-MINI. Fiind astfel protejată baza de date are un ridicat procent de fiabilitate lucru necesar în exploatarea unor aplicații de acest gen. Suplimentar pentru mărirea gradului de siguranță în exploatare s-a elaborat o baterie de programe care verifică coerența logică a bazei de date. Dintre testele care se fac amintim : verificarea bunei înlănțuiri a realizărilor din cadrul inelelor componente ale bazei de date, verificarea concordanței dintre numărul de realizări existente într-o entitate și valoarea contorului de prezență.

Procedura de interogare este o procedură de tip LOGIN.CMD existentă pentru toate UIC-urile utilizatorilor din compartimentele funcționale care se apelează automat atunci cînd prin intermediul comenzii HEL un anumit utilizator comunicînd numele și parola sa solicită accesul la minicalculator. Această procedură lansează programul SOCRATE-MINI DIALOG care comunică utilizatorului, folosind tehnica meniurilor, ca situații poate obține și așteaptă ca acesta să solicite una dintre ele. Cînd utilizatorul nu mai dorește alte situații procedura lansează automat comanda BYE care încheie sesiunea de lucru cu minicalculatorul. Prin acest mecanism s-a redus dialogul specializat operator-calculator, cunoștințele informatice necesare celui ce dorește interogarea bazei de date fiind foarte mici și ușor de învățat de către personalul compartimentelor funcționale.



Prin intermediul anumitor mecanisme de securitate, folosindu-se atît facilitățile limbajului SOCRATE-MINI cît și un mecanism propriu aplicației, se limitează accesul utilizatorilor la informațiile strict necesare lor și numai la acestea.

Situațiile oferite utilizatorilor sînt :

● **Activitatea operativă a flotei la data ...**

— Situație care conține date privitoare la navele în marș (port plecare, port destinație, distanța parcursă, distanța rămasă, consum combustibil, evenimente deosebite etc.), în rada unui port (data sosirii în radă, marfa ce urmează să fie încărcată, descărcată în acel port, cauza staționării etc.), în porturi (marfa de operat, cantitatea operată, rest de operat etc.), în reparații (șantierul unde se efectuează reparația, tipul reparației, durata planificată, cînd se va termina operația etc.), precum și date de sinteză a activității flotei maritime pe o anumită perioadă (distanța parcursă, marfa operată, consumul de combustibil, cauza și durata eventualelor evenimente deosebite apărute pe parcursul derulării activității).

Toate aceste informații pot fi obținute la cerere pentru o navă, un anumit tip de nave, un anumit grup de nave (mineraliere, petroliere, cargouri) sau pentru întreaga flotă.

Informațiile pot fi copiate pe listing cu ajutorul unor hard-copy atașate terminalelor, sau extrase direct pe fișiere de pe suporturi reutilizabile, urmînd apoi să fie listate asincron la imprimantă. Această din urmă soluție se aplică pentru cazurile în care volumul listelor este mare.

Cele prezentate mai sus sînt dealtfel valabile și pentru listele ce vor fi prezentate în continuare.

● **Situația operării navelor flotei în porturile românești**

Oferă informații privitoare la navele ce urmează să sosească, navele aflate în radă (de cînd se află în radă, cauza care nu le permite să înceapă operarea etc.), nave sub operațiuni (ce marfă operează, ce întrerupere a survenit în operația ei, ce cantitate s-a operat, cît a mai rămas de operat, dana la care se află nava etc.).

Aceste informații sînt mult mai detaliate decît cele din situația anterioară datorită faptului că pe lîngă datele din radiograme sînt folosite informațiile furnizate de rapoartele dispecerilor din porturile românești unde se află sub operațiuni aceste nave.

● **Activitatea flotei maritime în ... de la ... la ...**

Această situație furnizează pe baza informațiilor survenite din rapoartele comandanților navelor modul cum a decurs operarea într-un anumit port, în porturile dintr-o anumită țară sau zonă geografică într-un anumit interval de timp. Informațiile-

furnizate de ea nu mai au un caracter operativ, ele reprezintă o reflectare a actelor de constatare încheiate de comandant cu reprezentanții portului în care s-a desfășurat operarea. Pe baza analizei acestora, pentru o navă, un anumit tip de navă sau un anumit grup de nave se pot trage concluzii asupra condițiilor specifice de operare în porturile respective, asupra măsurilor necesare a fi adoptate pentru a înlătura eventualele fenomene negative ce se semnalează în activitatea flotei.

#### ● Proгноza duratelor de staționare în ...

Pe baza prelucrării datelor statistice privitoare la operarea navelor se preliminează un anumit nivel al cheltuielilor de staționare și de operare pentru un anumit port. Proгноza se face după cum am spus și mai înainte, ținând cont și de natura mărfurilor ce urmează a fi operate precum și de condițiile specifice portului unde urmează să fie încărcată/descărcată marfa.

Aceste date preliminare sînt strict necesare pentru analiza unei oferte de transport precum și pentru alte analize efectuate de compartimentele funcționale. Situația poate fi cerută pentru o navă, un anumit tip de nave, un anumit grup de nave, într-un port sau în porturile unei anumite țări sau zone geografice.

#### ● Proгноza cheltuielilor portuare în ...

Situația oferă informații preliminate privitoare la nivelul cheltuielilor portuare ce urmează a fi făcute pentru staționarea sau operarea unui anumit tip de navă, într-un anumit port, informații necesare antecalculării eficienței unei oferte de transport.

Datele privitoare la această situație cît și cele privitoare la ultimele două situații folosesc atît informațiile furnizate de actualizările operative cît și date de sinteză preluate cu ajutorul programului IFX1100 din aplicațiile financiar-contabile existente în exploatare pe calculatoarele FELIX C-256/1024.

#### ● Situația consumului de combustibil.

Aceasta oferă date privitoare la consumurile de combustibili în marș, în staționare sau sub operațiuni ale navelor flotei maritime, într-un anumit interval de timp.

Ea este utilă deoarece cheltuielile legate de consumul de combustibil au o pondere ridicată în totalul cheltuielilor efectuate pentru un anumit transport. Cunoașterea lor exactă în orice moment permite o mai bună gospodărire a resurselor financiare ale întreprinderii precum și luarea unor măsuri care să permită economisirea unor cantități apreciabile de combustibil.

- **Situația abaterilor de la consumul de combustibil stabilit**

Este o situație pe baza căreia se analizează navele care au avut consumuri peste cuantumul stabilit și cauzele care au generat aceste consumuri exagerate, permițând luarea unor decizii operative pentru înlăturarea cauzelor ce au generat astfel de cazuri.

Informațiile furnizate privesc: nava care a săvârșit abaterea, data abaterii, viteza cu care navighează, consumul suplimentar etc.

- **Disponibilitățile capacităților de transport la data...**

Pe baza datelor privitoare la contractele care se derulează, a celor preliminare privitoare la durata operării navelor în porturile ce vor fi atinse, precum și a informațiilor privitoare la durata reparațiilor se furnizează la cerere o situație care preli-minează ce nave vor avea capacități disponibile de transport, în ce porturi se vor afla la acea dată, când este planificată nava să intre în reparație etc. Situația poate fi obținută la cerere pentru un anumit tip de nave, un anumit grup de nave sau pentru toată flota.

Pe baza informațiilor furnizate se ia decizia pentru asigurarea de con-tracte de transport, permițând reducerea timpilor de imobilizare a capacitaților flotei maritime.

- **Planificarea transportului mărfurilor de masă**

În funcție de datele referitoare la cererile de transport furnizate de utilizator precum și pe baza analizei capacității de transport disponibile pentru tipurile de nave cerute, ținându-se cont de datele preliminare privitoare la durata operării navelor, duratele de marș, se stabilesc graficele de transport mărfuri de masă, urmărind reducerea consumului de combustibil, reducerea timpilor de imobilizare a capacitaților de transport, realizarea în timp util a transporturilor de mărfuri prioritare pentru economia națională.

## **4. REZOLVAREA PE BAZA DE DATE IMPLEMENTATĂ A ACTIVITĂȚII DE PROGRAMARE A TRANSPORTULUI DE MĂRFURI**

### **Programarea transportului de mărfuri și antecalculul ofertei**

Această activitate este una dintre cele mai complexe în cadrul procesului de conducere operativă a activității flotei maritime, cu mari implicații în buna desfășurare a ei, necesitând vehicularea unor volume de date mari, unele dintre ele, avînd un caracter necuantificabil. Aceasta a făcut imposibilă stabilirea unui algoritm de

determinare automată a navei ce urmează să efectueze transportul, aplicația limitându-se a asista specialistul în luarea deciziei.

*Această funcțiune a aplicației a fost realizată de 4 module care se execută în strînsă corelație.* Primul modul PREOF1 primește pe baza unui dialog om-mașină informațiile privitoare la oferta de transport :

- tipul navlului (pe tonă, pe mc etc.) ;
- valoarea navlului ;
- porturile ce urmează a fi atinse ;
- mărfurile ce urmează a fi încărcate/descărcate ;
- datele cînd aceste mărfuri sînt disponibile ;
- condiții speciale de operare ;
- norma de operare din porturile de pe traseu (dacă aceasta este prevăzută în contract) ;
- canalele sau strîmtorile ce urmează să fie atinse în cadrul voiajului.

Aceste date sînt validate și corelate cu informațiile existente în bază și dacă nu sînt detectate erori se trece la execuția modului PREOF2. El are ca obiectiv principal selecționarea navelor care prin caracteristicile tehnice se pretează la transportarea mărfurilor conținute de oferta de transport și care în același timp sînt disponibile în vecinătatea portului de încărcare în intervalul de timp precizat de ofertă. Pentru stabilirea acestor nave se folosesc mecanisme de prognoză a duratelor de operare și de marș prezentate anterior.

Pe parcursul selecționării navelor care îndeplinesc condițiile cerute se furnizează și o listă de nave care datorită fie lipsii unor informații elementare privitoare la contractele lor fie datorită necorelării dintre datele de contract și cele operative nu pot fi analizate și deci nu sînt luate în considerare la programarea transportului respectiv.

Odată stabilită lista navelor ce îndeplinesc toate condițiile necesare se lansează în lucru modulul PREOF3 care are ca principal obiectiv calculul cheltuielilor necesitate de respectivul voiaj, pentru fiecare din navele selectate anterior și furnizarea către utilizator a unei situații conținînd navele care pot efectua transportul cu toate datele privitoare la costul acestuia (cheltuieli fixe, cheltuieli de operare, cheltuieli pentru combustibili, cheltuieli privitoare la personal), în ordinea crescătoare a totalului acestor cheltuieli. Pe baza acestei liste, ținînd seama și de alte informații care nu au putut fi cuantificate, se determină care este nava ce urmează să efectueze voiajul. Odată determinată această navă, modulul PREOF4 emite ANTECALCULUL OFERTA în care se recapitulează detaliat și sub o formă prestabilită toate informațiile luate în considerare la angajarea acestui transport, oferindu-se în plus date privitoare la rentabilitatea lui. Pentru o mai bună familiarizare cu problema va fi prezentată în continuare sursa modului PREOF3 precum și un exemplar din lista emisă de acesta pentru o anumită ofertă de transport împreună cu antecalculul ofertă privitor la ea. De observat că datele conținute în situațiile ce vor fi prezentate au un caracter de exemplu, nereflectînd un caz real.

## Implementarea bazei de date

Odată încheiată faza de proiectare a bazei de date, echipa mixtă de proiectanți din cadrul Centrului de calcul și specialiști ai Întreprinderii de transport maritim, a trecut la implementarea aplicației.

Pentru a putea să se desfășoare corect această fază, în paralel cu proiectarea bazei de date, s-a efectuat instruirea personalului compartimentelor funcționale, precum și a personalului operativ din cadrul oficiului de calcul.

Pe măsura introducerii în exploatare, programele au fost verificate cum se comportă în condiții reale, s-au remediat erorile detectate și s-a obișnuit personalul cu noul flux informațional.

Datorită faptului că algoritmi folosiți în cadrul aplicației au în general un caracter euristic, s-a verificat modul în care răspunde unor situații reale și s-au îmbunătățit permanent pe parcursul implementării. Astfel într-un timp scurt s-a realizat pe de o parte verificarea cu date reale a programelor aplicației, organizarea fluxului informațional nou introdus cât și acomodarea personalului operativ cu un nou stil de muncă. Odată depășită această fază aplicația a intrat în exploatare, echipa „ADMINISTRATOR“ a bazei de date din partea întreprinderii, preluând toate sarcinile de supraveghere și control a bazei de date, urmînd ca pe măsura apariției unor evenimente deosebite care ar depăși competența acestora să se apeleze și la proiectanții aplicației, pentru remedierea unor astfel de incidente.

În general din experiența căpătată la proiectarea acestei aplicații, considerăm că în afara algoritmilor specifici proiectantul are de rezolvat două mari categorii de probleme și anume : punerea la punct cu tehnicile de realizare specifice programelor interactive, precum și punerea la punct cu tehnicile necesare la realizarea mecanismelor de reluare în caz de incident.

Pentru ca lucrurile să se poată desfășura normal este necesar ca echipa de proiectare să aibă o concepție unitară asupra acestora și în general să poată cunoaște pe viu modul în care au fost rezolvate ele în alte aplicații similare. Trebuie subliniată din nou ponderea mare pe care o are proiectarea schemei conceptuale în performanțele/deficiențele viitoarei aplicații.

Credem că niciodată nu poate fi considerat timp pierdut timpul alocat studierii structurii bazei de date.

Pentru programatorii cu mai puțină experiență în SOCRATE, atragem atenția că acest limbaj prin gama largă de posibilități pe care o oferă poate deruta și de aceea alegerea funcțiunii corecte într-un anumit context trebuie făcută cu atenție.

Acestea sînt principalele observații făcute de noi pe baza experienței căpătate la proiectarea aplicației descrise cu speranța că vor ajuta în munca lor pe cei ce vor aborda în viitor aplicații similare.

# 5. LISTINGURI COMPLETE PENTRU SCHEMA CONCEPTUALĂ, MANIPULAREA DATELOR, REZOLVAREA STUDIULUI DE CAZ

## Schema conceptuală

OCDEF V3.1 27-AUG-87 11:56 PAGINA .

1	DEBUT	
2	ENTITE 500 NAVA	ENTITATE FR
3	DEBUT	
4	INEL-ACT-OPERATIVA ANNEAU AVEC CHAINE DOUBLE FIN	
5	INEL-ACT-PORTUARA ANNEAU AVEC CHAINE DOUBLE FIN	
6	DATA-INTRARII-IN-EXPLOATARE	BLOC
7	DEBUT	
8	ZI DE 1 A 31	
9	LUNA DE 1 A 12	
10	AN DE 1940 A 2100	
11	FIN	SFIRSIT BLOC
12	DATA-EXPIRARE-ACTE-REPARATIE	BLOC
13	DEBUT	
14	ZI DE 1 A 31	
15	LUNA DE 1 A 12	
16	AN DE 1980 A 2100	
17	FIN	SFIRSIT BLOC
18	DATA-INCEPERE-REPARATIE-PLAN	BLOC
19	DEBUT	
20	ZI DE 1 A 31	
21	LUNA DE 1 A 12	
22	AN DE 1980 A 2100	
23	FIN	SFIRSIT BLOC
24	COD-NAVA MOT 7 AVEC CLE UNIQUE FIN	
25	DENUMIRE-NAVA MOT 20 AVEC CLE UNIQUE FIN	
26	TIP-COMBUSTIBIL-MARS-MOT-PR MOT 2	
27	TIP-COMBUSTIBIL-MARS-MOT-AUX MOT 2	
28	TIP-COMBUSTIBIL-1-STATIONARE MOT 2	
29	TIP-COMBUSTIBIL-2-STATIONARE MOT 2	
30	COD-PORT-ACTUAL MOT 8	
31	COD-PORT-ULTIM-PLECARE MOT 8	
32	COD-PORT-DESTINATIE MOT 8	
33	TIP-ANGAJARE-NAVA MOT 1	
34	GRUP-NAVE (4 10) (PETROLIER MINERALIER CARGOU ANIMALIER)	
35	TIP-REPARATIE (6 3) (RCP RGR RT RC1 RC2 RK)	
36	EMISFERA-LATITUDINII (2 1) (N S)	
37	EMISFERA-LONGITUDINII (2 1) (E W)	
38	SECTIA DE 1 A 9	
39	CAPACITATE-TONE-CARGO-VARA DE 1000 A 250000	
40	CAPACITATE-TONE-CARGO-IARNA DE 1000 A 250000	
41	CAPACITATE-TDW DE 1000 A 500000	
42	CAPACITATE-TRN DE 300 A 500000	
43	CAPACITATE-TRB DE 1000 A 500000	
44	CAPACITATE-MC DE 500 A 500000	
45	VITEZA-IMPUSA DE 100 A 800	
46	L-1 DE 0 A 250000	
47	PESCAJ DE 1 A 10000	
48	NR-ULTIMA-ACT-OPERATIVA DE 1 A 50000	
49	CAPACITATE-TANC-COMBUST-TO DE 1 A 99999	
50	DURATA-PLAN-REPARATIE DE 1 A 999	
51	DURATA-PLAN-PREG-SANTIER DE 1 A 999	
52	CHELTUIELI-FIXE-PE-ZI DE 1000 A 100000	
53	AFUNDAREA-IN-TO-PE-INCH DE 1 A 1000	
54	VALOARE-FONDURI-FIXE-IN-MII DE 500 A 900000	
55	INDEMNIZATIA-HRANA-DOLARI DE 1 A 9000	
56	INDEMNIZATIA-STRAINATATE-DOL DE 1 A 9000	
57	CANTIT-CONSUM-1-ZI-STAT-KG DE 100 A 300000	
58	CANTIT-CONSUM-2-ZI-STAT-KG DE 100 A 300000	
59	CANTITATE-CONSUM-MARS-AUX DE 100 A 300000	
60	NUMAR-VOIAJ DE 1 A 999	
61	NUMAR-HAMBARE DE 1 A 30	

62 TIP-ACT-NAVA DE 0 A 9  
 63 LATITUDINEA-GRADE DE 0 A 90  
 64 LATITUDINEA-MINUTE DE 0 A 59  
 65 LONGITUDINE-GRADE DE 0 A 180  
 66 LONGITUDINE-MINUTE DE 0 A 59  
 67 DISTANTA-PINA-LA-DESTINATIE DE 0 A 30000  
 68 ENTITE 10 CONSUM-COMBUSTIBIL  
 69 DEBUT ENTITATE 1  
 70 VITEZA DE 0 A 800  
 71 CONSUM DE 100 A 300000  
 72 FIN  
 73 ENTITE 30 TRASEUL-NAVEI SFIRSIT 1  
 74 DEBUT ENTITATE 2  
 75 COD-PORT MOT 8  
 76 DATA-INTRARE-IN-PORT DE 19790101 A 21001231  
 77 DISTANTA DE 10 A 30000  
 78 PROCENT-REDUCERE-VITEZA DE 0 A 99  
 79 L-1 DE 0 A 999  
 80 ENTITE 20 MARFA-OPERATA ENTITATE 2.1  
 81 DEBUT  
 82 COD-MARFA MOT 3  
 83 TIP-OPERARE (2 4) (INC DESC)  
 84 CANTITATE-TONE DE 1 A 250000  
 85 L-1 DE 0 A 999  
 86 FIN  
 87 FIN SFIRSIT 2.1  
 88 FIN SFIRSIT 2  
 89 ENTITE 5000 PORT SFIRSIT PR  
 90 DEBUT  
 91 INEL-ACT-PORTUARA ANNEAU AVEC CHAINE DOUBLE FIN  
 92 INEL-ACT-OPERATIVA ANNEAU AVEC CHAINE DOUBLE FIN  
 93 COD-PORT MOT 8 AVEC CLE UNIQUE FIN  
 94 DENUMIRE-PORT MOT 25 AVEC CLE UNIQUE FIN  
 95 DENUMIRE-TARA MOT 20 AVEC CLE FIN  
 96 DENUMIRE-ZONA MOT 18 AVEC CLE FIN  
 97 TIP-PORT (6 10) (PORT CANAL STRIMTOARE TRAVERS TARA ZONA)  
 98 EMISFERA-LATITUDINII (2 1) (N S)  
 99 EMISFERA-LONGITUDINII (2 1) (E W)  
 100 LATITUDINEA-GRADE DE 0 A 90  
 101 LATITUDINEA-MINUTE DE 0 A 60  
 102 LONGITUDINEA-GRADE DE 0 A 180  
 103 LONGITUDINEA-MINUTE DE 0 A 60  
 104 NR-NAVE-ZILE-ASTEPT-RADA-PETR DE 0 A 30  
 105 NR-NAVE-ZILE-ASTEPT-RADA-MIN DE 0 A 30  
 106 NR-NAVE-ZILE-ASTEPT-RADA-CARG DE 0 A 30  
 107 ZILE-ASTEPT-RADA-PETR DE 0 A 4500  
 108 ZILE-ASTEPT-RADA-MIN DE 0 A 4500  
 109 ZILE-ASTEPT-RADA-CARG DE 0 A 4500  
 110 VECINATATE-NORD-GRADE DE 0 A 60  
 111 TONE-ZILE-OPERARE-PETR DE 0 A 5000000  
 112 VECINATATE-NORD-MINUTE DE 0 A 59  
 113 TONE-ZILE-OPERARE-MIN DE 0 A 3000000  
 114 VECINATATE-SUD-GRADE DE 0 A 60  
 115 TONE-ZILE-OPERARE-CARG-MFU DE 0 A 500000  
 116 VECINATATE-SUD-MINUTE DE 0 A 59  
 117 VECINATATE-EST-GRADE DE 0 A 120  
 118 TONE-ZILE-OPERARE-CARG-MFM DE 0 A 500000  
 119 VECINATATE-EST-MINUTE DE 0 A 59  
 120 VECINATATE-VEST-GRADE DE 0 A 120  
 121 TONE-ZILE-OPERARE-CARG-MFG DE 0 A 500000  
 122 ZILE-OPERARE-PETR DE 0 A 4500

123 ZILE-OPERARE-MIN DE 0 A 4500  
 124 ZILE-OPERARE-CARG-MFU DE 0 A 4500  
 125 VECINATATE-VEST-MINUTE DE 0 A 59  
 126 ZILE-OPERARE-CARG-MFM DE 0 A 4500  
 127 ZILE-OPERARE-CARG-MFG DE 0 A 4500  
 128 ENTITE 25 CHELT-PORT-INC-DESC  
 129 DEBUT  
 130 TIP-NAVA MOT 1  
 131 AN-OPERARE-PORT DE 1978 A 2100  
 132 LUNA-OPERARE-PORT DE 1 A 12  
 133 TDW DE 0 A 250000  
 134 TRN DE 0 A 250000  
 135 TRB DE 0 A 250000  
 136 CHELTUIELI-PORTUARE DE 0 A 10000000  
 137 CHELTUIELI-INC-DESC DE 0 A 10000000  
 138 FIN  
 139 FIN SFIRSIT PR  
 140 ENTITE 50000 ACT-OPERATIVA  
 141 DEBUT  
 142 REFERINTA-NAVA REFERE INEL-ACT-OPERATIVA DE UNE NAVA  
 143 AVEC CHAINE DOUBLE FIN  
 144 REFERINTA-PORT REFERE INEL-ACT-OPERATIVA DE UN PORT  
 145 AVEC CHAINE DOUBLE FIN  
 146 DATA-ACTIVITATII BLOC  
 147 DEBUT  
 148 ZI DE 1 A 31  
 149 LUNA DE 1 A 12  
 150 AN DE 1980 A 2100  
 151 FIN SFIRSIT BL  
 152 COD-NAVA MOT 7  
 153 CONSUM-COMB-GREU DE 0 A 300000  
 154 CONSUM-COMB-USOR DE 0 A 300000  
 155 NUMAR-CALATORIE DE 1 A 999  
 156 ENTITE 6 SUBACTIVITATE  
 157 DEBUT  
 158 COD-PORT MOT 3  
 159 COD-SUBACTIVITATE MOT 5  
 160 MOD-PARCURS-MILE (2 1) (P G)  
 161 DURATA-MINUTE DE 1 A 1440  
 162 DANA MOT 4  
 163 MILE-PARCURSE DE 0 A 1000  
 164 CANTITATE-MARFA-OPERATA DE 0 A 250000  
 165 L-3 DE 0 A 16000  
 166 ENTITE 3 INTRERUPERI-ACTIVITATE  
 167 DEBUT  
 168 COD-INTRERUPERE MOT 5  
 169 DURATA-MINUTE DE 1 A 1440  
 170 FIN SFIRSIT 1.1  
 171 FIN SFIRSIT 1.  
 172 FIN SFIRSIT FF.  
 173 ULTIMA-ZI INVERSE TOUT ACT-OPERATIVA  
 174 ENTITE 500 SUBACT-INTR ENTITATE FF  
 175 DEBUT  
 176 COD-SUBACTIVITATE MOT 5 AVEC CLE UNIQUE FIN  
 177 DENUMIRE-SUBACTIVITATE MOT 28  
 178 TIP-SUBACTIVITATE (7 4) (A-MA A-PO A-MX A-RE I-IN I-EX I-MA)  
 179 FIN SFIRSIT FF  
 180 ENTITE 100 GRUFA-MARFA ENTITATE FF  
 181 DEBUT  
 182 COD-GRUFA-MARFA MOT 3 AVEC CLE UNIQUE FIN  
 183 DENUMIRE-MARFA MOT 25



SQCDEF V3.1 27-AUG-87 11:56 PAGINA 4

184 TIP-MARFA (3 5) (USOR MEDIU GREU)  
185 FIN 'SFIRSIT PR  
186 ENTITE 100 TIP-COMBUSTIBIL ENTITATE PR  
187 DEBUT  
188 COD-COMBUSTIBIL MOT 2 AVEC CLE UNIQUE FIN  
189 DENUMIRE-COMBUST MOT 22  
190 PRET-PER-TONA DE 0 A 40000  
191 PRET-PER-TONA-DOLARI DE 0 A 10000  
192 FIN 'SFIRSIT PR  
193 ENTITE 15000 ACT-PORTUARA ENTITATE PR  
194 DEBUT  
195 REFERINTA-NAVA REFERE INEL-ACT-PORTUARA DE UNE NAVA  
196 AVEC CHAINE DOUBLE FIN  
197 REFERINTA-PORT REFERE INEL-ACT-PORTUARA DE UN PORT  
198 AVEC CHAINE DOUBLE FIN  
199 DATA-SOSIT-RADA  
200 DEBUT  
201 ZI DE 1 A 31  
202 LUNA DE 1 A 12  
203 AN DE 1978 A 2100  
204 ORA DE 0 A 24  
205 MINUT DE 0 A 59  
206 FIN  
207 DATA-ACOSTAT-DANA  
208 DEBUT  
209 ZI DE 1 A 31  
210 LUNA DE 1 A 12  
211 AN DE 1978 A 2100  
212 ORA DE 0 A 24  
213 MINUT DE 0 A 59  
214 FIN  
215 DATA-DEPUS-NOTICE  
216 DEBUT  
217 ZI DE 1 A 31  
218 LUNA DE 1 A 12  
219 AN DE 1978 A 2100  
220 ORA DE 0 A 24  
221 MINUT DE 0 A 59  
222 FIN  
223 DATA-ACCEPTAT-NOTICE  
224 DEBUT  
225 ZI DE 1 A 31  
226 LUNA DE 1 A 12  
227 AN DE 1978 A 2100  
228 ORA DE 0 A 24  
229 MINUT DE 0 A 59  
230 FIN  
231 DATA-INCEPUT-OPERARE  
232 DEBUT  
233 ZI DE 1 A 31  
234 LUNA DE 1 A 12  
235 AN DE 1978 A 2100  
236 ORA DE 0 A 24  
237 MINUT DE 0 A 59  
238 FIN  
239 DATA-TERMINAT-OPERARE  
240 DEBUT  
241 ZI DE 1 A 31  
242 LUNA DE 1 A 12  
243 AN DE 1978 A 2100  
244 ORA DE 0 A 24

SOCDEF V3.1' 27-AUG-87 11:56 PAGINA 5

245	MINUT DE 0 A 59	
246	FIN	
247	DATA-PLECAT-DIN-PORT	
248	DEBUT	
249	ZI DE 1 A 31	
250	LUNA DE 1 A 12	
251	AN DE 1978 A 2100	
252	ORA DE 0 A 24	
253	MINUT DE 0 A 59	
254	FIN	
255	COD-PORT MOT 8	
256	COD-NAVA MOT 7	
257	TIP-OPERARE (4 10) (INCARCARE DESCARCARE INC-DESC TRANZIT)	
258	COD-MARFA MOT 3	
259	NUMAR-CALATORIE DE 0 A 999	
260	CANTITATE-MARFA DE 0 A 250000	
261	DURATA-STATIONARE DE 1 A 9999	
262	DURATA-OPERARE DE 0 A 9999	
263	L-1 DE 0 A 250000	
264	ENTITE 50 INTRERUPERE	ENTITATE 1
265	DEBUT	
266	COD-INTRERUPERE MOT 5	
267	DURATA DE 1 A 500000	
268	FIN	SFIRSIT 1
269	FIN	SFIRSIT FR
270	ENTITE 300 PORT-NAVE	
271	DEBUT	
272	COD-PORT MOT 8 AVEC CLE UNIQUE FIN	
273	ENTITE 200 NAVE	
274	DEBUT	
275	NR-REALIZ DE 1 A 500	
276	IDENTIF DE 0 A 3	
277	FIN	
278	FIN	
279	ENTITE 100 CAPACITATI-DISPONIBILE	
280	DEBUT	
281	GRUP-NAVE MOT 8 AVEC CLE UNIQUE FIN	
282	ANUL DE 1980 A 2100	
283	CAPACITATE-CARGO-VARA DE 1000 A 250000	
284	CAPACITATE-CARGO-IARNA DE 1000 A 250000	
285	CAPACITATE-MC DE 500 A 350000	
286	ENTITE 366 ZILE	
287	DEBUT	
288	DISPONIBILE DE 0 A 500	
289	ANGAJATE DE 0 A 500	
290	FIN	
291	ENTITE 10 PESCAJ-AF-TO-INCH	
292	DEBUT	
293	CANTITATE-MARFA DE 0 A 300000	
294	PESCAJ DE 0 A 10000	
295	AFUNDARE DE 1 A 1000	
296	FIN	
297	FIN	
298	ENTITE 1 DATE-PAR	
299	DEBUT	
300	NAVA-TIP MOT 30	
301	DATA-CURENTA	
302	DEBUT	
303	ZI DE 1 A 31	
304	LUNA DE 1 A 12	
305	AN DE 1980 A 2100	

SOCDEF V3.1 27-AUG-87 11:56 PAGINA 6

306 FIN  
 307 MAI-DORITI-MODIFICARI (2 2) (DA NU)  
 308 MAI-DORITI-SI-ALTE-MODIFICARI (2 2) (DA NU)  
 309 DORITI-MODIFICARE (2 2) (DA NU)  
 310 DORITI-CORECTAREA-DATEI (2 2) (DA NU)  
 311 CE-ENTITATE-DORITI-SA-STERGETI (5 13)  
 312 (PORT NAVA ACT-OPERATIVA ACT-PORTUARA GATA)  
 313 FENTRU MOT 30  
 314 IN MOT 30  
 315 CE-SITUATIE-DORITI MOT 30  
 316 MAI-DORITI-SI-ALTE-SITUATII (2 2) (DA NU)  
 317 PORTUL MOT 30  
 318 NAVA MOT 30  
 319 DATA-TEST  
 320 DEBUT  
 321 ZI DE 1 A 31  
 322 LUNA DE 1 A 12  
 323 AN DE 1978 A 2100  
 324 FIN  
 325 CE-ACTUALIZARE-DORITI MOT 30  
 326 TIP-INTEROGARE DE 1 A 9  
 327 CONTINUAM (2 2) (DA NU)  
 328 DISTANTA-PINA-LA-DESTINATIE DE 0 A 29990  
 329 AVETI-INTRERUPERI (2 2) (DA NU)  
 330 NR-REALIZARI DE 0 A 30000  
 331 NR-REALIZARE DE 0 A 30000  
 332 NR-REALIZI DE 0 A 90000  
 333 DORITI (2 2) (DA NU)  
 334 ZI-INCEPUT DE 1 A 31  
 335 LUNA-INCEPUT DE 1 A 12  
 336 AN-INCEPUT DE 1980 A 2100  
 337 ZI-SFIRSIT DE 1 A 31  
 338 LUNA-SFIRSIT DE 1 A 12  
 339 COD-PORT-SUBACTIVITATE MOT 30  
 340 AN-SFIRSIT DE 1980 A 2100  
 341 MARFA MOT 25  
 342 CANTITATE-MARFA DE 1 A 200000  
 343 COD-PORT-RELUARE MOT 30  
 344 CE-TIP-DE-SITUATII-DORITI (2 8) (ANALITIC TOTAL)  
 345 DORITI-PE-O-PERIOADA (2 2) (DA NU)  
 346 CURS-LEU-DOLAR DE 0 A 1000000  
 347 COEFICIENT-CORECTIE-CURS DE 0 A 1000000  
 348 ZIUA-CURENTA BLOC  
 349 DEBUT  
 350 ZI DE 1 A 31  
 351 LUNA DE 1 A 12  
 352 AN DE 1980 A 2100  
 353 FIN SFIRSIT BL.  
 354 NUMAR-VOIAJ DE 1 A 999  
 355 DORITI-MODIFICARE-PORT (2 2) (DA NU)  
 356 NUMARUL-PORTULUI-DE-MODIFICAT DE 1 A 25  
 357 DORITI-ACTUALIZARE (2 2) (DA NU)  
 358 DORITI-ADAUGARE-MARFURI (2 2) (DA NU)  
 359 DORITI-ADAUGARE-PORT (2 2) (DA NU)  
 360 NUMARUL-PORTULUI-DE-ADAUGAT DE 1 A 25  
 361 DISTANTA-PINA-LA-EL DE 10 A 30000  
 362 SINT-MARFURI-DE-OPERAT (2 2) (DA NU)  
 363 MAI-SINT-MARFURI (2 2) (DA NU)  
 364 MAI-DORITI-SA-ADAUGATI-PORTURI (2 2) (DA NU)  
 365 DORITI-STERGEREA-UNUI-PORT (2 2) (DA NU)  
 366 NUMARUL-PORTULUI-DE-STERS DE 1 A 25

367 MAI-AVETI-DE-STERS-PORTURI (2 2) (DA NU)  
368 MAI-DORITI-SA-ACTUALIZATI (2 2) (DA NU)  
369 MAI-DORITI-MODIFICARE-PORT (2 2) (DA NU)  
370 LATITUDINE-VECINATATE-NORD DE 0 A 18000  
371 LATITUDINE-VECINATATE-SUD DE 0 A 18000  
372 LONGITUDINE-VECINATATE-EST DE 0 A 36000  
373 LONGITUDINE-VECINATATE-VEST DE 0 A 36000  
374 TIP-NAVLU MOT 30  
375 CINE-SUPPORTA-CHELTUIELILE MOT 30  
376 VALOARE-NAVLU-DOLARI DE 0 A 100000000  
377 VALOARE-NAVLU-CENTI DE 0 A 99  
378 CONDITII-INC-DESC MOT 30  
379 DEMURRAGE DE 0 A 1000000  
380 CINE-PLATESTE-DEMURRAGE MOT 30  
381 TIP-OPERARE MOT 30  
382 DATA-REPARATIE DE 19790101 A 21001231  
383 MAI-SINT-PORTURI (2 2) (DA NU)  
384 TIP-STAT-PORT DE 0 A 9  
385 CANTITATE-TOT-TO DE 0 A 100000000  
386 CANTITATE-TOT-MC DE 0 A 100000000  
387 DORITI-ELIMINAREA-UNEI-NAVE (2 2) (DA NU)  
388 NUMARUL-NAVEI-DE-ELIMINAT DE 1 A 100  
389 DORITI-FISA-DE-ANTECALCUL (2 2) (DA NU)  
390 TAXE-TRECERE DE 0 A 100000000  
391 DISTANTA DE 0 A 30000  
392 DURATA-ZILE-TRAVERSARE DE 0 A 99  
393 CU-ZONE-DE-VECINATATE (2 2) (DA NU)  
394 ANUL DE 1980 A 2100  
395 CAPACITATE-TONE DE 1000 A 250000  
396 CAPACITATE-MC DE 500 A 350000  
397 NUMARUL-NAVEI-ALESE DE 1 A 100  
398 CANTITATE-PROCURAT-DIN-TARA DE 0 A 300000  
399 DURATA-ZILE-GOL DE 0 A 999  
400 CANTITATE-TOTALA-MARFA-TO DE 0 A 1000000000  
401 CANTITATE-TOTALA-MARFA-MC DE 0 A 1000000000  
402 CANTITATE-TONE-LA-UN-TRANSPORT DE 0 A 300000  
403 DURATA-STAT-RADA DE 0 A 999  
404 DURATA-OPERARE DE 0 A 999  
405 PROCENT-REDUCERE-VOIAJ DE 1 A 99  
406 FIN  
407 ENTITE 12 ZILE-LUNI  
408 DEBUT  
409 ZIUA DE 28 A 31  
410 FIN  
411 ENTITE 30 L-TRASEUL-NAVEI  
412 DEBUT  
413 COD-PORT MOT 8  
414 DATA-INTRARE-IN-PORT DE 19790101 A 21001231  
415 NORMA-DE-OPERARE DE 0 A 600000  
416 DISTANTA DE 0 A 30000  
417 NR-ZILE DE 0 A 9999  
418 PROCENT-REDUCERE-VITEZA DE 0 A 99  
419 ENTITE 20 L-MARFA-OPERATA  
420 DEBUT  
421 COD-MARFA MOT 3  
422 TIP-OPERARE (2 4) (INC DESC)  
423 CANTITATE-TONE DE 0 A 300000  
424 CANTITATE-MC DE 0 A 300000  
425 FIN  
426 FIN  
427 ENTITE 100 L-NAVA

428 DEBUT  
429 COD-NAVA MOT 7  
430 COD-PORT MOT 8  
431 DATA-ELIBERARII  
432 DEBUT  
433 ZI DE 1 A 31  
434 LUNA DE 1 A 12  
435 AN DE 1979 A 2100  
436 FIN  
437 DATA-REPARATIE DE 19800101 A 21001231  
438 CHELTUIELI-OPERARE DE 0 A 1000000000  
439 CHELTUIELI-TRECERI DE 0 A 1000000000  
440 CHELTUIELI-PORTUARE DE 0 A 1000000000  
441 DISTANTA-GOL DE 0 A 100000  
442 DISTANTA-INC DE 0 A 100000  
443 DURATA-STATIONARE DE 0 A 999  
444 DURATA-STAT-RADA DE 0 A 999  
445 DURATA-MARS DE 0 A 999  
446 CONSUM-COMB-USOR-TO DE 0 A 1000000  
447 CONSUM-COMB-GREU-TO DE 0 A 1000000  
448 CONSUM-COMB-USOR-VAL DE 0 A 1000000000  
449 CONSUM-COMB-GREU-VAL DE 0 A 1000000000  
450 CHELTUIELI-FIXE DE 0 A 1000000000  
451 CHELTUIELI-INDEMNIZATII DE 0 A 1000000000  
452 TOTAL-CHELTUIELI DE 0 A 1000000000  
453 ENTITE 10 PORT-ESCALA  
454 DEBUT  
455 COD-PORT MOT 8  
456 FIN  
457 FIN  
458 ENTITE 1 ANGAJARI  
459 DEBUT  
460 NR-REALIZARE-NAVA DE 1 A 500  
461 NR-VOIAJE DE 0 A 600  
462 NR-REALIZARE-CD DE 1 A 100  
463 NR-NAVE DE 0 A 50  
464 CHELTUIELI-OPERARE DE 0 A 1000000000  
465 CHELTUIELI-TRECERI DE 0 A 1000000000  
466 CHELTUIELI-PORTUARE DE 0 A 1000000000  
467 DISTANTA-GOL DE 0 A 100000  
468 DISTANTA-INC DE 0 A 100000  
469 DURATA-STATIONARE DE 0 A 999  
470 DURATA-STAT-RADA DE 0 A 999  
471 DURATA-MARS DE 0 A 999  
472 CONSUM-COMB-USOR-TO DE 0 A 1000000  
473 CONSUM-COMB-GREU-TO DE 0 A 1000000  
474 CONSUM-COMB-USOR-VAL DE 0 A 1000000000  
475 CONSUM-COMB-GREU-VAL DE 0 A 1000000000  
476 CHELTUIELI-FIXE DE 0 A 1000000000  
477 CHELTUIELI-INDEMNIZATII DE 0 A 1000000000  
478 ENTITE 366 DISPONIBIL  
479 DEBUT  
480 REST-DISPONIBIL DE 0 A 500  
481 FIN  
482 FIN  
483 ENTITE 366 REALIZARI-1  
484 DEBUT  
485 NR-REALIZ-1 DE 1 A 366  
486 FIN  
487 ENTITE 200 REALIZARI-2  
488 DEBUT

S00DEF V3.1 27-AUG-87 11:56 PAGINA 9

489 NR-REALIZ-2 DE 1 A 366  
490 FIN  
491 ENTITE 20 PARAM  
492 DEBUT  
493 DATA-CURENTA  
494 DEBUT  
495 ZI DE 1 A 31  
496 LUNA DE 1 A 12  
497 AN DE 1980 A 2100  
498 FIN  
499 ZIUUA-CURENTA  
500 DEBUT  
501 ZI DE 1 A 31  
502 LUNA DE 1 A 31  
503 AN DE 1980 A 2100  
504 FIN  
505 IN MOT 27  
506 PENTRU MOT 27  
507 NAVA-TIP MOT 27  
508 MARFA MOT 27  
509 CANTITATE-MARFA DE 1 A 500000  
510 DORITI (2 2) (DA NU)  
511 CE-TIP-DE-SITUATII-DORITI (2 8) (ANALITIC TOTAL)  
512 ZI-INCEPUT DE 1 A 31  
513 LUNA-INCEPUT DE 1 A 12  
514 AN-INCEPUT DE 1980 A 2100  
515 ZI-SFIRSIT DE 1 A 31  
516 LUNA-SFIRSIT DE 1 A 12  
517 AN-SFIRSIT DE 1980 A 2100  
518 CONTINUAM (2 2) (DA NU)  
519 AVETI-INTRERUPERI (2 2) (DA NU)  
520 NUMAR-VOIAJ DE 0 A 999  
521 MODIFICARI (2 2) (DA NU)  
522 TIP1 DE 0 A 10000000  
523 TIP2 DE 0 A 10000000  
524 DORITI-PE-0-PERIOADA (2 2) (DA NU)  
525 FIN  
526 FORMAL RITMURI-LUNARE  
527 DEBUT  
528 FORMAL 12 RITM-LUNA  
529 DEBUT  
530 RITM DILATE 2  
531 FIN  
532 FIN  
533 FORMAL FISIER-CHELTUIELI  
534 DEBUT  
535 COD-NAVA MOT 7  
536 COD-GRUP-NAVE DILATE (0 2)  
537 COD-TIP-NAVE DILATE (2 2)  
538 REST-COD-NAVA DILATE (4 3)  
539 AN-OPERARE DILATE 4  
540 LUNA-OPERARE DILATE 2  
541 CHELTUIELI-PORTUARE DILATE 10  
542 CHELTUIELI-OPERARE DILATE 10  
543 COD-PORT MOT 8  
544 FIN  
545 FORMAL ACTIVITATE-IN-PORTURI  
546 DEBUT  
547 FILLER-1 MOT 1  
548 COD-NAVA MOT 7  
549 COD-GRUP-NAVE DILATE (1 2)

550 COD-TIP-NAVE DILATE (3 2)  
 551 REST-COD-NAVA DILATE (5 3)  
 552 NUMAR-CALATORIE DILATE 3  
 553 COD-PORT MOT 8  
 554 TIP-OPERARE DILATE 1  
 555 FILLER-2 MOT 20  
 556 ZI-SOSIT-RADA DILATE 2  
 557 LUNA-SOSIT-RADA DILATE 2  
 558 AN-SOSIT-RADA DILATE 4  
 559 ORA-SOSIT-RADA DILATE 2  
 560 MINUT-SOSIT-RADA DILATE 2  
 561 ZI-ACOSTAT-DANA DILATE 2  
 562 LUNA-ACOSTAT-DANA DILATE 2  
 563 AN-ACOSTAT-DANA DILATE 4  
 564 ORA-ACOSTAT-DANA DILATE 2  
 565 MINUT-ACOSTAT-DANA DILATE 2  
 566 ZI-INCEPUT-OPERARE DILATE 2  
 567 LUNA-INCEPUT-OPERARE DILATE 2  
 568 AN-INCEPUT-OPERARE DILATE 4  
 569 ORA-INCEPUT-OPERARE DILATE 2  
 570 MINUT-INCEPUT-OPERARE DILATE 2  
 571 ZI-TERMINAT-OPERARE DILATE 2  
 572 LUNA-TERMINAT-OPERARE DILATE 2  
 573 AN-TERMINAT-OPERARE DILATE 4  
 574 ORA-TERMINAT-OPERARE DILATE 2  
 575 MINUT-TERMINAT-OPERARE DILATE 2  
 576 ZI-DEPUS-NOTICE DILATE 2  
 577 LUNA-DEPUS-NOTICE DILATE 2  
 578 AN-DEPUS-NOTICE DILATE 4  
 579 ORA-DEPUS-NOTICE DILATE 2  
 580 MINUT-DEPUS-NOTICE DILATE 2  
 581 ZI-ACCEPTAT-NOTICE DILATE 2  
 582 LUNA-ACCEPTAT-NOTICE DILATE 2  
 583 AN-ACCEPTAT-NOTICE DILATE 4  
 584 ORA-ACCEPTAT-NOTICE DILATE 2  
 585 MINUT-ACCEPTAT-NOTICE DILATE 2  
 586 ZI-PLECAT-DIN-PORT DILATE 2  
 587 LUNA-PLECAT-DIN-PORT DILATE 2  
 588 AN-PLECAT-DIN-PORT DILATE 4  
 589 ORA-PLECAT-DIN-PORT DILATE 2  
 590 MINUT-PLECAT-DIN-PORT DILATE 2  
 591 COD-MARFA DILATE 3  
 592 CANTITATE-MARFA DILATE 6  
 593 DURATA-OPERARE DILATE 4  
 594 DURATA-STATIONARE DILATE 4  
 595 FORMAL 50 GRUPA-INTRERUPERE  
 596 DEBUT  
 597 COD-INTRERUPERE MOT 5  
 598 VALOARE-INTRERUPERE DILATE 6  
 599 FIN  
 600 FIN  
 601 FORMAL GRUP-NR-REALIZARI  
 602 DEBUT  
 603 FORMAL 360 GR-NR-REAL  
 604 DEBUT  
 605 NR-REALIZARE DILATE 5  
 606 FIN  
 607 FIN  
 608 FORMAL MARFURI-F  
 609 DEBUT  
 610 FORMAL 30 MARFA-F

IN ZILE SI 1  
 IN ZILE SI 1

611	DEBUT	672	FILLLLER MOT 1
612	COD-MARFA MOT 3	673	FIN
613	CANTITATE-TONE DILATE 6	674	FORMAL FISIER-NAVE
614	CANTITATE-MC DILATE 6	675	DEBUT
615	FIN	676	COD-NAVA MOT 7
616	FIN	677	COD-GRUP-NAVE DILATE (0 2)
617	FORMAL ORDONARE	678	COD-TIP-NAVE DILATE (2 2)
618	DEBUT	679	REST-COD-NAVA DILATE (4 3)
619	FORMAL 100 ORDINE	680	DENUMIRE-NAVA MOT 15
620	DEBUT	681	DENUMIRE-TIP MOT 15
621	TOTAL-CH DILATE 10	682	NUMAR-HAMBARE DILATE 1
622	NUMAR-REALIZ DILATE 3	683	AN-INTRARE-IN-EXPL DILATE 4
623	FIN	684	FILLER-1 MOT 6
624	FIN	685	CAPACITATE-TDW DILATE 6
625	FORMAL MEM-MARFA	686	CAPACITATE-MC DILATE 6
626	DEBUT	687	FILLER-2 MOT 3
627	FORMAL 20 GRUPE-MF	688	VITEZA-IMPUSA-PE-ORĂ DILATE 3
628	DEBUT	689	CAPACITATE-TANC-COMBUSTIBIL DILATE 5
629	COD-MARFA MOT 3	690	CONSUM-COMBUSTIBIL-MILA DILATE 4
630	CANTITATE-TONE DILATE 6	691	CONSUM-COMBUSTIBIL-ZI DILATE 5
631	FIN	692	FILLER-3 MOT 21
632	FIN	693	INDEMNIZATIE-PE-ZI DILATE 4
633	FORMAL SEMILUNA	694	CHELTUIELI-FIXE-ZI DILATE 6
634	DEBUT	695	DEPLASAMENT DILATE 6
635	FORMAL 16 ANGAJATE	696	FILLER-4 MOT 3
636	DEBUT	697	FIN
637	NUMAP DILATE 3	698	FORMAL FISIER-PORTURI
638	FIN	699	DEBUT
639	FIN	700	COD-PORT MOT 8
640	FORMAL FACTMAR	701	COD-ZONA DILATE (0 2)
641	DEBUT	702	COD-TARA DILATE (2 3)
642	COD-NAVA MOT 7	703	REST-COD-PORT DILATE (5 3)
643	NR-CALATORIE DILATE 3	704	NUME-PORT MOT 25
644	NR-PORT DILATE 1	705	NUME-TARA MOT 20
645	ZI DILATE 2	706	NUME-ZONA MOT 18
646	LUNA DILATE 2	707	FILL-1 MOT 1
647	AN DILATE 2	708	FIN
648	COD-PORT MOT 8	709	FORMAL TOTINTR
649	VITEZA-IMPUSA DILATE 3	710	DEBUT
650	DISTANTA-RAHASA DILATE 5	711	FORMAL 40 TOTINTR1
651	MILE-INC DILATE 5	712	DEBUT
652	MILE-GOL DILATE 5	713	CODINTR MOT 5
653	MARS-INC DILATE 4	714	TOTALINTR DILATE 10
654	MARS-GOL DILATE 4	715	TIPINTR DILATE 1
655	STATIONARE DILATE 4	716	DENINTR MOT 28
656	STAT-AFARA DILATE 4	717	ZILEINT DILATE 7
657	REPARATII DILATE 4	718	OREINT DILATE 3
658	FORMAL 10 INTRERUPERI	719	FIN
659	DEBUT	720	FIN
660	COD-INTR DILATE 3	721	FORMAL CAPACIT-NAVA
661	DURATA DILATE 4	722	DEBUT
662	FIN	723	TIPC1 MOT 2
663	DENUMIRE-NAVA MOT 20	724	CODNAVI MOT 7
664	FORMAL 31 ZILE	725	FORMAL 6 CAPACITATI
665	DEBUT	726	DEBUT
666	ZIUA DILATE 2	727	CAPACITATE DILATE 6
667	FIN	728	FIN
668	FORMAL 40 GR-NR-REAL	729	FILL MOT 35
669	DEBUT	730	FIN
670	NR-REALIZARE DILATE 5	731	FORMAL COMB-CONS-VITEZA
671	FIN	732	DEBUT



730	TIPC2 MOT 2	794	CODPORT MOT 8
734	CODNAV2 MOT 7 NRCART2 DILATE 1	795	LATGRAD DILATE 2
735	FORMAL 4 TIPCOMB	796	LATMIN DILATE 2
736	DEBUT	797	EMISLAT MOT 1
737	TIP-COMBUST MOT 2	798	LONGRAD DILATE 3
738	FIN	799	LONGMIN DILATE 2
739	CONS1STAT DILATE 6	800	EMISLONG MOT 1
740	CONS2STAT DILATE 6	801	ZVNG DILATE 2
741	VITNOD DILATE 3	802	ZVNM DILATE 2
742	FORMAL 5 VIT-CONSUM	803	ZVSG DILATE 2
743	DEBUT	804	ZVSM DILATE 2
744	VITEZANOD DILATE 3	805	ZVEG DILATE 2
745	CONSUMNOD DILATE 6	806	ZVEM DILATE 2
746	FIN	807	ZVVG DILATE 2
747	FIL2 MOT 2	808	ZVVM DILATE 2
748	FIN	809	FIL5 MOT 43
749	FORMAL CONS-VITEZA	810	FIN
750	DEBUT	811	FORMAL SUBACTIVITATI
751	TIPC21 MOT 2	812	DEBUT
752	CODNAV21 MOT 7	813	TIPC6 MOT 2
753	NRCART21 DILATE 1	814	TIPACT1 MOT 1
754	FORMAL 5 VIT-CONSUM1	815	CGD-SUBACT MOT 5
755	DEBUT	816	DENUMSUBACT MOT 28
756	VITEZANOD1 DILATE 3	817	TIPSUBACT MOT 4
757	CONSUMNOD1 DILATE 6	818	FIL6 MOT 40
758	FIN	819	FIN
759	CONS-MARS-AUX DILATE 6	820	FORMAL GRUPE-MARFA
760	FIL21 MOT 19	821	DEBUT
761	FIN	822	TIPC7 MOT 2
762	FORMAL REPARATII	823	TIPACT2 MOT 1
763	DEBUT	824	COD-GRMARF MOT 3
764	TIPC3 MOT 2	825	DENMARF MOT 25
765	CODNAV3 MOT 7	826	TIPMARFA MOT 5
766	ZIEXP DILATE 2	827	FIL7 MOT 44
767	LNEXP DILATE 2	828	FIN
768	ANEXP DILATE 4	829	FORMAL GRUPE-COMB
769	ZIPL DILATE 2	830	DEBUT
770	LNPL DILATE 2	831	TIPC8 MOT 2
771	ANPL DILATE 4	832	TIPACT3 MOT 1
772	TIPREPAR MOT 3	833	CGDCOMB MOT 2
773	DPLANREP DILATE 3	834	DENCOMB MOT 22
774	DPLANPREG DILATE 2	835	FRET-TONA DILATE 5
775	FIL3 MOT 47	836	FIL8 MOT 48
776	FIN	837	FIN
777	FORMAL CARACT-TEH	838	FORMAL STERG-ACT-OPERATIV
778	DEBUT	839	DEBUT
779	TIPC4 MOT 2	840	TIPC3 MOT 3
780	CODNAV4 MOT 7	841	CODNAVFL1 MOT 7
781	SECTIA1 DILATE 1	842	Z11 DILATE 2
782	PESCAJ1 DILATE 5	843	LN1 DILATE 2
783	AFUNDARE DILATE 4	844	ANI DILATE 4
784	CHELTFI DILATE 6	845	FIL3 MOT 62
785	VALFONDFIX DILATE 6	846	FIN
786	INDHRANA DILATE 4	847	FORMAL MARFA
787	INDSTRAIN DILATE 4	848	DEBUT
788	TIPANGAJ MOT 1	849	COD-MARFA MOT 3
789	FIL4 MOT 40	850	DEN-MARFA MOT 25
790	FIN	851	TPP-MARFA MOT 5
791	FORMAL POZITIE-PORT	852	PLUS MOT 3
792	DEBUT	853	FIN
793	TIPC5 MOT 2	854	FORMAL SUBACTIVITATE

OCDEF V3.1 27-AUG-87 11:56 PAGINA 15

855 DEBUT  
856 COD-SUBACT MOT 5  
857 DEN-SUBACT MOT 28  
858 TIP-SUBACT MOT 4  
859 PLUS MOT 3  
860 FIN  
861 FORMAL COMBUSTIBIL  
862 DEBUT  
863 COD-COMBUST MOT 2  
864 DEN-COMBUST MOT 22  
865 PRET-TOCOMB DILATE 5  
866 PLUS MOT 3  
867 FIN  
868 FORMAL NAVAR  
869 DEBUT  
870 COD-NAVA MOT 7  
871 DENUMIRE-NAVA MOT 20  
872 GRUP-NAVE MOT 3  
873 AN-INTRARE-IN-EXPLOATARE DILATE 4  
874 CAPACITATE-TONE-CARGO-VARA DILATE 6  
875 CAPACITATE-TONE-CARGO-IARNA DILATE 6  
876 CAPACITATE-TDW DILATE 6  
877 CAPACITATE-MC DILATE 6  
878 SECTIA DILATE 1  
879 PESCAJ DILATE 5  
880 AFUNDAREA-IN-TO-PE-INCH DILATE 4  
881 VITEZA-IMPUSA DILATE 3  
882 VALOARE-FONDURI-FIXE DILATE 6  
883 CHELTUIELI-FIXE-PE-ZI DILATE 6  
884 INDEMNIZATIE-HRANA-DOLARI DILATE 4  
885 INDEMNIZATIE-STRAINATATE-DOL DILATE 4  
886 NUMAR-HAMBARE DILATE 1  
887 TIP-ANGAJARE-NAVA MOT 1  
888 ZI-EXPIRARE-ACTE-REPARATIE DILATE 2  
889 LUNA-EXPIRARE-ACTE-REPARATIE DILATE 2  
890 AN-EXPIRARE-ACTE-REPARATIE DILATE 4  
891 TIP-REPARATIE MOT 3  
892 DURATA-PLAN-REPARATIE DILATE 3  
893 DURATA-PLAN-PREG-SANTIER DILATE 2  
894 ZI-INCEPERE-REPARATIE-PLAN DILATE 2  
895 LUNA-INCEPERE-REPARATIE-PLAN DILATE 2  
896 AN-INCEPERE-REPARATIE-PLAN DILATE 4  
897 TIP-COMBUSTIBIL-MARS-PR MOT 2  
898 TIP-COMBUSTIBIL-MARS-AUX MOT 2  
899 TIP-COMBUSTIBIL-1-STATIONARE MOT 2  
900 TIP-COMBUSTIBIL-2-STATIONARE MOT 2  
901 CAPACITATE-TANC-COMBUST-KG DILATE 6  
902 CANTIT-CONSUM-1-ZI-STAT-KG DILATE 6  
903 CANTIT-CONSUM-2-ZI-STAT-KG DILATE 6  
904 FORMAL 10 CONS-COMB  
905 DEBUT  
906 VITEZA DILATE 3  
907 CONSUM DILATE 6  
908 FIN  
909 PLUS MOT 3  
910 FIN  
911 FORMAL PORTUL  
912 DEBUT  
913 COD-PORT MOT 8  
914 DENUMIRE-PORT MOT 25  
915 DENUMIRE-TARA MOT 20

916	DENUMIRE-ZONA MOT 18	977	FIL5 MOT 19
917	TIP-PORT MOT 10	978	FORMAL 10 GR-COMBUSTIBIL-B
918	EMISFERA-LATITUDINII MOT 1	979	DEBUT
919	EMISFERA-LONGITUDINII MOT 1	980	VITEZA-B DILATE 4
920	LATITUDINEA-GRADE DILATE 2	981	CONSUM-B DILATE 9
921	LATITUDINEA-MINUTE DILATE 2	982	FIN
922	LONGITUDINEA-GRADE DILATE 3	983	FIL10 MOT 1
923	LONGITUDINEA-MINUTE DILATE 2	984	FIN
924	ZONA-VEGINATATE-NORD-GRADE DILATE 2	985	FORMAL F-CAPDISP
925	ZONA-VEGINATATE-NORD-MINUTE DILATE 2	986	DEBUT
926	ZONA-VEGINATATE-SUD-GRADE DILATE 2	987	FIL1 DILATE 1
927	ZONA-VEGINATATE-SUD-MINUTE DILATE 2	988	GRUP-NAVE-CD MOT 8
928	ZONA-VEGINATATE-EST-GRADE DILATE 2	989	NR-SUTEI-DE-ZILE-CD DILATE 1
929	ZONA-VEGINATATE-EST-MINUTE DILATE 2	990	ANUL-CD DILATE 4
930	ZONA-VEGINATATE-VEST-GRADE DILATE 2	991	CAPACITATE-CARGO-VARA-CD DILATE 9
931	ZONA-VEGINATATE-VEST-MINUTE DILATE 2	992	CAPACITATE-CARGO-IARNA-CD DILATE 9
932	FIN	993	CAPACITATE-MC-CD DILATE 9
933	FORMAL F-NAVA	994	FORMAL 366 ZILE-CB
934	DEBUT	995	DEBUT
935	FIL1 DILATE 1	996	DISPONIBILE-CB DILATE 4
936	COD-NAVA-N MOT 7	997	ANGAJATE-CB DILATE 4
937	DENUMIRE-NAVA-N MOT 20	998	FIN
938	AN-IE-N DILATE 4	999	FORMAL 9 PESCAJ-AF-TO-INC-CB
939	LUNA-IE-N DILATE 2	1000	DEBUT
940	ZI-IE-N DILATE 2	1001	CANTITATE-MARFA-CB DILATE 9
941	AN-ER-N DILATE 4	1002	PESCAJ-CB DILATE 9
942	LUNA-ER-N DILATE 2	1003	AFUNDARE-CB DILATE 4
943	ZI-ER-N DILATE 2	1004	FIN
944	AN-IR-N DILATE 4	1005	FIL2 MOT 1
945	LUNA-IR-N DILATE 2	1006	FIN
946	ZI-IR-N DILATE 2	1007	FORMAL F-TRASNAVA
947	TIP-COMBUSTIBIL-MARS-MOT-PR-N MOT 2	1008	DEBUT
948	TIP-COMBUSTIBIL-MARS-MOT-AUX-N MOT 2	1009	FORMAL 5 F-TRASNAV1
949	TIP-COMBUSTIBIL-1-STATIONARE-N MOT 2	1010	DEBUT
950	TIP-COMBUSTIBIL-2-STATIONARE-N MOT 2	1011	FIL1 DILATE 1
951	FIL2 MOT 24	1012	COD-NAVA-TN MOT 7
952	TIP-ANGAJARE-NAVA-N MOT 1	1013	NR-PORT-TN DILATE 2
953	TIP-REPARATIE-N MOT 3	1014	COD-PORT-TN MOT 8
954	FIL3 MOT 2	1015	DATA-INTRARE-IN-PORT-TN DILATE 8
955	SECTIA-N DILATE 1	1016	NORMA-DE-OPERARE-TN DILATE 9
956	CAPACITATE-TONE-CARGO-VARA-N DILATE 9	1017	DISTANTA-TN DILATE 9
957	CAPACITATE-TONE-CARGO-IARNA-N DILATE 9	1018	NR-ZILE-TN DILATE 4
958	CAPACITATE-TDW-N DILATE 9	1019	PROCENT-REDUCERE-VITEZA-TN DILATE 2
959	CAPACITATE-TRN-N DILATE 9	1020	FORMAL 7 MARFA-OPERATA-TB
960	CAPACITATE-TRB-N DILATE 9	1021	DEBUT
961	CAPACITATE-MC-N DILATE 9	1022	COD-MARFA-TB MOT 3
962	VITEZA-IMPUSA-N DILATE 4	1023	TIP-OPERARE-TB DILATE 1
963	PESCAJ-N DILATE 9	1024	CANTITATE-TONE-TB DILATE 9
964	CAPACITATE-TANC-COMBUST-TO-N DILATE 9	1025	CANTITATE-MC-TB DILATE 9
965	DURATA-PLAN-REPARATIE-N DILATE 4	1026	FIN
966	DURATA-PLAN-PREG-SANTIER-N DILATE 4	1027	FIN
967	CHELTUIELI-FIXE-PE-ZI-N DILATE 9	1028	FIN
968	AFUNDAREA-IN-TO-PE-INCH-N DILATE 4	1029	FORMAL F-PORT
969	VALOARE-FONDURI-FIXE-IN-MII-N DILATE 9	1030	DEBUT
970	INDEMNIZATIA-HRANA-DOLARI-N DILATE 4	1031	FIL1 DILATE 1
971	INDEMNIZATIA-STRAINAT-DOLARI-N DILATE 4	1032	COD-PORT-P MOT 8
972	CANTIT-CONSUM-1-ZI-STAT-KG-N DILATE 9	1033	DENUMIRE-PORT-P MOT 25
973	CANTIT-CONSUM-2-ZI-STAT-KG-N DILATE 9	1034	DENUMIRE-TARA-P MOT 20
974	CANTIT-CONSUM-MARS-AUX-N DILATE 9	1035	DENUMIRE-ZONA-P MOT 18
975	FIL4 DILATE 4	1036	TIP-PORT-P MOT 1
976	NUMAR-HAMBARE-N DILATE 2	1037	EMISFERA-LATITUDINII-P MOT ,

1038	EMISFERA-LONGITUDINII-P MOT 1	1099	AN-PP-AP DILATE 4
1039	LĂTITUDINEA-GRADE-P DILATE 2	1100	LUNA-PP-AP DILATE 2
1040	LĂTITUDINEA-MINUTE-P DILATE 2	1101	ZI-PP-AP DILATE 2
1041	LONGITUDINEA-GRADE-P DILATE 3	1102	ORA-PP-AP DILATE 2
1042	LONGITUDINEA-MINUTE-P DILATE 2	1103	MINUT-PP-AP DILATE 2
1043	NR-NAVE-ZILE-ASTEPT-RADA-PET-P DILATE 2	1104	NUMAR-CALATORIE-AP DILATE 3
1044	NR-NAVE-ZILE-ASTEPT-RADA-MIN-P DILATE 2	1105	DURATA-STATIONARE-AP DILATE 4
1045	NR-NAVE-ZILE-ASTEPT-RADA-CAR-P DILATE 2	1106	DURATA-OPERARE-AP DILATE 4
1046	ZILE-ASTEPT-RADA-PET-P DILATE 4	1107	AN-SR-AP DILATE 4
1047	ZILE-ASTEPT-RADA-MIN-P DILATE 4	1108	LUNA-SR-AP DILATE 2
1048	ZILE-ASTEPT-RADA-CAR-P DILATE 4	1109	ZI-SR-AP DILATE 2
1049	VECINATATE-NORD-GRADE-P DILATE 2	1110	ORA-SR-AP DILATE 2
1050	TONE-ZILE-OPERARE-PET-P DILATE 9	1111	MINUT-SR-AP DILATE 2
1051	VECINATATE-NORD-MINUTE-P DILATE 2	1112	AN-AC-AP DILATE 4
1052	TONE-ZILE-OPERARE-MIN-P DILATE 9	1113	LUNA-AC-AP DILATE 2
1053	VECINATATE-SUD-GRADE-P DILATE 2	1114	ZI-AC-AP DILATE 2
1054	TONE-ZILE-OPERARE-CAR-MFU-P DILATE 9	1115	ORA-AC-AP DILATE 2
1055	VECINATATE-SUD-MINUTE-P DILATE 2	1116	MINUT-AC-AP DILATE 2
1056	VECINATATE-EST-GRADE-P DILATE 3	1117	AN-DN-AP DILATE 4
1057	VECINATATE-EST-MINUTE-P DILATE 2	1118	LUNA-DN-AP DILATE 2
1058	VECINATATE-VEST-GRADE-P DILATE 3	1119	ZI-DN-AP DILATE 2
1059	VECINATATE-VEST-MINUTE-P DILATE 2	1120	ORA-DN-AP DILATE 2
1060	TONE-ZILE-OPERARE-CAR-MFM-P DILATE 9	1121	MINUT-DN-AP DILATE 2
1061	TONE-ZILE-OPERARE-CAR-MFG-P DILATE 9	1122	AN-ACC-AP DILATE 4
1062	ZILE-OPERARE-PET-P DILATE 4	1123	LUNA-ACC-AP DILATE 2
1063	ZILE-OPERARE-MIN-P DILATE 4	1124	ZI-ACC-AP DILATE 2
1064	ZILE-OPERARE-CAR-MFU DILATE 4	1125	ORA-ACC-AP DILATE 2
1065	ZILE-OPERARE-CAR-MFM DILATE 4	1126	MINUT-ACC-AP DILATE 2
1066	ZILE-OPERARE-CAR-MFG DILATE 4	1127	AN-IO-AP DILATE 4
1067	FORMAL 25 CHELT-PORT-INC-DESC-B	1128	LUNA-IO-AP DILATE 2
1068	DEBUT	1129	ZI-IO-AP DILATE 2
1069	AN-OPERARE-B DILATE 4	1130	ORA-IO-AP DILATE 2
1070	LUNA-OPERARE-B DILATE 2	1131	MINUT-IO-AP DILATE 2
1071	TDW-B DILATE 9	1132	AN-TO-AP DILATE 4
1072	CHELT-FORT-PER-TONA-B DILATE 9	1133	LUNA-TO-AP DILATE 2
1073	CHELT-OPER-PER-TONA-B DILATE 9	1134	ZI-TO-AP DILATE 2
1074	FIN	1135	ORA-TO-AP DILATE 2
1075	FIL13 MOT 2	1136	MINUT-TO-AP DILATE 2
1076	FIN	1137	FORMAL 50 INTRERUPERI-I-APB
1077	FORMAL F-FSMC	1138	DEBUT
1078	DEBUT	1139	COD-INTRERUPERE-I-APB MOT 5
1079	FIL1 DILATE 1	1140	DURATA-I-APB DILATE 9
1080	TIP-ARTICOL-FS DILATE 1	1141	FIN
1081	COD-SUBACTIVITATE-FS MOT 5	1142	FIL5 MOT 3
1082	DENUMIRE-FS MOT 25	1143	FIN
1083	TIP-SUBACTIVITATE-FS MOT 4	1144	FORMAL F-ACTOPER
1084	FIL2 MOT 16	1145	DEBUT
1085	TIP-MARFA-FS MOT (32 5)	1146	FORMAL 93 NR-REALIZ-AO
1086	FIL3 MOT (37 15)	1147	DEBUT
1087	PRET-PER-TONA-LEI-FS DILATE (32 9)	1148	NR-REALIZARE DILATE 6
1088	PRET-PER-TONA-DOL-FS DILATE (41 9)	1149	FIN
1089	FIL7 MOT (50 2)	1150	COD-NAVA-ABB MOT 7
1090	FIN	1151	COD-PORT-ACTUAL-N MOT 8
1091	FORMAL F-ACTPORT	1152	COD-PORT-ULTIM-FLECARE-N MOT 8
1092	DEBUT	1153	COD-PORT-DESTINATIE-N MOT 8
1093	FIL1 DILATE 1	1154	NUMAR-VOIAJ-N DILATE 3
1094	COD-PORT-AP MOT 8	1155	DIST-PINA-LA-DESTINATIE-N DILATE
1095	COD-NAVA-AP MOT 7	1156	TIP-ACT-NAVA-N DILATE 1
1096	TIP-OPERARE-AP DILATE 1	1157	FIL1 DILATE 1
1097	COD-MARFA-AP MOT 3	1158	AN-AO DILATE 4
1098	CANTITATE-MARFA-AP DILATE 6	1159	LUNA-AO DILATE 2

```

BCDEF V3.1 27-AUG-87 11:57 PAGINA 20
1160 ZI-AO DILATE 2
1161 COD-NAVA-AO MOT 7
1162 COD-PORT-ULTIM-AO MOT 8
1163 NUMAR-CALATORIE-AO DILATE 3
1164 CONSUM-COMB-GREU-AO DILATE 6
1165 CONSUM-COMB-USOR-AO DILATE 6
1166 FORMAL 6 SUBACTIVITATE-AB
1167 DEBUT
1168 COD-PORT-AB MOT 8
1169 COD-SUBACTIVITATE-AB MOT 5
1170 MOD-PARCURS-MILE-AB MOT 1
1171 DURATA-MINUTE-S-AB DILATE 4
1172 DANA-AB MOT 4
1173 MILE-PARCURSE-AB DILATE 4
1174 CANTITATE-MARFA-OPERATA-AB DILATE 9
1175 FORMAL 3 INTR-ACT-AB
1176 DEBUT
1177 COD-INTR-I-AB MOT 5
1178 DURATA-MINUTE-I-AB DILATE 4
1179 FIN
1180 FIN
1181 FIL2 MOT 3
1182 FIN
1183 FORMAL F-INCBZ1
1184 DEBUT
1185 TIPART DILATE 1
1186 FORMAL 90 F-REST
1187 DEBUT
1188 FIL1 MOT 10
1189 FIN
1190 FORMAL 39 W-LUZ
1191 DEBUT
1192 WLUNAZ DILATE 3
1193 FIN
1194 FIL2 MOT 2
1195 /* ARTPARAM */
1196 TIP-ARTICOL-FS DILATE 1 1
1197 DATA1-AO-W DILATE 1 8
1198 AN1-AO-W DILATE 1 4
1199 LN1-AO-W DILATE 5 2
1200 ZI1-AO-W DILATE 7 2
1201 DENUMIRE-NAVA MOT 8 20
1202 DATA2-AO-W DILATE 9 8 DATA2-Z5 MOT 9 8
1203 AN2-AO-W DILATE 9 4
1204 LN2-AO-W DILATE 13 2
1205 ZI2-AO-W DILATE 15 2
1206 NRFISTRW DILATE 17 12
1207 NRFISTRW7 DILATE 29 2
1208 NRFISTRW8 DILATE 31 2
1209 /* F-ACTOPER */
1210 COD-NAVA-ABB MOT 558 7
1211 COD-NAVA-AO MOT 611 7
1212 /* F-TRASNAVA */
1213 COD-NAVA-TN MOT 1 7
1214 /* F-ACT-PORT */
1215 COD-NAVA-AP DILATE 9 7
1216 W-DATA-INC DILATE 39 8
1217 AN-DI DILATE 39 4
1218 LUNA-DI DILATE 43 2
1219 ZI-DI DILATE 45 2
1220 W-LUNA-DI DILATE 47 9
1221 W-AN-DI DILATE 56 9
1222 NR-ZILE DILATE 65 9
1223 N2 DILATE 74 9
1224 NNNNNNN2 MOT 1018 2
1225 FIN
1226 FIN
1227 ?
*** STRUCTURA CATALOGATA ***

```

\*\*\* DOMENIUL STR - 43712 OCTETI \*\*\*

\*\*\* DOMENIUL DIC - 3105000 OCTETI \*\*\*

\*\*\* DOMENIUL FIS - 88569984 OCTETI \*\*\*

TIP	IDENTIFICATOR	N.R.	DIMENS.
ENTI	NAVA	500	20768
ENTI	PORT	5000	992
ENTI	ACT-OPERATIVA	50000	832
ENTI	SUBACT-INTR	500	64
ENTI	GRUPA-MARFA	100	32
ENTI	TIP-COMBUSTIBIL	100	32
ENTI	ACT-PORTUARA	15000	1888
ENTI	PORT-NAVE	300	6464
ENTI	CAPACITATI-DISPONIBILE	100	12160
ENTI	DATE-PAR	1	768
ENTI	ZILE-LUNI	12	32
ENTI	L-TRASEUL-NAVEI	30	672
ENTI	L-NAVA	100	416
ENTI	ANGAJARI	1	11840
ENTI	REALIZARI-1	366	32
ENTI	REALIZARI-2	200	32
ENTI	PARAM	20	192
FORM	RITMURI-LUNARE	1	24
FORM	FISIER-CHELTUIELI	1	41
FORM	ACTIVITATE-IN-PORTURI	1	691
FORM	GRUP-NR-REALIZARI	1	1800
FORM	MARFURI-F	1	450
FORM	ORDONARE	1	1300
FORM	MEM-MARFA	1	180
FORM	SEMILUNA	1	48

FORM	FACTMAR	1	416
FORM	FISIER-NAVE	1	120
FORM	FISIER-PORTURI	1	72
FORM	TOTINTR	1	2160
FORM	CAPACIT-NAVA	1	80
FORM	COMB-CONS-VITEZA	1	80
FORM	CONS-VITEZA	1	80
FORM	REPARATII	1	80
FORM	CARACT-TEH	1	80
FORM	POZITIE-PORT	1	80
FORM	SUBACTIVITATI	1	80
FORM	GRUPE-MARFA	1	80
FORM	GRUPE-COMB	1	80
FORM	STERG-ACT-OPERATIVA	1	80
FORM	MARFA	1	36
FORM	SUBACTIVITATE	1	40
FORM	COMBUSTIBIL	1	32
FORM	NAVAR	1	236
FORM	PORTUL	1	108
FORM	F-NAVA	1	388
FORM	F-CAPDISP	1	3168
FORM	F-TRASNAVA	1	1020
FORM	F-PORT	1	1012
FORM	F-FSMC	1	52
FORM	F-ACTPORT	1	824
FORM	F-ACTOPER	1	1016
FORM	F-INCBAZ1	1	1020

STRFLOTA.LST=FLOTA/CVF

### MODUL PEOF 3. Cheltuielile voiajului pentru diferitele nave

```

1  :DEFFRO PEOF3 :EXP
2  D X1=UN NAVA D X7 D X8 D X9
3  D X2=UN PORT
4  D X3=UN L-NAVA
5  D X4=UN L-TRASEUL-NAVEI
6  D X5=UN L-MARFA-OPERATA DE UN L-TRASEUL-NAVEI
7  D X6=UN DATE-PAR
8  M X6=UN DATE-PAR
9  M X7=FORMAL ORDONARE DANS BUFFER 1
10 M Y1=1
11 FAIRE
12 M X8=FORMAL ORDINE Y1 DE X7
13 M TOTAL-CH DE X8=0
14 M NUMAR-REALIZ DE X8=0 M Y1=Y1 + 1
15 SI Y1 < 100 ALORS REFAIRE FIN
16 FIN
17 M Y2=0
18 M Z1=COD-PORT DE UN L-TRASEUL-NAVEI 1
19 POUR TOUT L-NAVA X3
20 M Z2=COD-PORT DE X3 M Y3=0 M Y1=0
21 M CHELTUIELI-OPERARE DE X3=0
22 M CHELTUIELI-TRECERI DE X3=0
23 M CHELTUIELI-PORTUARE DE X3=0
24 SI Z1 ^= Z2 ALORS
25 I (1 6) 'PENTRU'
26 M Z3=COD-NAVA DE X3
27 I (+1 15) DENUMIRE-NAVA DE UN NAVA AVEC COD-NAVA=Z3 ;
28 DE X0
29 I (+1 10) 'PE RELATIA'
30 I (+1 20) DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z2 ;
31 DE X0
32 I (+1 1) '...'
33 I (+1 20) DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z1 ;
34 DE X0
35 I (+1 8) 'COMUNICA' ECRIRE
36 M DISTANTA DE X6=U
37 M DISTANTA DE X6=EXT
38 SI DISTANTA DE X6=U ALORS
39 CALCUL-DIST-INIT
40 SINON
41 M Y12=DISTANTA DE X6
42 FIN
43 SINON
44 M Y12=0
45 FIN
46 M Y2=0
47 POUR TOUT L-TRASEUL-NAVEI X4 DE X0
48 M Y11=DISTANTA DE X4
49 SI Y11=U ALORS M Y11=0 FIN
50 M Y1=Y1 + Y11
51 M Z3=COD-PORT DE X4 M Z4=COD-NAVA DE X3
52 M Y8=0
53 POUR TOUT L-MARFA-OPERATA X5
54 M Z5=COD-MARFA DE X5 M Y10=CANTITATE-TONE DE X5
55 M Y8=Y8 + Y10
56 FIN
57 M IN DE X6=Z5
58 M Z5=TIP-PORT DE UN PORT AVEC COD-PORT=Z3 ; DE X0
59 CALCUL-CHELTUIELI-PORT
60 M Y25=CHELTUIELI-OPERARE DE X3
61 M Y25=Y25 + Y21

```



```

62 M CHELTUIELI-OPERARE DE X3=Y25
63 SI Z5='CANAL' OU Z5='STRIMTOARE' ALORS
64 M Y25=CHELTUIELI-TRECERI DE X3
65 M Y25=Y25 + Y22
66 M CHELTUIELI-TRECERI DE X3=Y25
67 SINON
68 M Y25=CHELTUIELI-PORTUARE DE X3
69 M Y25=Y25 + Y22
70 M CHELTUIELI-PORTUARE DE X3=Y25
71 FIN
72 M Z5=IN DE X6
73 M Z3=COD-PORT DE X4
74 M Z4=COD-NAVA DE X3
75 M X1=UN NAVA AVEC COD-NAVA=Z4 ; DE X0
76 M Y25=NUMDE X4
77 M Z4=U M Z4=TIP-MARFA DE UN GRUPA-MARFA AVEC
78 COD-GRUPA-MARFA=Z5 ; DE X0
79 SI Z4=U ALORS M Z4='MEDIU' FIN M Y18=0
80 SI NORMA-DE-OPERARE DE X4 ^= U ALORS
81 SI Y25=1 ET Z2=Z3 ALORS M Y23=0 SINON
82 CALCUL-STATIONARE M Y23=Y21
83 FIN
84 M Y10=NORMA-DE-OPERARE DE X4
85 M Y18=Y3 / Y10 M Y21=Y8 - Y18 * Y10
86 M Y22=Y10 / 2
87 SI Y21 > Y22 ALORS M Y18=Y18 +1 FIN
88 M Y18=Y18 + Y23
89 M Y2=Y2 + Y23
90 SINON
91 SI Y25=1 ET Z2=Z3 ALORS
92 CALCUL-STATIONARE M Y18=Y18 - Y21
93 SINON
94 CALCUL-STATIONARE
95 M Y2=Y2 + Y21
96 FIN
97 FIN
98 M Y3=Y3 + Y18
99 FIN
100 M Y4=VITEZA-IMPUSA DE X1
101 M Y22=Y12 * 100 / Y4
102 M Y23=Y22 / 100
103 M Y22=Y22 - Y23 * 100
104 SI Y22 > 49 ALORS M Y23=Y23 + 1 FIN
105 SI Y22 > 24 ET Y23=0 ALORS M Y23=1 FIN
106 M Y22=Y1 * 100 / Y4
107 M Y24=Y22 / 100
108 M Y22=Y22 - Y24 * 100
109 SI Y22 > 49 ALORS M Y24=Y24 + 1 FIN
110 M Y6=Y3 + Y23 + Y24
111 M DISTANTA-GOL DE X3=Y12
112 M DISTANTA-INC DE X3=Y1
113 M DURATA-STATIONARE DE X3=Y3
114 M DURATA-STAT-RADA DE X3=Y2
115 M Y23=Y23 + Y24
116 M DURATA-MARS DE X3=Y23
117 M Y5=0
118 CONSUM-COMBUSTIBIL-MARS
119 M CONSUM-COMB-USOR-TO DE X3=Y21
120 M CONSUM-COMB-GREU-TO DE X3=Y22
121 M CONSUM-COMB-USOR-VAL DE X3=Y23
122 M CONSUM-COMB-GREU-VAL DE X3=Y24

```

```

SOCMAG      V3.1  27-AUG-87 13:14  PAGINA  3

123 M Y5=1 CONSUM-COMBUSTIBIL-MARS M Y5=2
124 FAIRE
125 M Y25=CONSUM-COMB-USOR-TO DE X3
126 M Y25=Y21 + Y25
127 M CONSUM-COMB-USOR-TO DE X3=Y25
128 M Y25=CONSUM-COMB-GREU-TO DE X3
129 M Y25=Y22 + Y25
130 M CONSUM-COMB-GREU-TO DE X3=Y25
131 M Y25=CONSUM-COMB-USOR-VAL DE X3
132 M Y25=Y25 + Y23
133 M CONSUM-COMB-USOR-VAL DE X3=Y25
134 M Y25=CONSUM-COMB-GREU-VAL DE X3
135 M Y25=Y24 + Y25
136 M CONSUM-COMB-GREU-VAL DE X3=Y25
137 SI Y5=2 ALORS M Y5=3
138 CONSUM-COMBUSTIBIL-STATIONNARE
139 REFAIRE
140 FIN
141 FIN
142 M Y1=CHELTUIELI-FIXE-PE-ZI DE X1
143 M Y2=CURS-LEU-DOLAR DE X6
144 M Y1=Y1 * 5 / Y2
145 M Y3=INDEMNIZATIA-HRANA-DOLARI DE X1
146 M Y4=INDEMNIZATIA-STRAINATATE-DOL DE X1
147 M Y2=Y1 * Y6
148 M CHELTUIELI-FIXE DE X3=Y2
149 M Y2=(Y3 + Y4) * Y6
150 M CHELTUIELI-INDEMNIZATII DE X3=Y2
151 M Y21=CHELTUIELI-OPERARE DE X3
152 M Y22=CHELTUIELI-PORTUARE DE X3
153 M Y23=CHELTUIELI-TRECERI DE X3
154 M Y24=CONSUM-COMB-GREU-VAL DE X3
155 M Y19=CONSUM-COMB-USOR-VAL DE X3
156 M Y18=CHELTUIELI-FIXE DE X3
157 M Y18=Y18 + Y19 + Y21 + Y22 + Y24 + Y23 + Y2
158 M TOTAL-CHELTUIELI DE X3=Y18
159 M Y21=1
160 SI TIP-NAVLU DE X6='TIME-CHARTER' ALORS
161 M Y18=Y18 / Y6
162 FIN
163 FAIRE
164 M X8=FORMAL ORDINE Y21 DE X7
165 M Y20=NUMAR-REALIZ DE X8
166 SI Y20=0 ALORS
167 M Y19=NUMBE X3
168 M NUMAR-REALIZ DE X8=Y19
169 M TOTAL-CH.DE X8=Y18
170 SORTIE
171 FIN
172 M Y25=TOTAL-CH DE X8
173 SI Y18 >= Y25 ALORS M Y21=Y21 + 1 REFAIRE FIN
174 M Y22=Y21 + 1
175 FAIRE
176 M X8=FORMAL ORDINE Y22 DE X7
177 M Y23=NUMAR-REALIZ DE X8
178 SI Y23=0 ALORS SORTIE FIN
179 M Y22=Y22 + 1
180 SI Y22 < 101 ALORS REFAIRE FIN
181 FIN
182 M Y24=Y22 - 1
183 FAIRE

```

```

184 M X8=FORMAL ORDINE Y24 DE X7
185 M Y17=NUMAR-REALIZ DE X8
186 M Y15=TOTAL-CH DE X8
187 M X8=FORMAL ORDINE Y22 DE X7
188 M NUMAR-REALIZ DE X8=Y17
189 M TOTAL-CH DE X8=Y15
190 SI Y24=Y21 ALORS SORTIE FIN
191 M Y24=Y24 - 1
192 M Y22=Y22 - 1 REFAIRE
193 FIN
194 M X8=FORMAL ORDINE Y21 DE X7
195 M Y19=NUMDE X3
196 M NUMAR-REALIZ DE X8 =Y19
197 M TOTAL-CH DE X8=Y18
198 FIN
199 FIN
200 M Y1=VALOARE-NAVLU-DOLARI DE X6 M Y4=Y1
201 M Y2=VALOARE-NAVLU-CENTI DE X6 M Y5=Y2 + 100
202 SI TIP-NAVLU DE X6='PE TONA' ALORS
203 M Y3=CANTITATE-TOT-TO DE X6
204 M Y4=Y2 * Y3 / 100
205 M Y5=Y2 * Y3 - Y4 * 100
206 SI Y5 > 49 ALORS M Y4=Y4 + 1 FIN
207 M Y5=Y1 * Y3 M Y4=Y4 + Y5 M Y5=100
208 SINON
209 SI TIP-NAVLU DE X6='PE MC' ALORS
210 M Y3=CANTITATE-TOT-MC DE X6
211 M Y4=Y2 * Y3 / 100
212 M Y5=Y2 * Y3 - Y4 * 100
213 SI Y5 > 49 ALORS M Y4=Y4 + 1 FIN M Y5=Y1 *
214 M Y4=Y4 + Y5 M Y5=100
215 FIN
216 FIN
217 I ' ' I ' ' I ' '
218 I (1 30) 'DATELE OFERTEI DE TRANSPORT' ECRIRE
219 I ' '
220 I (1 9) 'NAVLU TIP'
221 I (+1 13) TIP-NAVLU DE X6
222 I (+0 10) Y4
223 I (+0 1) ' '
224 I (+2 -2) Y5
225 I (+1 6) 'DOLARI'
226 SI TIP-NAVLU DE X6='TIME-CHARTER' ALORS
227 I (+1 5) 'PE ZI'
228 FIN
229 ECRIRE
230 I ' '
231 I (1 30) 'NAVELE CARE POT EFECTUA TRAN'
232 I (+0 30) 'SPORTUL' ECRIRE
233 I ' '
234 M Y1=1
235 FAIRE
236 M X8=FORMAL ORDINE Y1 DE X7
237 M Y2=NUMAR-REALIZ DE X8
238 SI Y2=0 OU Y1 > 100 ALORS SORTIE FIN
239 I (1 4) Y1
240 I (+0 1) ' '
241 M X3=UN L-NAVA Y2
242 M Z1=COD-NAVA DE X3
243 M X1=UN NAVA AVEC COD-NAVA=Z1 ;
244 I (6 15) DENUMIRE-NAVA DE X1

```

```

245 I (+1 9) 'LIBERA LA'
246 M Y3=ZI DE DATA-ELIBERARII DE X3
247 M Y3=Y3 + 100
248 M Y4=LUNA DE DATA-ELIBERARII DE X3
249 M Y4=Y4 + 100
250 I (+3 -2) Y3
251 I (+0 1) '/'
252 I (+2 -2) Y4
253 I (+0 1) '/'
254 I (+4 -4) AN DE DATA-ELIBERARII DE X3
255 I (+1 2) 'LA'
256 M Z1=COD-PORT DE X3
257 M X2=UN PORT AVEC COD-PORT=Z1 :
258 I (+1 25) DENUMIRE-PORT DE X2
259 ECRIRE
260 I (8 8) 'MILE GOL'
261 I (+0 10) DISTANTA-GOL DE X3
262 I (+1 9) 'MILE INC.'
263 I (+0 10) DISTANTA-INC DE X3
264 I (+1 9) 'ZILE MARS'
265 I (+0 10) DURATA-MARS DE X3
266 I (+1 10) 'ZILE STAT.'
267 I (+0 10) DURATA-STATIONARE DE X3
268 ECRIRE
269 M Y3=CHELTUIELI-OPERARE DE X3
270 M Y4=CHELTUIELI-TRECERI DE X3
271 M Y5=CHELTUIELI-PORTUARE DE X3
272 M Y6=CHELTUIELI-FIXE DE X3
273 M Y7=CHELTUIELI-INDEMNIZATII DE X3
274 M Y8=CONSUM-COMB-GREU-TO DE X3
275 M Y9=CONSUM-COMB-USOR-TO DE X3
276 M Y10=CONSUM-COMB-GREU-VAL DE X3
277 M Y11=CONSUM-COMB-USOR-VAL DE X3
278 SI CONDITII-INC-DESC DE X6='FIO' OU
279 CGNDITII-INC-DESC DE X6='FIOST' ALORS
280 M Y12=0
281 SINON
282 SI CONDITII-INC-DESC DE X6='TERMS' OU
283 CONDITII-INC-DESC DE X6='LINER-OUT' ALORS
284 M Y12=Y3
285 SINON
286 M Y12=Y3 / 2
287 FIN
288 FIN I (8 9) 'TOTAL CH.'
289 M Y13=Y12 + Y4 + Y5 +Y7 + Y10 +Y11 + Y6
290 SI TIP-NAVLU DE X6='TIME-CHARTER' ALORS
291 M Y14=DURATA-MARS DE X3
292 M Y15=DURATA-STATIONARE DE X3
293 M Y14=Y14 + Y15
294 M Y15=Y13 /Y14
295 M Y16=Y13 - Y15 * Y14
296 M Y16=Y16 * 100
297 M Y17=Y16 / Y14
298 M Y18=Y16 - Y17 * Y14
299 SI Y18 > 49 ALORS M Y17=Y17 + 1 FIN
300 SI Y17 > 99 ALORS M Y15=Y15 + 1 M Y17=0 FIN
301 I (+0 10) Y15
302 M Y17=Y17 + 100
303 I (+0 1) '.'
304 I (+2 -2) Y17
305 I (+1 4) '$/ZI'

```

```

306 SINON
307 I (+0 10) Y13
308 I (+1 1) '$'
309 FIN
310 SI Y8 ^= 0 ALORS
311 I(+1 3) 'CG='
312 I (+0 8) Y8
313 I (+1 2) 'TO'
314 I (+0 9) Y10
315 I (+1 1) '$'
316 FIN
317 SI Y9 ^= 0 ALORS
318 I (+1 3) 'CU='
319 I (+0 8) Y9
320 I (+1 2) 'TO'
321 I (+0 9) Y11
322 I (+1 1) '$'
323 FIN
324 ECRIRE
325 I (8 2) 'FX'
326 I (+0 9) Y6
327 I (+1 1) '$'
328 I (+1 3) 'IN'
329 I (+0 9) Y7
330 I (+1 1) '$'
331 I (+1 3) 'PO'
332 I (+0 10) Y5
333 I (+1 1) '$'
334 SI Y4 ^= 0 ALORS
335 I (+1 3) 'TR'
336 I (+0 9) Y4
337 I (+1 1) '$'
338 FIN
339 SI Y12 ^= 0 ALORS
340 I (+1 3) 'OP'
341 I (+0 9) Y12
342 I (+1 1) '$'
343 FIN ECRIRE
344 M Y1=Y1 + 1 REFAIRE
345 FIN
346 M DORITI-FISA-DE-ANTECALCUL DE X6=U
347 M DORITI-FISA-DE-ANTECALCUL DE X6=EXT
348 SI DORITI-FISA-DE-ANTECALCUL DE X6=U ALORS
349 M DORITI-FISA-DE-ANTECALCUL DE X6='NU'
350 FIN
351 SI DORITI-FISA-DE-ANTECALCUL DE X6= DA ALORS
352 FAIRE
353 M NUMARUL-NAVEI-ALESE DE X6=U
354 M NUMARUL-NAVEI-ALESE DE X6=EXT
355 M Y1=NUMARUL-NAVEI-ALESE DE X6
356 M X8=FORMAL ORDINE Y1 DE X7
357 M Y2=NUMAR-REALIZ DE X8
358 SI EXISTE UN L-NAVA X3 Y2 ALORS M Y1=0 SINON
359 I (1 30) 'NU EXISTA IN LISTA NAVELOR SEL'
360 I (+1 18) 'ECTATE NAVA CU NR.'
361 I (+0 4) Y1 ECRIRE REFAIRE
362 FIN
363 FIN M NUMARUL-NAVEI-ALESE DE X6=Y2
364 FIN
365 D X1 D X2 D X3 D X4 D X5 D X6 D X7 D X8 D X9
366

```

367 :FDEF

368 ?

PROGRAM <PREOF3>  
ERORI DETECTATE: 0  
PREOF3=PREOF3

INSERAT

## ÎNSERAREA MACROINSTRUCȚIUNILOR (Trasee, combustibil, nave disponibile)

```

1  :DEFMAC SCRIERE-TRASEU
2  :EXP
3  M Y1=1
4  POUR TOUT TRASEUL-NAVEI X2 DE X1
5  M Z2=COD-PORT DE X2
6  SI Z2=U ALORS M Z3='NEDEFINIT' SINON
7  M Z3=DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z2 ; DE X0
8  FIN
9  I (1 3) Y1
10 I (+0 1) ' '
11 I (+1 25) Z3
12 M Y2=DATA-INTRARE-IN-PORT DE X2
13 SI Y2 ^= U ALORS
14 M Y3=Y2 / 10000
15 M Y4=Y2 - Y3 * 10000
16 M Y5=Y4 / 100
17 M Y4=Y4 - Y5 * 100 M Y4=Y4 + 100 M Y5=Y5 + 100
18 I (+3 -2) Y4
19 I (+0 1) ' / '
20 I (+2 -2) Y5
21 I (+0 1) ' / '
22 I (+4 -4) Y3 SI Y1 > 1 ALORS
23 I (+1 19) 'DISTANTA PINA LA EL' FIN
24 SINON SI Y1 = 1 ALORS
25 I (+12 19) 'DISTANTA PINA LA EL' FIN
26 FIN
27 SI Y1 = 1 ALORS
28 M Y2=DISTANTA DE X2
29 I (+0 6) Y2 I (+1 4) 'MILE'
30 FIN
31 ECRIRE M Y2=)
32 POUR TOUT MARFA-OPERATA X3
33 I (3 3) Y2
34 I (+0 1) ' '
35 I (+1 4) TIP-OPERARE DE X3
36 M Z4=COD-MARFA DE X3
37 SI Z4=U ALORS M Z5='NEDEFINITA' SINON
38 M Z5=DENUMIRE-MARFA DE UNE GRUPE-MARFA AVEC
39 COD-GRUPE-MARFA=Z4 ; DE X0
40 FIN
41 I (+1 25) Z5
42 I (+0 7) CANTITATE-TONE DE X3
43 I (+1 4) 'TONE' ECRIRE
44 M Y2=Y2 + J
45 FIN
46 M Y1=Y1 + 1
47 FIN
48 :FDEF
49 ?
50 :DEFMAC RITMURI
51 :EXP
52 M Y7=0 M Y8=0
53 FAIRE
54 M Y7=Y7 + 1
55 SI Y7=2 ALORS
56 SI Y12=0 ALORS M Y8=Y8 + 29
57 SINON M Y8=Y8 + 28 FIN
58 SINON M Y9=ZIUA DE UNE ZILE-LUNI Y7 M Y8=Y8 + Y9 FIN
59 SI Y8 >= Y25 ALORS SORTIE FIN REFAIRE
60 FIN
61 M X2=FORMAL RITH-LUNA Y7 DE X1 M Y7=RITH DE X2

```

```

62 M Y7=Y7 + 1 M RITM DE X2=Y7
63 :FDEF ?
64 :DEFMAC SCADERE
65 :EXP
66   POUR TOUTE REALIZARI-1 X5
67     M Y1=NR-REALIZ-1 DE X5
68     POUR UN DISPONIBIL X4 Y1 DE X3
69       M Y2=REST-DISPONIBIL DE X4
70       M Y2=Y2 - 1
71       M REST-DISPONIBIL DE X4=Y2
72     FIN
73   FIN
74 :FDEF ?
75 :DEFMAC CALCUL-LIMITE
76 :EXP
77 M Y4=1 M Y5=0
78 FAIRE
79 SI Y4=Y11 ALORS
80   M Y5=Y5 + Y10
81   SORTIE
82   FIN
83 SI Y4=2 ALORS
84 SI Y12=0 ALORS
85   M Y5=Y5 + 29
86   SINON
87     M Y5=Y5 + 28
88   FIN
89 SINON
90   M Y6=ZIU DE UNE ZILE-LUNI Y4
91   M Y5=Y5 + Y6
92   FIN
93   M Y4=Y4 + 1
94   REFAIRE
95   FIN
96 :FDEF ?
97 :DEFMAC CUMUL
98 :EXP
99 POUR TOUTE ZILE X3 DE X2
100 M Y20=NUMDE X3
101 M Y15=DISPONIBILE DE X3 M Y16=ANGAJATE DE X3
102 SI Y13=0 ALORS
103   SI Y20 >= Y10 ET Y20 <= Y11 ALORS SUIVANT FIN
104   SINON
105   SI Y20 < Y10 ALORS SUIVANT FIN
106   SI Y20 > Y11 ALORS SORTIE FIN
107   FIN
108   SI Y12=0 ALORS
109     M Y15=Y15 + 1 M DISPONIBILE DE X3=Y15
110     SINON
111       M Y16=Y16 + 1
112       SI Y16 > Y15 ALORS
113         M Y21=1 M Y22=1 M Y23=ANUL DE X4
114         M Y5=Y20 - 1 M Y25=0
115         FAIRE
116         SI Y25=Y5 ALORS SORTIE FIN
117         M Y4=ZIU DE UNE ZILE-LUNI Y22 M Y25=Y25 + 1
118         SI Y22=2 ALORS
119           SI Y1=366 ALORS M Y4=29 SINON M Y4=28 FIN
120         FIN
121         SI Y21 < Y4 ALORS M Y21=Y21 + 1
122         SINON

```

```

123         M Y21=1
124         SI Y22 < 12 ALORS
125             M Y22=Y22 + 1
126             SINON
127                 M Y22=1 M Y23=Y23 + 1
128             FIN
129         FIN
130         REFAIRE
131         FIN
132         M Y21=Y21 + 100 M Y22=Y22 + 100
133         I (1) 'IN DATA ' I (+2 -2) Y21 I (+0) '/'
134         I (+2 -2) Y22 I (+0) '/' I (+4 -4) Y23
135         I (+1) 'TOATE CELE' I (+0) Y15 I (+1) 'NAVE SINT OCUPATE'
136         ECRIRE
137         SINON
138             M ANGAJATE DE X3=Y16
139         FIN
140         FIN
141         FIN
142         :FDEF ?
143         :DEFMAC CONSUM-COMBUSTIBIL-MARS
144         :EXP
145         M Y11=0
146         M Y12=0
147         M Y13=0
148         M Y14=0
149         M Y15=0
150         SI Y5=0 ALORS
151             SI PAS UN CONSUM-COMBUSTIBIL 1 DE X1 ALORS
152                 SI PAS UN CONSUM-COMBUSTIBIL 6 DE X1 ALORS
153                     M Y5=2
154                 SINON
155                     M Y5=11
156                 FIN
157             FIN
158         SINON
159             SI PAS UN CONSUM-COMBUSTIBIL 6 DE X1 ALORS
160                 SI PAS UN CONSUM-COMBUSTIBIL 1 DE X1 ALORS
161                     M Y5=2
162                 SINON
163                     M Y5=10
164                 FIN
165             FIN
166         FIN
167         M Y2=Y5 / 10
168         M Y2=Y5 - Y2 * 10
169         SI Y2 < 2 ALORS
170             POUR TOUT CONSUM-COMBUSTIBIL X9 DE X1
171                 M Y16=NUMDE X9
172                 SI Y2=0 ET Y16 > 5 ALORS SORTIE FIN
173                 SI Y2=1 ET Y16 < 6 ALORS SUIVANT FIN
174                 M Y11=VITEZA DE X9
175                 M Y17=CONSUM DE X9
176                 SI Y11 < Y4 ALORS
177                     M Y13=Y15
178                     M Y12=Y14
179                     M Y14=Y11
180                     M Y15=Y17
181                 SUIVANT
182             FIN
183             SI Y11 >= Y4 ALORS

```



```

184      M Y12=Y14
185      M Y13=Y15
186      SORTIE
187      FIN
188      FIN
189      SI Y11=0 ET Y12=0 ALORS M Y5=2 SINON
190      SI Y12=0 OU Y12=Y11 ALORS
191      M Y20=Y17 * Y4 / Y11
192      SINON
193      M Y20=(Y17 - Y13) * (Y4 - Y12) / (Y11 - Y12) + Y13
194      FIN
195      FIN
196      FIN
197      M Y24=0
198      M Y21=0
199      M Y22=0
200      M Y23=0
201      SI Y2=2 ALORS M Y2=2
202      SINON M Y25=0 M Y16=0
203      SI TIP-COMBUSTIBIL-MARS-MOT-AUX DE X1 ^= U ALORS
204      M Y25=CANTITATE-CONSUM-MARS-AUX DE X1
205      SI Y25=U ALORS
206      M Y25=CANTIT-CONSUM-1-ZI-STAT-KG DE X1
207      FIN
208      FIN
209      M Z6=TIP-COMBUSTIBIL-MARS-MOT-PR DE X1
210      M Y11=Z6
211      M Z7=TIP-COMBUSTIBIL-MARS-MOT-AUX DE X1
212      M Y12=Z7
213      M Y13=PRET-PER-TONA DE UN TIP-COMBUSTIBIL AVEC
214      COD-COMBUSTIBIL=Z6 ;
215      M Y14=CURS-LEU-DOLAR DE X6
216      SI Y5=0 OU Y5=11 ALORS M Y15=DISTANTA-GOL DE X3
217      SINON
218      M Y15=DISTANTA-INC DE X3
219      FIN
220      M Y20=Y20 / 10
221      M Y20=Y20 * Y15 / (Y4 * 100)
222      M Y25=Y25 / 10
223      M Y25=Y25 * Y15 / (Y4 * 100)
224      SI Z7 ^= U ALORS
225      M Y15=PRET-PER-TONA DE UN TIP-COMBUSTIBIL AVEC
226      COD-COMBUSTIBIL=Z7 ;
227      M Y16=Y25 * Y15 / Y14
228      FIN
229      M Y17=Y20 * Y13 / Y14
230      SI Y11 < 35 ALORS
231      M Y21=Y21 + Y20
232      M Y23=Y23 + Y17
233      SINON
234      M Y22=Y22 + Y20
235      M Y24=Y24 + Y17
236      FIN
237      SI Y12 >= 35 ALORS
238      M Y21=Y21 + Y25
239      M Y23=Y23 + Y16
240      SINON
241      M Y22=Y22 + Y25
242      M Y24=Y24 + Y16
243      FIN
244      FIN

```

```

SOCMAG      V3.1  27-AUG-87 12:29  PAGINA 5

245 :FDEF
246 ?
247 :DEFMAC CONSUM-COMBUSTIBIL-STATIONARE
248 :EXP
249 M Z6=TIP-COMBUSTIBIL-1-STATIONARE DE X1
250 M Z7=TIP-COMBUSTIBIL-2-STATIONARE DE X1
251 M Y11=Z6 M Y17=DURATA-STATIONARE DE X3
252 M Y12=Z7
253 M Y13=CANTIT-CONSUM-1-ZI-STAT-KG DE X1
254 M Y14=CANTIT-CONSUM-2-ZI-STAT-KG DE X1
255 M Y15=CURS-LEU-DOLAR DE X6
256 SI Z7 ^= U ALORS
257 M Y16=PRET-PER-TONA DE UN TIP-COMBUSTIBIL AVEC
258 COD-COMBUSTIBIL=Z7 ;
259 M Y24=DURATA-STAT-RADA DE X3
260 M Y23=Y17 - Y24
261 M Y25=Y23 * Y14 / 1000
262 M Y18=Y25 * Y16 / Y15
263 FIN
264 M Y16=PRET-PER-TONA DE UN TIP-COMBUSTIBIL AVEC
265 COD-COMBUSTIBIL=Z6 ;
266 M Y20=Y17 * Y13 / 1000
267 M Y19=Y20 * Y16 / Y15
268 M Y21=0 M Y22=0 M Y23=0 M Y24=0
269 SI Y11 < 35 ALORS
270 M Y21=Y21 + Y20
271 M Y23=Y23 + Y19
272 SINON
273 M Y22=Y22 + Y20
274 M Y24=Y24 + Y19
275 FIN
276 SI Y12 < 35 ALORS
277 M Y21=Y21 + Y25
278 M Y23=Y23 + Y18
279 SINON
280 M Y22=Y22 + Y25
281 M Y24=Y24 + Y18
282 FIN
283 :FDEF
284 ?
285 :DEFMAC CALCUL-CHELTUIELI-PORT
286 :EXP
287 M X1=UN NAVA AVEC COD-NAVA=Z4 ; DE X0
288 M Y11=CAPACITATE-TONE-CARGO-VARA DE X1
289 M Y23=Y11 / 100
290 M Y22=Y11 - Y23 * 100
291 SI Y22 > 49 ALORS M Y23=Y23 + 1 FIN
292 M Z6=GRUP-NAVE DE X1
293 SI Z6='PETROLIER' ALORS
294 M Z6='1'
295 SINON
296 SI Z6='MINERALIER' ALORS
297 M Z6='2'
298 SINON
299 M Z6='3'
300 FIN
301 FIN M Y25=0 M Y24=0
302 FAIRE
303 M X2=UN PORT AVEC COD-PORT=Z3 ; DE X0
304 M Y5=0
305 M Y6=0

```

```

306 M Y10=0
307 M Y11=0
308 M Y15=0
309 M Y16=0
310 POUR TOUTE CHELT-PORT-INC-DESC X9 DE X2
311 M Z7=TIP-NAVA DE X9 M Y4=TDW DE X9
312 SI Z6=Z7 ALORS
313     M Y20=Y4 / 100
314     M Y21=Y20 * 100
315     M Y13=Y4 / Y21
316     SI Y13 > 49 ALORS M Y20=Y20 + 1 FIN
317     M Y19=Y20
318     M Y20=AN-OPERARE-PORT DE X9
319     M Y21=LUNA-OPERARE-PORT DE X9
320     M Y13=CHELTUIELI-PORTUARE DE X9
321     M Y14=CHELTUIELI-INC-DESC DE X9
322     M Y13=Y13 * 1000 / Y19
323     M Y14=Y14 * 1000 / Y19
324     SI Y13 > Y5 ALORS
325         M Y5=Y13
326         M Y10=Y20
327         M Y11=Y21
328     FIN
329     SI Y14 > Y6 ALORS
330         M Y6=Y14
331         M Y15=Y20
332         M Y16=Y21
333     FIN
334     FIN
335     FIN
336     SI Y5=0 ET Z3 ^= '00000000' ALORS
337         SI Z5='PORT' ALORS
338             M Z3='00000000' REFAIRE
339         SINON
340             I (1 6) 'PENTRU'
341             I (+1 15) DENUMIRE-NAVA DE X1
342             I (+1 2) 'IN'
343             I (+1 30) DENUMIRE-PORT DE X2
344         ECRIRE
345             FAIRE
346                 M TAXE-TRECERE DE X6=U
347                 M TAXE-TRECERE DE X6=EXT
348                 SI TAXE-TRECERE DE X6=U OU TAXE-TRECERE DE X6=0
349                     ALORS I (1 15) 'EROARE REPETATI' ECRIRE REFAIRE
350                 FIN M Y5=TAXE-TRECERE DE X6
351                 M Y21=CURS-LEU-DOLAR DE X6,
352                 M Y23=100
353                 M Y5=Y5 * Y21
354             FIN
355         FIN
356     FIN
357     SI Y25=0 ALORS M Y25=Y5 FIN
358     SI Z5 ^= 'PORT' ET Z3 ^= '00000000' ALORS M Y24=0
359     SINON
360         SI Y6=0 ET Z3 ^= '00000000' ALORS
361             M Z3='00000000' REFAIRE
362         FIN
363         M Y24=Y6
364     FIN
365     FIN
366     M Y5 =CURS-LEU-DOLAR DE X6

```

```

367 M Y19=Y25 / Y5
368 M Y9 =Y24 / Y5
369 M Y25=Y8 / 100
370 M Y24=Y8 - Y25 * 100
371 SI Y24 > 49 ALORS M Y25=Y25 + 1 FIN
372 M Y21=Y9 * Y25 / 100
373 M Y22=Y19 * Y23 / 100
374 :FDEF
375 ?
376 :DEFMAC CALCUL-DIST-INIT
377 :EXP
378 M X2=UN PORT AVEC COD-PORT=Z1 ; DE X0
379 M Y11=LATITUDINEA-GRADE DE X2
380 M Y12=LATITUDINEA-MINUTE DE X2
381 M Z3=EMISFERA-LATITUDINII DE X2
382 M Y13=LONGITUDINEA-GRADE DE X2
383 M Y14=LONGITUDINEA-MINUTE DE X2
384 M Z4=EMISFERA-LONGITUDINII DE X2
385 M X2=UN PORT AVEC COD-PORT=Z2 ; DE X0
386 M Y15=LATITUDINEA-GRADE DE X2
387 M Y16=LATITUDINEA-MINUTE DE X2
388 M Z5=EMISFERA-LATITUDINII DE X2
389 M Z6=EMISFERA-LONGITUDINII DE X2
390 M Y17=LONGITUDINEA-GRADE DE X2
391 M Y18=LONGITUDINEA-MINUTE DE X2
392 SI Z3=Z5 ALORS
393 M Y19=Y11 - Y15
394 M Y20=Y12 - Y16
395 M Y21=Y19 * 60 + Y20
396 M Y12=Y21
397 SI Y21 < 0 ALORS M Y12=-1 M Y12=Y12 * Y21 FIN
398 SINON
399 M Y19=Y11 + Y15
400 M Y20=Y12 + Y16
401 M Y12=Y19 * 60 + Y20
402 FIN
403 SI Z4=Z6 ALORS
404 M Y19=Y13 - Y17
405 M Y20=Y14 - Y18
406 M Y21=Y19 * 60 + Y20
407 SINON
408 M Y19=Y13 + Y17
409 M Y20=Y14 + Y18
410 M Y21=Y19 * 60 + Y20
411 SI Y19 > 180 ALORS
412 M Y21=21600 - Y21
413 FIN
414 FIN
415 M Y13=Y21
416 SI Y21 < 0 ALORS M Y13=-1 M Y13=Y13 * Y21 FIN
417 M Y12=Y12 + Y13
418 :FDEF
419 ?
420 :DEFMAC CAUTA-TRASEU
421 :EXP
422 M Z6=' ' M Z7=' '
423 M Y23=0
424 POUR TOUT TRASEUL-NAVEI X3 DE X1
425 SI DATA-INTRARE-IN-PORT=U ALORS
426 SI Y23=0 ALORS
427 M Z6=COD-PORT DE X3 M Y21=NUMDE X3

```

```

428         M Y23=1
429     FIN
430 SINON
431     M Y23=0
432     M Z7=COD-PORT DE X3 M Y22=NUMDE X3
433     FIN
434     FIN
435     SI Y23=0 ALORS M Z6=' ' FIN
436     SI Y8=0 ALORS
437         SI Z3 ^= Z6 ALORS M Y8=1 FIN
438     SINON M Y21=Y22
439         SI Z3 ^= Z7 ALORS M Y8=1 FIN
440     FIN
441     SI Y8 ^= 1 ALORS
442         M X3=UN TRASEUL-NAVEI Y21 DE X1
443     FIN
444     :FDEF
445         ?
446     :DEFMAC CALCUL-STATIONARE
447     :EXP
448     M Z5=GRUP-NAVE DE X1
449     M Z6=' '
450     FAIRE
451     M X2=UN PORT AVEC COD-PORT=Z3 ;
452     SI Z5='PETROLIER' ALORS
453         M Y4=ZILE-ASTEPT-RADA-PETR DE X2
454         M Y5=NR-NAVE-ZILE-ASTEPT-RADA-PETR DE X2
455         M Y6=ZILE-OPERARE-PETR DE X2
456         M Y7=TONE-ZILE-OPERARE-PETR DE X2
457     SINON
458         SI Z5='MINERALIER' ALORS
459             M Y4=ZILE-ASTEPT-RADA-MIN DE X2
460             M Y5=NR-NAVE-ZILE-ASTEPT-RADA-MIN DE X2
461             M Y6=ZILE-OPERARE-MIN DE X2
462             M Y7=TONE-ZILE-OPERARE-MIN DE X2
463     SINON
464         M Y4=ZILE-ASTEPT-RADA-CARG DE X2
465         M Y5=NR-NAVE-ZILE-ASTEPT-RADA-CARG DE X2
466         SI Z4='USOR' ALORS
467             M Y6=ZILE-OPERARE-CARG-MFU DE X2
468             M Y7=TONE-ZILE-OPERARE-CARG-MFU DE X2
469     SINON
470         SI Z4='MEDIU' ALORS
471             M Y6=ZILE-OPERARE-CARG-MFM DE X2
472             M Y7=TONE-ZILE-OPERARE-CARG-MFM DE X2
473         SINON
474             M Y6=ZILE-OPERARE-CARG-MFG DE X2
475             M Y7=TONE-ZILE-OPERARE-CARG-MFG DE X2
476         FIN
477     FIN
478     FIN
479     FIN
480     SI Z3 ^= '00000000' ALORS
481         SI TIP-PORT DE X2= 'CANAL' OU
482             TIP-PORT DE X2= 'STRIMTOARE' ALORS M Z6='
483         SI Y5=0 ALORS
484             I (1 7) 'PENTRU'
485             I (+1 25) DENUMIRE-PORT DE X2 ECRIRE
486             M DURATA-ZILE-TRAVERSARE DE X6=U
487             M DURATA-ZILE-TRAVERSARE DE X6=EXT
488             SI DURATA-ZILE-TRAVERSARE DE X6=U ALORS

```

```

489         M Y4 =0
490     SINON
491         M Y4 =DURATA-ZILE-TRAVERSARE DE 'X6
492     FIN
493     SI NR-NAVE-ZILE-ASTEPT-RADA-PETR DE X2=0 ALORS
494         M NR-NAVE-ZILE-ASTEPT-RADA-PETR DE X2=1
495         M ZILE-ASTEPT-RADA-PETR DE X2=Y4
496     FIN
497     SI NR-NAVE-ZILE-ASTEPT-RADA-MIN DE X2=0 ALORS
498         M NR-NAVE-ZILE-ASTEPT-RADA-MIN DE X2=1
499         M ZILE-ASTEPT-RADA-MIN DE X2=Y4
500     FIN
501     SI NR-NAVE-ZILE-ASTEPT-RADA-CARG DE X2=0 ALORS
502         M NR-NAVE-ZILE-ASTEPT-RADA-CARG DE X2=1
503         M ZILE-ASTEPT-RADA-CARG DE X2=Y4
504     FIN
505         M Y5=1
506     FIN
507         M Y6=0
508     SINON
509         SI Y7=1 OU Y5=0 ALORS
510             M Z3='00000000' REFAIRE
511         FIN
512     FIN
513     FIN
514     SI Y5=0 ALORS M Y5=1 FIN
515     M Y21=Y4 * 100 / Y5
516     M Y23=Y6 * Y8 / Y7
517     M Y24=Y6 * Y8 - Y23 * Y7
518     M Y24=Y24 * 100 / Y7
519     M Y23=Y23 * 100 + Y24
520     M Y23=Y23 + Y21
521     M Y18=Y23 / 100
522     M Y24=Y23 - Y18 * 100
523     SI Y24 > 49 ALORS M Y18=Y18 + 1 FIN
524     M Y23=Y21 / 100
525     M Y24=Y21 - Y23 * 100
526     M Y21=Y23
527     SI Y24 > 49 ALORS M Y21=Y21 + 1 FIN
528     SI Z6=' ' ET Y18=0 ALORS M Y18=1 M Y21=1
529     SINON
530         SI Y18=Y21 ET Y8 ^= 0 ET Y18 ^= 0 ALORS
531             M Y18=Y18 + 1
532     FIN
533     FIN
534     FIN
535     :FDEF
536     ?
537     :DEFMAC CALCUL-DATA
538     :EXP
539     M Y21=Y11
540     M Y22=Y12
541     M Y23=Y13
542     M Y25=0
543     SI Y5=U ALORS M Y5=0 FIN
544     FAIRE
545     SI Y25=Y5 ALORS SORTIE FIN M Y4=ZIU A DE UN ZILE-LUNI Y22
546         M Y25=Y25 + 1
547     SI Y22=2 ALORS
548         M Y24=Y23 / 4
549         M Y24=Y23 - Y24 * 4

```

```

550   SI Y24=0 ALORS
551     M Y4=29
552   SINON
553     M Y4=28
554   FIN
555   FIN
556   SI Y21 < Y4 ALORS
557     M Y21=Y21 + 1
558   SINON
559     M Y21=1
560   SI Y22 < 12 ALORS
561     M Y22=Y22 + 1
562   SINON
563     M Y22=1
564     M Y23=Y23 + 1
565   FIN
566   FIN
567   REFAIRE
568   FIN
569   M Y11=Y21
570   M Y12=Y22
571   M Y13=Y23
572   :FDEF
573   ?
574   :DEFMAC CALCUL-ZONA-VEC
575   :EXP
576   M Y1=LATITUDINEA-GRADE DE X2
577   M Y2=LATITUDINEA-MINUTE DE X2
578   SI EMISFERA-LATITUDINII DE X2='N' ALORS
579     M Y1=Y1 + 90
580   SINON
581     M Y1=90 - Y1
582     SI Y2 ^= 0 ALORS M Y2=60 - Y2 M Y1=Y1 - 1 FIN
583   FIN
584   M Y3=Y1 * 60 + Y2
585   M Y4=VEGINATATE-NORD-GRADE DE X2
586   M Y2=VEGINATATE-NORD-MINUTE DE X2
587   M Y6=Y4 * 60 + Y5
588   M Y5=Y3 + Y6
589   M Y21=Y5 / 60
590   M Y22=Y5 - Y21 * 60
591   M Y5=Y21 * 100 + Y22
592   SI Y5 > 18000 ALORS M Y5=18000 FIN
593   M Y4=VEGINATATE-NORD DE X6=Y5
594   M Y4=VEGINATATE-SUD-GRADE DE X2
595   M Y5=VEGINATATE-SUD-MINUTE DE X2
596   M Y6=Y4 * 60 + Y5
597   M Y5=Y3 - Y6
598   M Y21=Y5 / 60
599   M Y22=Y5 - Y21 * 60
600   M Y5=Y21 * 100 + Y22
601   SI Y5 < 0 ALORS M Y5=0 FIN
602   M Y4=VEGINATATE-SUD DE X6=Y5
603   M Y1=LONGITUDINEA-GRADE DE X2
604   M Y2=LONGITUDINEA-MINUTE DE X2
605   SI EMISFERA-LONGITUDINII DE X2='W' ALORS
606     M Y1=360 - Y4
607     SI Y2 ^= 0 ALORS M Y2=60 - Y2 M Y1=Y1 - 1 FIN
608   FIN
609   M Y3=Y1 * 60 + Y2
610   M Y4=VEGINATATE-VEST-GRADE DE X2

```

SOCMAG V3.1 27-AUG-87 12:29 PAGINA 11

```

611 M Y5=VEGINATATE-VEST-MINUTE DE X2
612 M Y6=Y4 * 60 + Y5
613 M Y5=Y3 - Y6
614 SI Y5 < 0 ALORS M Y5=21600 + Y5
615 M Y7=Y5 / 60
616 M Y8=Y5 - Y7 * 60
617 M Y5=Y7 * 60 + Y8
618 FIN
619 M Y21=Y5 / 60
620 M Y22=Y5 - Y21 * 60
621 M Y5=Y21 * 100 + Y22
622 M LONGITUDINE-VEGINATATE-VEST DE X6=Y5
623 M Y4=VEGINATATE-EST-GRADE DE X2
624 M Y5=VEGINATATE-EST-MINUTE DE X2
625 M Y6=Y4 * 60 + Y5
626 M Y5=Y3 + Y6
627 SI Y5 > 21600 ALORS M Y5=Y5 - 21600 FIN
628 M Y21=Y5 / 60
629 M Y22=Y5 - Y21 * 60
630 M Y5=Y21 * 100 + Y22
631 M LONGITUDINE-VEGINATATE-EST DE X6=Y5
632 :FDEF
633 ?
634 :DEFMAC CALCUL-PERIOADA
635 :EXP
636 FAIRE
637 M Y21=Y11 + Y12 * 100 + Y13 * 10000
638 M Y22=Y14 + Y15 * 100 + Y16 * 10000
639 SI Y21 > Y22 ALORS SORTIE FIN
640 M Y21=Y12 + 1 M Y4=0
641 M Y22=Y15
642 SI Y13 < Y16 ALORS M Y22=Y22 + 12 FIN
643 M Y5=Y13 / 4
644 M Y6=Y13 - Y5 * 4
645 SI Y6=0 ALORS M Y24=29 SINON M Y24=28 FIN
646 M Y5=Y16 / 4
647 M Y6=Y16 - Y5 * 4
648 SI Y6=0 ALORS M Y25=29 SINON M Y25=28 FIN
649 FAIRE
650 SI Y21=U OU Y22=U ALORS SORTIE FIN
651 SI Y21 >= Y22 ALORS SORTIE FIN
652 SI Y21 > 12 ALORS M Y23=Y21 - 12 M Y4=1 SINON
653 M Y23=Y21
654 FIN
655 SI Y23 ^= 2 ALORS
656 M Y5=ZIUUA DE UN ZILE-LUNI Y23
657 M Y17=Y17 + Y5
658 SINON
659 SI Y4=0 ALORS M Y17=Y17 + Y24 SINON
660 M Y17=Y17 + Y25 FIN
661 FIN
662 M Y21=Y21 + 1 REFAIRE
663 FIN
664 SI Y12=Y15 ET Y13=Y16 ALORS M Y17=Y14 - Y11 SINON
665 SI Y12=2 ALORS M Y22=Y24 SINON
666 M Y22=ZIUUA DE UN ZILE-LUNI Y12
667 FIN
668 M Y17=Y17 + Y22 - Y11 + Y14
669 FIN
670 FIN
671 :FDEF

```



```

672
673 :DEFMAC DETERMINARE-DISPONIBIL
674 :EXP
675 FAIRE
676 SI NUMAR-VOIAJ DE X1=991 OU NUMAR-VOIAJ DE X1=U ALORS
677 I (5 15) DENUMIRE-NAVA DE X1
678 I (+1 30) 'IESITA DIN EXPLOATARE' ECRIRE
679 I ' ' SORTIE
680 FIN
681 M Z7=COD-PORT-ACTUAL DE X1
682 M Y5=0
683 SI Z7 ^= 100000000 ALORS
684 POUR TOUT INEL-ACT-OPERATIVA X8 DE X1
685 M Y15=D TOUT SUBACTIVITATE DE X8
686 FAIRE
687 SI Y15=0 ALORS SORTIE FIN
688 M X9=UN SUBACTIVITATE Y15 DE X8
689 M Z2=COD-PORT DE X9
690 M Z3=COD-SUBACTIVITATE DE X9
691 M Y29=2
692 SI TIP-ACT-NAVA DE X1 < Y29 ALORS
693 SI Z3='90007' OU Z3='90008' OU Z3='90009'
694 OU Z3='90010' OU Z3='90011' OU Z3='90012'
695 OU Z3='90013' ALORS M Y5=2 SORTIE
696 FIN
697 FIN
698 SI Z2=Z7 ALORS
699 M Y11=ZI DE DATA-ACTIVITATII DE X8
700 M Y12=LUNA DE DATA-ACTIVITATII DE X8
701 M Y13=AN DE DATA-ACTIVITATII DE X8
702 SI Z3='90007' OU Z3='90008' OU Z3='90009'
703 OU Z3='90010' OU Z3='90011' OU Z3='90012'
704 OU Z3='90013' ALORS
705 M Y25=Y11 + Y12 * 100 + Y13 * 10000
706 FIN
707 SINON M Y5=1 SORTIE
708 FIN
709 M Y15=Y15 - 1 REFAIRE
710 FIN
711 SI Y5 > 0 ALORS SORTIE FIN
712 FIN
713 FIN
714 D X8 D X9
715 M TIP-STAT-PORT DE X6=Y5
716 SI TIP-ACT-NAVA DE X1 > 1 ALORS
717 M Y4=DURATA-PLAN-REPARATIE DE X1
718 M Y5=DURATA-PLAN-PREG-SANTIER DE X1
719 M Y5=Y5 + Y4
720 M Y13=Y25 / 10000
721 M Y4 =Y25 - Y13 * 10000
722 M Y12=Y4 / 100
723 M Y11=Y4 - Y12 * 100
724 M DATA-REPARATIE DE X6=Y25
725 CALCUL-DATA
726 M Y25=Y11 + Y12 * 100 + Y13 * 10000
727 FIN
728 M Y29=2
729 SI TIP-ACT-NAVA DE X1 ^= U ET TIP-ACT-NAVA DE X1 < Y29 ALORS
730 M Y14=ZI DE ZIUA-CURENTA DE X6
731 M Y15=LUNA DE ZIUA-CURENTA DE X6
732 M Y16=AN DE ZIUA-CURENTA DE X6 M Y17=0

```

```

733   CALCUL-PERIOADA
734   FIN
735   M Y19=0
736   SI TIP-ACT-NAVA DE X1 > 1 ALORS
737     M Y14=Y11
738     M Y15=Y12
739     M Y16=Y13
740     M Y11=ZI DE ZIUA-CURENTA DE X6
741     M Y12=LUNA DE ZIUA-CURENTA DE X6
742     M Y13=AN DE ZIUA-CURENTA DE X6 M Y17=(
743     CALCUL-PERIOADA
744     M Y19=Y17
745   FIN
746   SI COD-PORT-ACTUAL DE X1 ^= '00000000' E1
747     TIP-ACT-NAVA DE X1 ^= 0 ALORS M Y8=2
748     M Z3=COD-PORT-ACTUAL DE X1 CAUTA-TRASEU
749     SI Y8=1 ALORS
750       I (5 15) DENUMIRE-NAVA DE X1
751       I (+1 20) 'NU ARE PORTUL ACTUAL
752       M Z4=DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z3 ;
753       I (+1 20) Z4
754       I (+1 20) 'IN TRASEU ' ECRIRE
755       I ' '
756     SORTIE
757   FIN
758   M Y8=0 M Y20=NUMDE X3 M Y20=Y20 + 1
759   POUR TOUT MARFA-OPERATA X8 DE X3
760     M Y10=CANTITATE-TONE DE X8
761     M Y8=Y8 + Y10
762     M Z5=COD-MARFA DE X8
763   FIN
764   M Z4=U
765   M Z4=TIP-MARFA DE UN GRUPA-MARFA AVEC COD-GRUPA-MARFA=Z5
766     DE X0
767   SI Z4=U ALORS M Z4='MEDIU' FIN M Y18=0
768   CALCUL-STATIONARE
769   M Y29=2
770   SI TIP-ACT-NAVA DE X1 < Y29 ALORS
771     SI Y17 < Y18 ALORS
772       M Y19=Y18 - Y17
773     FIN
774   SINON
775     M Y19=Y19 + Y18
776   FIN
777   SINON
778     M Z3=COD-PORT-DESTINATIE DE X1
779     SI Z3=U ALORS
780       I (5 15) DENUMIRE-NAVA DE X1
781       I (+1 30) 'PORT DESTINATIE NEDEFINIT'
782     ECRIRE
783     I ' ' SORTIE
784   FIN
785   M Y8=0
786   CAUTA-TRASEU
787   SI Y8=1 ALORS
788     I (5 15) DENUMIRE-NAVA DE X1
789     I (+1 21) 'NU ARE PORTUL DESTIN.'
790     M Z4=DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z3 ;
791     DE X0
792     I (+1 20) Z4
793     I (+1 20) 'IN TRASEU ' ECRIRE

```

```

794      I      SORTIE
795      FIN
796      M Y10=DISTANTA-PINA  *LESTINATIE DE X3
797      M Y4=VITEZA-IMPUSA DE X1
798      M Y6=Y10 / Y4
799      M Y7=Y10 - Y6 * Y4
800      M Y5=Y4 / 2
801      SI Y7 >= Y5 ALORS M Y6=Y6 + 1 FIN
802      M Y19=Y6
803      M Y20=NUMDE X3
804      FIN
805      M Y8=0
806      M Y15=Y20 - 1
807      /* IN LOC DE DEPUIS Y15 */
808      POUR TOUT TRASEUL-NAVEI X3 DE X1
809      M Y29=NUMDE X3
810      SI Y29 < Y15 ALORS SUIVANT FIN
811      M Y20=NUMDE X3
812      SI Y8=0 ET COD-PORT-ACTUAL DE X1 ^= '00000000' ET
813      TIP-ACT-NAVA DE X1 ^= U ALORS M Y8=1
814      FIN
815      SI Y8=1 ALORS
816      M Y10=DISTANTA DE X3
817      M Y4=VITEZA-IMPUSA DE X1
818      M Y6=Y10 / Y4
819      M Y7=Y10 - Y6 * Y4
820      M Y4=Y4 / 2
821      SI Y7 >= Y4 ALORS M Y6=Y6 + 1 FIN
822      M Y19=Y19 + Y6
823      FIN
824      M Y8=0
825      M Y18=0
826      POUR TOUT MARFA-OPERATA X8 DE X3
827      M Y10=CANTITATE-TONE DE X8
828      M Y8=Y8 + Y10
829      M Z5=COD-MARFA DE X8
830      FIN
831      M Z4=U
832      M Z4=TIP-MARFA DE UN GRUPA-MARFA AVEC COD-GRUPA-MARFA=Z5
833      DE X0
834      SI Z4=U ALORS M Z4='MEDIU' FIN
835      M Z3=COD-PORT DE X3
836      CALCUL-STATIONARE
837      M Y8=1
838      M Y19=Y19 + Y18
839      FIN
840      SI Y8=0 ALORS M Y20=Y15 FIN
841      M Y11=ZI DE ZIUA-CURENTA DE X6
842      M Y12=LUNA DE ZIUA-CURENTA DE X6
843      M Y13=AN DE ZIUA-CURENTA DE X6
844      M Y5=Y19 + 1
845      CALCUL-DATA
846      M Z2=COD-PORT DE UN TRASEUL-NAVEI Y20 DE X1
847      M X2=UN PORT AVEC COD-PORT=Z2 ; DE X0
848      I (1 15) DENUMIRE-NAVA DE X1
849      I (+1 12) 'EXP.ACT.GAR.'
850      M Y9=ZI DE DATA-EXPIRARE-ACTE-REPARATIE DE X1
851      M Y9=Y9 + 100
852      I (+3 -2) Y9
853      M Y9=LUNA DE DATA-EXPIRARE-ACTE-REPARATIE DE X1
854      M Y9=Y9 + 100

```

```

855 I (+0 1) '/'
856 I (+2 -2) Y9
857 I (+0 1) '/'
858 I (+4 -4) AN DE DATA-EXPIRARE-ACTE-REPARATIE DE X1
859 I (+1 15) 'PLANIFICARE REP'
860 M Y9=ZI DE DATA-INCEPERE-REPARATIE-PLAN DE X1
861 M Y9=Y9 + 100
862 I (+3 -2) Y9
863 M Y9=LUNA DE DATA-INCEPERE-REPARATIE-PLAN DE X1
864 I (+0 1) '/'
865 I (+2 -2) Y9
866 I (+0 1) '/'
867 I (+4 -4) AN DE DATA-INCEPERE-REPARATIE-PLAN DE X1
868 ECRIRE
869 I (3 24) 'ESTE DISPONIBILA LA DATA'
870 M Y11=Y11 + 100
871 M Y12=Y12 + 100
872 I (+3 -2) Y11
873 I (+0 1) '/'
874 I (+2 -2) Y12
875 I (+0 1) '/'
876 I (+4 -4) Y13
877 I (+1 2) 'IN'
878 I (+1 25) DENUMIRE-PORT DE X2
879 ECRIRE
880 SI TIP-ACT-NAVA DE X1 > 1 ALORS
881 I (3 30) 'SE AFLA IN REPARATIE DE L'
882 I (+0 6) 'A DATA'
883 M Y25=DATA-REPARATIE DE X6
884 M Y21=Y25 / 10000
885 M Y24=Y25 - Y21 * 10000
886 M Y22=Y24 / 100
887 M Y23=Y24 - Y22 * 100
888 M Y23=Y23 + 100
889 M Y22=Y22 + 100
890 I (+3 -2) Y23
891 I (+0 1) '/'
892 I (+2 -2) Y22
893 I (+0 1) '/'
894 I (+4 -4) Y21
895 I (+1 2) 'IN'
896 M Z6=COD-PORT-ACTUAL DE X1
897 I (+1 25) DENUMIRE-PORT DE UN PORT AVEC COD-PORT=Z6 ;
898 DE X0
899 ECRIRE
900 FIN
901 I ' '
902 FIN
903 :FDEF
904 .

```

MACRO <SCRIERE-TRASEU>	INSERAT
MACRO <RITMURI>	INSERAT
MACRO <SCADERE>	INSERAT
MACRO <CALCUL-LIMITE>	INSERAT
MACRO <CUMUL>	INSERAT
MACRO <CONSUM-COMBUSTIBIL-MARS>	INSERAT
MACRO <CONSUM-COMBUSTIBIL-STATIONARE>	INSERAT
MACRO <CALCUL-CHELTUIELI-PORT>	INSERAT
MACRO <CALCUL-DIST-INIT>	INSERAT
MACRO <CAUTA-TRASEU>	INSERAT
MACRO <CALCUL-STATIONARE>	INSERAT
MACRO <CALCUL-DATA>	INSERAT
MACRO <CALCUL-ZONA-VEC>	INSERAT
MACRO <CALCUL-PERIOADA>	INSERAT
MACRO <DETERMINARE-DISPONIBIL>	INSERAT

ERORI DETECTATE: 0  
MACRO.LST=INTERM.SOC

## Alegerea navei. Economia voiajului

DATELE OFERTEI DE TRANSPORT

NAVLU TIP PE TONA 260000 DOLARI

NAVELE CARE POT EFECTUA TRANSPORTUL

1. DUNAREA LIBERA LA 03/04/1987 LA CONSTANTZA  
MILE GOL 0 MILE INC. 3000 ZILE MARS 9 ZILE STAT. 26  
TOTAL CH. 156280 \$ CG= 243 TO 27405 \$ CU= 76 TO 12032 \$  
FX 96250 \$ IN 3675 \$ PO 16918 \$
2. PRUTUL LIBERA LA 04/04/1987 LA CONSTANTZA  
MILE GOL 0 MILE INC. 3000 ZILE MARS 9 ZILE STAT. 26  
TOTAL CH. 156280 \$ CG= 243 TO 27405 \$ CU= 76 TO 12032 \$  
FX 96250 \$ IN 3675 \$ PO 16918 \$
3. MURESUL LIBERA LA 04/04/1987 LA CONSTANTZA  
MILE GOL 0 MILE INC. 3000 ZILE MARS 8 ZILE STAT. 27  
TOTAL CH. 217572 \$ CG= 378 TO 42630 \$ CU= 38 TO 6016 \$  
FX 116655 \$ IN 4725 \$ PO 47546 \$
4. OLTUL LIBERA 05/05/1987 LA CONSTANTZA  
MILE GOL 0 MILE INC. 3000 ZILE MARS 8 ZILE STAT. 27  
TOTAL CH. 233722 \$ CG= 378 TO 42630 \$ CU= 140 TO 22166 \$  
FX 116655 \$ IN 4725 \$ PO 47546 \$
5. IZA LIBERA LA 10/04/1987 LA GALATZ  
MILE GOL 130 MILE INC. 3000 ZILE MARS 9 ZILE STAT. 32  
TOTAL CH. 235117 \$ CG= 388 TO 43757 \$ CU= 147 TO 23274 \$  
FX 115005 \$ IN 5535 \$ PO 47546 \$

CALCUL ECONOMIC VOIAJ

\*\*\*\*\*

NAVA ALEASA : MURESUL

A. CONDITIILE OFERTEI :

NAVLU: PE TONA 20.00 \$  
CONDITII-INC-DESC = FIO

TRASEUL OFERTEI

CONSTANTZA NORMA DE OPERARE 900 TO/ZI  
INC MARFURI GENERALE DIVERSE 13000 TO

ANTWERP NORMA DE OPERARE 1100 TO/ZI  
DESC MARFURI GENERALE DIVERSE 13000 TO

VITEZA NAVEI 384 MILE PE ZI

B. DURATA VOIAJ :

CONSTANTZA NUMAR ZILE STATIONARE 14  
CONSTANTZA - ANTWERP 3000 MILE 8.0 ZILE  
ANTWERP NUMAR ZILE STATIONARE 13  
TOTAL ZILE VOIAJ 35

## C. NECESARUL DE COMBUSTIBIL :

1. ZILE MARS :
  - A). INCARCAT : 3000 MILE 8 ZILE
  - B). BALAST : 0 MILE 0 ZILE
  - C). TOTAL : 3000 MILE 8 ZILE
  - D). PROCENT MARS IN BALAST/TOTAL VOIAJ : 0 %
2. ZILE STATIONARE : 27
3. CAPACITATE TANCURI COMBUSTIBIL (TONE) : 1500
4. CONSUM COMBUSTIBIL : USOR 38 TONE GREU 378 TONE
5. PROCURARE COMBUSTIBIL :
  - A). DIN TARA : USOR 38 TONE GREU 378 TONE
  - B). DIN STRAINATATE :

## D. CHELTUIELI :

1. CHELTUIELI IN LEI CONVERTITI LA 18 LEI/\$
  - A). CHELTUIELI FIXE : 35 ZILE = 116655 \$
  - B). CHELTUIELI COMBUSTIBIL : 416 T0 = 43646 \$
  - C). TOTAL CHELTUIELI (A+B) : 165301 \$
2. CHELTUIELI EFECTIVE IN VALUTA :
  - A). CHELTUIELI PORTUARE : 2 PORTURI = 47546 \$
  - B). TAXE CANALE,STRIMTORI : 0 TRECERI = 0 \$
  - C). CHELTUIELI MANIPULARE : 13000 T0 = 0 \$
  - D). COMBUSTIBIL : 0 TONE = 0 \$
  - E). ALOC.HRANA + IND.STRAINATATE : 35 ZILE = 4725 \$
  - F). TOTAL CHELTUIELI VALUTARE : 52271 \$
3. TOTAL CHELTUIELI : 217572
4. PROCENT CHELTUIELI EFECTIVE IN VALUTA / VENIT BRUT = 20.14 %
5. PROCENT CHELTUIELI EFECTIVE IN VALUTA / TOTAL CHELTUIELI = 24.02 %

E. NAVLU VALUTAR BRUT : 260000 \$

F. NAVLU VALUTAR BRUT PE ZI EXPLOATARE : 7428.58 \$

G. VENIT NET : 42428 \$

H. VENIT NET PE ZI EXPLOATARE : 1212.23 \$

ANEXĂ

## Noutățile versiunii 4 a produsului SOCRATE-MINI

În volumul 2 al acestei cărți este descrisă versiunea 3 a produsului SOCRATE-MINI. În intervalul scurs între predarea manuscrisului la tipografie și apariția cărții a fost elaborată versiunea 4 a produsului, care conține modificări importante. Prezentăm în continuare noutățile introduse prin lansarea noii versiuni.

### 1. Verificarea și refacerea coerenței fizice și logice a bazei de date

#### Extensii ale utilitarului STS

Pentru fiecare lanț de squatterii rupt sau eronat sînt afișate informații privitoare la fișierul, blocul și subpaginile afectate, precum și adresa virtuală a subpaginii.

Pentru necorelările dintre dicționarul de pagini pline ale fiecărui domeniu și antetele de pagină se comunică numărul fișierului, blocului, paginii virtuale și subpaginii afectate.

Aceste informații, afișate în cazul unei incoerențe fizice a bazei de date, pot fi utilizate după refacerea coerenței fizice cu utilitarul VRU, pentru a constata care dintre caracteristicile bazei de date au fost afectate datorită refacerii.

A fost introdus un nou comutator /LN:n prin care se solicită afișarea lanțurilor de squat-teri avînd în componență n subpagini sau mai multe.

Aceste informații permit analiza modului în care s-a încărcat baza de date, funcție de descrierea structurii și de dimensiunea spațiului real.

## Utilitarul VRU

### Introducere în VRU

Programul VRU (VeRify Utility) este un program utilitar al S.G.B.D. SOCRATE-MINI ce are ca scop verificarea coerenței logice a unei baze de date.

Se oferă în plus și un set de funcțiuni prin care se poate încerca refacerea coerenței logice și fizice a bazei de date.

Funcțiunile utilitarului sînt :

- funcțiuni de verificare a coerenței logice a bazei de date care asigură :
  - verificarea corelației între lanțul de biți și conținutul realizărilor unei entități ;
  - verificarea corelației între lanțul de biți al unei caracteristici de tip invers și lanțul de biți al entității inversate ;
  - verificarea caracteristicilor de tip inel ;
  - verificarea caracteristicilor de tip referință simplă ;
  - verificarea caracteristicilor declarate cheie ;
- funcțiuni de refacere a coerenței logice a bazei de date care asigură :
  - refacerea corelației între lanțul de biți al unei entități și realizările acelei entități ;
  - refacerea corelației între lanțul de biți al unei caracteristici de tip invers și lanțul de biți al entității inversate ;
  - refacerea unei caracteristici de tip inel ;
  - refacerea unei caracteristici de tip referință simplă ;
  - refacerea lanțului de squat-teri pentru un domeniu dat ;
- funcțiuni de ștergere care asigură :
  - ștergerea unei caracteristici declarate cheie ;
  - ștergerea tuturor numelor de programe precompilate și de macro-instrucțiuni din dicționar ;
  - ștergerea unei caracteristici de tip inel ;
- funcțiunea de afișare care asigură :
  - afișarea conținutului unui domeniu dat de la o adresă virtuală și pe o lungime dată ;

Aceste funcțiuni se pot apela de utilizator sub forma unor programe scrise într-un limbaj a cărui sintaxă este dată în continuare.

Lansarea utilitarului se poate face doar dacă sesiunea a fost deschisă în prealabil.

În timpul lucrului cu utilitarul, baza este blocată.

Deoarece funcțiunile de verificare și refacere inel cu calificare TOUT fac apel la utilitarul de sortare al sistemului de operare, este necesar ca acesta să fie instalat înainte de începerea lucrului.

### Linia de comandă

Linia de comandă pentru lucrul cu VRU are următorul format :

>VRU fișier-ieșire/SW=fișier-intrare

— fișier-intrare conține comenzile cu funcțiunile ce trebuie executate ; tipul implicit este .VRU. ;

— fișier-intrare nu poate fi un fișier de comenzi indirecte ;

— fișier-ieșire este fișierul pe care sînt listate comenzile, mesajele afișate de funcțiile solicitate și eventualele mesaje de eroare ; tipul implicit este .LST. ;

— SW este comutatorul LP descris mai jos :

Comutator	Fișa la care se atașează	Poate fi negat	Valoarea implicită	Funcție
LP	ieșire	NU	—	inhibă tipărirea mesajelor de control ce încep cu '— —'

### Sintaxa comenzilor VRU

Apelul funcțiilor utilitarului se face prin programe scrise într-un limbaj asemănător cu limbajul SOCRATE, a cărui gramatică se prezintă în continuare :

<funcțiune>	:: = <verificare>	:: = %R <id-referință>
	:: = <refacere>	<calificare>
	:: = <ștergere>	:: = %R \$LBT <id-entitate>[<calificare>]
	:: = <afișare>	:: = %R \$DPG <nume-domeniu>
<verificare>	:: = %V <id-entitate>	:: = %R \$\$QT <nume-domeniu>
	[<calificare>]	<ștergere>
	:: = %V <id-invers>	:: = %S <id-caracteristică-cheie><calificare-TOUT>
	[<calificare>]	:: = %S <id-inel>
	:: = %V <id-inel>	<calificare-TOUT>
	[[<nr-iteratii>]<calificare>]	:: = %S \$PRG <nume-domeniu>
	:: = %V <id-referință>	<adresă-virtuală>
	<calificare>	<lungime>
	:: = %V <id-caracteristică-cheie><calificare>	<calificare>
	:: = %V \$LBT <id-entitate>[<calificare>]	:: = <calific-UN>
<refacere>	:: = %R <id-entitate>	:: = <calific-TOUT>
	[<calificare>]	:: = DE UN <id-1><nr-1>
	:: = %R <id-invers>	[DE UN <id-2><nr-2>... DE UN <id-4><nr-4>]
	[<calificare>]	<calific-TOUT>
	:: = %R <id-inel>	:: = DE TOUT <id-1>
	[[<nr-iteratii>]	[DE TOUT <id-2>... DE TOUT <id-4>]
	<calificare>	

### Comentarii

- semnele <, >, [, ] fac parte din meta-limbajul de descriere a gramatcii.
- identificatorii (id-entitate, id-invers, id-inel, id-referință, id-caracteristică-cheie, id-1,..., id-4) sînt succesiuni de 1-30 caractere (litere sau cifre) primul fiind obligatoriu literă.
- nr-iteratii este o valoare numerică cuprinsă între 0—2<sup>31</sup>—1; el apare doar la operațiile de verificare și refacere pentru caracteristici de tip inel; valoarea sa implicită este 2000; el semnifică numărul maxim de realizări presupuse a fi legate pe inelul respectiv și are ca scop evitarea buclor în cazul parcurgerii unor inele cu legăturile distruse.
- calificare-UN selectează o singură realizare asupra căreia se aplică funcțiunea; selecția se realizează prin numerele de realizare (nr-1,..., nr-4).
- calificare-TOUT selectează toate realizările entității respective.
- nume-domeniu este format dintr-o succesiune de 3 caractere (litere sau cifre) din care prima este obligatoriu literă și reprezintă numele domeniului așa cum a fost dat la generarea cu GFM.
- adresa-virtuală și lungime apar doar în funcția de afișare și semnifică adresa virtuală de la care se începe afișarea și lungimea (0—65535) pe care se face.
- \$PRG este cuvînt cheie și indică aplicarea funcțiilor de ștergere pe zona dicționar corespunzătoare numelor de macroui și programe precompilate.
- \$LBT este cuvînt cheie și indică aplicarea funcțiilor de verificare și refacere doar pe lanțul de biți corespunzător unei entități.
- \$DPG și \$\$QT sînt cuvinte chele ce apar doar la funcția de refacere și indică corectarea Dicționarului de PaGlni pline, respectiv a lanțului de SQuaTteri pentru un domeniu dat.
- sfîrșitul secvenței de funcțiuni ce se dorește executat este indicat prin ? sau CTRL-Z.
- la execuție, începutul fiecărei funcțiuni este anunțat de un mesaj ce începe cu două asteriscuri și conține tipul funcțiunii ce se execută, tipul și numele caracteristicii asupra căreia se exercită funcțiunea.



## Funcțiuni de verificare a coerenței logice a bazei de date

### Verificarea entităților

Această funcție realizează verificarea contorului de realizări al unei entități cu numărul de biți avind valoarea „1” din șirul de biți de prezență afectat entității și permite, opțional, verificarea conținutului realizărilor entității.

Sintaxă :

%V [\$LBT] <id-entiate> [<calificare>]

unde :

- \$LBT, opțional, precizează efectuarea numai a verificării contorului de realizări. Absența argumentului \$LBT determină, în plus, și verificarea conținutului unei realizări.
- <calificare> desemnează entitățile de nivel superior în care este inclusă entitatea ce se dorește a fi verificată.

Acțiune :

Verificarea contorului de realizări al unei entități se face prin parcurgerea biților din lanțul de biți al entității. Biții care au valoarea binară „1” sînt numărați și în final rezultatul este comparat cu contorul de realizări.

Verificarea conținutului realizărilor unei entități se face tot prin parcurgerea lanțului de biți de prezență. Pentru fiecare bit care are valoarea binară „0”, ceea ce presupune o realizare negenerată, se testează dacă conținutul corespunzător din bază este vid.

### Verificarea caracteristicilor de tip invers

Sintaxă :

%V <id-invers> [<calificare>]

Acțiune :

Această funcție verifică o caracteristică de tip invers, controlînd dacă biții care au valoarea binară egală cu „1” din lanțul de biți ai inversel au bitul corespondent în lanțul de biți al entității inversate de asemenea egal cu „1”.

În finalul verificării unui invers se mai testează și dacă contorul biților egal cu „1” al inversei este același cu valoarea numărată a lor în lanțul de biți afectat caracteristicilor de tip invers.

### Verificarea caracteristicilor de tip inel

Au fost prevăzute două posibilități de verificare pentru inele :

- a. verificarea unui inel dintr-o anumită realizare de entitate ;
- b. verificarea unui inel din toate realizările unei entități.

a. Verificarea unui inel dintr-o anumită realizare de entitate

Sintaxă :

%V id-anneau [<nr. max>] <calificator>

unde :

- <nr. max> este opțional și semnifică numărul maxim de realizări din inel ; valoarea sa implicită este 2000 ; el este utilizat pentru ieșirea forțată dintr-o buclă infinită, cauzată de poziționarea eronată a unei adrese din listă, chiar pe un element anterior din inel.
- <calificator> desemnează entitățile de nivel superior care conțin inelul, împreună cu numerele de realizare :  
DE UN id-ent <nr> [DE UN id-ent <nr>...]

Pentru inel simplu, cit și pentru inel dublu, verificarea se face în două faze :

- faza 1 : se verifică referințele selectate prin parcurgerea inelului ;
  - faza 2 : se selectează referințele prin parcurgerea tuturor realizărilor generate ale entității care conține referința.
- b. Verificare inel din toate realizările unei entități

Sintaxă :

%V id-anneau [<nr-max>] DE TOUT id-ent [DE TOUT id-ent ...]

Se ia capul de lanț din fiecare realizare și l se fac verificările de înlănțuire. Se face apoi o trecere prin toate realizările entității ce conține referința, obținîndu-se un fișier INELI.REF. Acesta se sortează după capul de lanț, obținîndu-se un fișier INELO.REF.

Fișierul sortat INELO.REF conține referințele grupate pe inele și constituie baza de plecare pentru faza a doua din cadrul verificării.

— Se contorizează totodată numărul total de referințe în fișierul INELO.REF, precum și numărul de referințe din lanțurile inelelor. Acești doi contori trebuie să coincidă.

— La sfârșit se șterg toate fișierele SY:INELI.REF; \*,INELO.REF;\*

*Observație importantă :*

Sortarea se face cu utilitarul SRT al sistemului; SRT-ul trebuie să fie instalat înainte de apelarea funcției de verificare a unui inel din toate realizările unei entități.

**Verificarea caracteristicilor de tip referință simplă**

*Sintaxă :*

%V <id-ref-simplă> <calificare>

*Acțiune :*

Această funcție verifică, numai pentru referințele simple, dacă legătura către realizarea din entitatea referită este corectă. Se consideră că o legătură este incorectă dacă :

- a — nu punctează pe început de realizare, sau numărul maxim de realizări declarat în structură, sau
- b — numărul de realizare pe care punctează este mai mare decât
- c — realizarea referită nu este generată.

**Verificarea caracteristicilor de tip cheie**

*Sintaxă :*

%V <id-cheie> de tout <Id-entitate>

unde :

- <id-cheie> este identificatorul cheii ce se dorește verificată ;
- <id-entitate> este numele entității care conține cheia.

*Acțiune :*

Această funcție verifică dacă cheilor înregistrate în spațiul dicționar DIC le corespund valori conforme în bază, ceea ce presupune că realizarea care conține cheia să fie generată și în bază caracteristică declarată cheie să aibă o valoare identică cu cea din dicționar. Cele două verificări se realizează prin parcurgerea arborelui corespunzător din dicționar, exact ca printr-o funcție SOCRATE „PAR”.

**Funcțiuni de refacere a coerenței logice a bazei de date**

**Refacere entități**

*Sintaxă :*

%R [\$LBT] <id-entitate> [<calificare>]

unde :

- \$LBT, <id-entitate>, <calificare> au aceeași semnificație ca pentru comanda corespunzătoare %V.

*Acțiune :*

Se parcurge mai întâi etapa de verificare, identică cu cea descrisă pentru comanda %V, furnizându-se mesaje de eroare corespunzătoare.

Neconcordanța între contorul de realizări și valoarea numărată a biților egali cu „1” din lanțul de biți de prezență se refacă prin înlocuirea contorului cu valoarea numărată.

Pentru o realizare de entitate negenerată, dar având informații scrise în bază, se realizează ștergerea acestor informații, considerate ca fiind eronate.

**Refacere caracteristică de tip invers**

*Sintaxă :*

%R <id-Invers> [<calificare>]

unde :

- <id-invers>, <calificare> au fost descrise pentru comanda %V.

*Acțiune :*

Se execută mai întâi etapa de verificare descrisă pentru comanda corespunzătoare %V, furnizându-se mesaje de eroare dacă este cazul.

Biții care au valoarea „1” în lanțul de biți ai inversei și care nu au bitul corespunzător tot „1” în lanțul de biți al entității capătă valoarea „0”.

Neconcordanța între contorul inversei și valoarea numărată a biților egali cu „1” din lanțul de biți se refacă prin înlocuirea contorului cu valoarea numărată.

**Refacerea caracteristicilor de tip Inel**

Ca și la verificare s-au prevăzut două posibilități pentru refacere :

a. refacere inel specificat prin numerele de realizare ale „taților” ;

b. refacere inel din toate realizările unei entități.

a. *Refacerea unui inel determinat prin numerele de realizare ale „taților”*

Sintaxă :

**%R id-anneau [<nr. max>] DE UN id-ent nr [ DE UN id-ent nr...]**

Acțiune :

a1. *Inel simplu*

- se parcurge mai întâi etapa de verificare identică cu cea de la comanda %V.
- dacă în faza 1 se găsesc referințe ce conțin cimpul cap de lanț eronat, se reface acest cimp cu valoarea corectă.
- Dacă inelul este eronat, se trece la refacerea lui : concret, se constituie un inel din realizările care conțin referințe ce au cimpul cap de lanț poziționat pe capul de lanț al inelului specificat.

— ordinea de adăugare în inel este cea crescătoare a numerelor de realizare, deci posibil să fie altă ordine decît cea inițială.

— se recomandă interogarea prin limbajul de cereri SOCRATE a inelului refăcut și văzut exact ce s-a recuperat din vechiul inel.

a2. *Inel dublu*

- Se parcurge mai întâi etapa de verificare, identică cu cea de la comanda %V.
- În caz că inelul este corect din punct de vedere al validării efectuate, nu se mai reface inelul.
- În etapa de verificare se parcurg cele două lanțuri de pointeri înainte și înapoi.

Dacă se întilnește vreo referință cu adresa cap de lanț eronată, dar care să nu puncteze vreun alt cap de lanț, se reface cimpul cap de lanț din referință cu adresa corectă.

b. *Refacerea unui inel din toate realizările unei entități*

Sintaxă :

**%R id-anneau [<nr-max>] DE TOUT id-ent [ DE TOUT id-ent ...]**

Acțiune :

- Se efectuează mai întâi o trecere prin fiecare inel din toate realizările pentru depistarea și eventual refacerea adresei cap de lanț a fiecărei referințe din inel.
- Se extrag apoi într-un fișier INEL1.REF toate referințele asociate inelului.
- În continuare, se sortează acest fișier, prin lansarea utilitarului SRT, rezultînd fișierul INEL0.REF. Sortarea se face după valoarea cimpului cap de lanț.
- Se parcurge apoi etapa de verificare a

fiecărui inel din toate realizările generate, iar în caz de incoerență, se trece la refacerea lui. Tehnica de refacere diferă după cum inelul este simplu sau dublu și este asemănătoare cu cea de la inelul cu cap de lanț determinat.

— Se contorizează totodată numărul total de referințe în fișierul INEL1.REF precum și numărul de referințe din lanțurile inelelor. La sfîrșit se șterg toate fișierele SY : INEL1.REF ; \*, INEL0.REF \*

*Observație importantă :*

Ca și la funcția de verificare inel din toate realizările unei entități, și la refacere trebuie instalat în prealabil utilitarul STR de sortare al sistemului.

**Refacerea caracteristicilor de tip referință simplă**

Sintaxă :

**%R <id-ref-simplă> <calificare>**

unde :

— <id-ref-simplă>, <calificare> au fost descrise pentru comanda corespunzătoare %V.

Acțiune :

Se parcurge mai întâi etapa de verificare descrisă pentru comanda %V, furnizîndu-se mesajele de eroare corespunzătoare.

Referințele considerate că punctează eronat sînt șterse.

## Funcțiuni de refacere a coerenței fizice a bazei de date

Refacerea corelațiilor dintre dicționarul de pagini pline și antetul paginilor domeniului

Sintaxă :

**%R \$DPG ddd**

unde : ddd este numele domeniului unde se dorește refacerea corelațiilor.

Datorită unor incidente hard/soft pe parcursul exploatării bazei de date pot apare la un moment dat necorelații între dicționarul de pagini pline, aflat în primele blocuri ale primului fișier al fiecărui domeniu, și antetul unor pagini componente ale domeniului respectiv. Astfel poate apărea situația ca în dicționarul de pagini pline, o anumită pagină să fie considerată ocupată complet, cu toate că din antetul ei să rezulte că mai sînt subpagini libere sau viceversa, o pagină a căruia antet arată că are toate subpaginile ocupate să fie considerată pagină cu subpagini libere datorită informațiilor din dicționarul de pagini pline.

Prin intermediul acestei funcțiuni se restabilește corelația dintre dicționarul de pagini pline și antetul paginilor din domeniu, luînd de bază informațiile din antetul fiecărei pagini de date a domeniului respectiv.

În general, această incoerență se recuperează fără urmări grave pentru baza de date, dar trebuie totuși subliniat faptul că ea a apărut datorită pierderii, în timpul incidentului, a unor informații din baza care pot avea o importanță mult mai mare decît cele recuperate prin această funcțiune.

Pentru a verifica consistența bazei astfel recuperate, este necesară efectuarea unor teste de coerență logică, atît prin funcțiunile oferite de utilitar cît și prin programe utilizator, dacă datorită naturii aplicației, verificările de coerență logică oferite de utilitar sînt insuficiente.

Refacerea lanțurilor de squatteri

Sintaxă :

**%R \$\$SQT ddd**

unde : ddd — numele domeniului pentru care se solicită refacerea.

Din cauza unor incidente hard/soft, pot apare pe parcursul exploatării unei baze de date, lanțuri de squatteri rupte sau incorect înlănțuite (eronate).

Lanțurile de squatteri sînt de fapt subpagini de date din cadrul bazei care au intrat în coliziune datorită algoritmului de hash-cod al sistemului de gestiune a bazei de date.

Funcțiunea își propune repararea acelor lanțuri de squatteri care pot fi reparate și ștergerea subpaginilor care, datorită lipsei de informații corecte, nu pot fi atașate la un lanț de squatteri corect.

Repararea acestor incoerențe permite ca baza să poată fi utilizată în continuare, dar în nici un caz nu trebuie trasă concluzia că prin dispariția incoerențelor fizice, baza este consistentă.

Pentru verificarea consistenței bazei, odată eliminate incoerențele fizice, este necesar să se efectueze teste de verificare a coerenței logice a bazei, atît cu ajutorul funcțiilor oferite de utilitar, cît și cu programe utilizator, dacă prin natura sa baza impune verificări ce nu au fost prevăzute în utilitarul VRU.

## Funcțiuni de ștergere

Ștergerea caracteristicilor de tip inel

Pentru funcția de ștergere a inelelor s-a prevăzut doar ștergerea integrală a unui inel din toate realizările unei entități.

Sintaxă :

**%S id-anneau DE TOUT id-ent [ DE TOUT id-ent... ]**

Acțiune :

Se pune pe zero capul de lanț din toate realizările entității ce conține inelul și de asemenea se pune pe zero referința din toate realizările entității ce conține referința.

Ștergerea caracteristicilor de tip cheie

Sintaxă :

**%S <id-cheie> DE TOUT <id-entitate>**

unde : <id-cheie> și <id-entitate> sînt descrise în paragraful Verificarea caracteristicilor de tip cheie.

*Acțiune :*

Această funcție are ca efect ștergerea necondiționată a zonei dicționar (DIC) ce conține arborele corespunzător cheii cu numele furnizat în <id-cheie>.

**Ștergerea zonei DIC corespunzătoare numelor de macroinstrucțiuni și programe precompilate**

*Sintaxă :*

`%S $PRG`

*Acțiune :*

Această funcție are ca efect ștergerea din dicționar, domeniul DIC, a tuturor numelor de macroinstrucțiuni și programe precompilate înregistrate. Ștergerea este însoțită și de actualizarea unor informații, aflate în domeniul PRG, referitoare la numărul de macroinstrucțiuni și de programe precompilate.

### Afișarea informațiilor dintr-un domeniu

*Sintaxă :*

`%I ddd vvv III`

unde :

- ddd este numele domeniului pentru care se dorește afișarea; — III este lungimea zonei; nu poate depăși valoarea 65 535.
- vvv este adresa virtuală de început a zonei ce conține informațiile de afișaj;

*Acțiune :*

Informațiile dorite sînt afișate pe fișierul de ieșire sub forma de dump, în format octal pe byte, plus o decodificare alfanumerică în dreapta paginii. Liniile au lungimea de 132 octeți.

Dacă domeniul dorit este un domeniu de date (ex. FIS), dump-ul este precedat de o listă a caracteristicilor care corespund zonei respective, conform structurii.

Pentru toate domeniile (mai puțin PRG) se verifică dacă adresa virtuală „vvv” aparține spațiului virtual al domeniului respectiv, conform structurii.

## 2. Gestuirea formatelor de ecran

Pentru a permite construirea facilă a programelor care utilizează formate de ecran, la interogarea sau actualizarea bazei de date, au fost concepute două noi componente ale S.G.B.D. SOCRATE-MINI.

### DSF – utilitar pentru definirea formatelor de ecran

#### Introducere

DSF este un program utilitar cu ajutorul căruia se definesc formate de ecran ale utilizatorului.

Funcțiile DSF sînt :

- crearea sau modificarea unui fișier ce conține descrierea unui model de cîmpuri de date și de șiruri de caractere comentariu pe display ce se constituie într-un format de date.
- asocierea, opțională, cu cîmpurile de date a unor programe precompilate,
- catalogarea în baza de date, care operează cu datele introduse în cîmpuri.
- asocierea cu cîmpurile de date a unor atribute legate de alinierea datelor introduse în cîmpuri și de facilități de introducere.

Înainte de apelul DSF sesiunea de lucru SOCRATE nu este necesar să fie deschisă.

## Comutatorii

Comutatorii DSF și funcțiunile lor sint prezentate în tabelul de mai jos :

Comutator	Fișierul la care se atașează	Poate fi negat	Valoare implicită	Funcțiune
/DF	ieșire	NU	—	Indică cerere de definire a unui format de date nou.
/EX	ieșire	NU	—	Indică cerere de vizua- a unui format de date existent.
/TER:VT100 VT52	ieșire	NU	VT100	Indică tipul terminalului pe care se face definirea formatului.
/SF:car	ieșire	NU	]	Specifică caracterul care va fi considerat terminatorul cimpului de date. SF : spațiu are ca efect lipsa terminatorului, caz în care nu se pot face definiții de cimpuri de date.

## Linia de comandă

>DSF nume-format/SW

unde :

— nume-format este numele formatului de date ce se dorește creat sau numele unui format existent.

— SW sint comutatori care pot lua una din valorile : DF, EX, TER, SF. Combinația simultană a comutatorilor DF și EX nu este permisă.

## Definirea formatului de date

Definirea unui format de date se face prin simpla aranjare pe ecran a cimpurilor sale și a șirurilor de caractere comentariu de către utilizator. Acesta desenează formatul pe ecran, după care programul reține structura de cimpuri, atributele acestora și de asemenea șirurile de caractere comentariu.

Atributele unui cimp de date sint :

- 1) — Poziția cimpului (linia și coloana);
- 2) — Lungimea cimpului considerată ca fiind cuprinsă între caracterele „:” și „]”;
- 3) — Numele cimpului
- 4) — Numele unui program precompilat aflat în baza de date asociat cu cimpul de date;
- 5) — Aliniere; după introducerea datei în cimp se face alinierea informațiilor introduse la dreapta sau stînga;
- 6) — Salt automat; la introducerea de date, valoarea aflată în cimp se consideră ca fiind introdusă anterior (duplicare).

Notă

Introducerea datelor în cimpurile de date se face sub controlul programului utilitar DRW.

Utilizatorul își plasează cimpurile de date și caracterele comentariu în poziția dorită pe ecran cu ajutorul următoarelor taste :

„↑” (săgeată în sus)    deplasare cu o linie în sus  
 „↓” (săgeată în jos)    deplasare cu o linie în jos  
 „→” (săgeată la dr.)    deplasare cu o coloană la dreapta  
 „←” (săgeată la st.)    deplasare cu o coloană la stînga  
 (CR)    deplasare în prima coloană a liniei curente  
 CTRL/H    deplasare în linia 1 și coloana 1 a ecranului (HOME)

Începutul unui cimp de date este specificat prin plasarea caracterului „:”, iar sfîrșitul prin caracterul „]” (sau prin caracterul terminator specificat cu ajutorul comutatorului SF), lungimea cimpului fiind considerată întotdeauna cuprinsă între „:” și terminator (implicit „]”). Cimpurile descrise fără început sau sfîrșit vor fi considerate incorecte și nu vor face parte din

formatul de date, afișându-se în acest sens mesaje de eroare corespunzătoare. Un cîmp de date nu se poate întinde pe două linii.

Numele cîmpului precede caracterul „:”, poate avea orice lungime și nu trebuie să conțină spații.

Atributele 1, 2 și 3 sînt reținute automat de către program din modul în care au fost aranjate cîmpurile pe ecran.

Atributele 4, 5 și 6 se pot specifica, opțional, tastînd CTRL/E în cîmpul pentru care se dorește precizarea acestor atribute. În urma tastării CTRL/E sînt puse, în ordine, următoarele întrebări :

— EXEC nume-precompilat

Dacă anterior a fost precizat un program precompilat, se afișează numele acestuia și se așteaptă fie modificarea numelui programului, fie ștergerea asocierii prin tastarea unui spațiu în prima poziție, fie păstrarea vechii informații prin tastarea (CR) în prima poziție. Terminarea tastării numelui precompilatului asociat se face prin apăsare (CR), sau prin completarea a 20 de caractere diferite de spațiu. (Lungimea maximă a numelui precompilatului este de 20 de caractere).

— ALINIERE(D,S)

Se tastează D dacă, la momentul introducerii de date, alinierea informației în cîmp se va face la dreapta, sau S dacă alinierea se va face la stînga.

— SALT AUTOMAT(D,N)

Se tastează D dacă, la momentul introducerii de date, cîmpul respectiv va păstra informația introdusă anterior, altfel se tastează N.

*Observație*

Cererea CTRL/E făcută atunci cînd cursorul ecranului se află în linia 1 și coloana 1 are ca efect asocierea unui program precompilat cu întregul format de date. Acest program are acces la toate cîmpurile de date introduse.

Nu există limitări în legătură cu numărul de tentative de specificare a atributelor 4, 5 și 6 pentru un cîmp de date, sau pentru formatul de date.

### Deplasare de cîmpuri de date

Un cîmp sau mai multe cîmpuri de date deja aranjate pe o linie oarecare a ecranului poate fi mutat într-o altă poziție pe aceeași linie sau pe o altă linie cu ajutorul tastelor funcționale : CTRL/A, CTRL/B și CTRL/C. Porțiunea ce se dorește mutată se încadrează între CTRL/A (poziția de început) și CTRL/B (poziția de sfîrșit). Se deplasează cursorul cu ajutorul tastelor menționate anterior în poziția dorită și se tastează CTRL/C. Ca efect, porțiunea specificată se va muta în poziția curentă a cursorului.

### Șiruri de caractere comentariu

Orice caracter ASCII, în afara terminatorului unui cîmp de date, poate face parte dintr-un șir de caractere deștris în afara cîmpurilor de date. Numărul de caractere al tuturor șirurilor comentariu din formatul de date este limitat la 1 800 de caractere. Caractere comentariu identice cu terminatorul cîmpului de date vor determina mesaje de eroare.

### Refacerea imaginii ecranului

În orice moment, cu excepția celui al precizării atributelor unui cîmp de date, poate fi invocată refacerea imaginii ecranului cu ajutorul CTRL/R. Pe ecran va apare, ca urmare a tastării CTRL/R, imaginea distrusă prin deprogramarea terminalului și cursorul va fi poziționat în coloana 1 și linia 1.

### Terminarea sesiunii de lucru

Terminarea sesiunii de definire a formatului de date se face prin tastarea CTRL/Z. Atributele cîmpurilor detectate automat și cele specificate direct sînt reținute în fișierul precizat în linia de comandă DSF. Dacă în linia de comandă a fost folosit comutatorul EX, se creează

o nouă versiune a fișierului fără ștergerea versiunii anterioare. Un format de date definit trebuie să aibă asociat cel puțin un program precompilat, în caz contrar, programul afișează un mesaj de eroare și nu permite terminarea sesiunii de lucru.

### Formate de date definite anterior

Vizualizarea și eventual modificarea unui format de date deja definit se poate face invocând comutatorul EX. Structura de câmpuri de date a formatului împreună cu șirurile comentariu este afișată pe ecran, după care sînt permise orice modificări ale atributelor câmpurilor, ștergerea sau adăugarea de noi câmpuri sau a șirurilor comentariu în conformitate cu cele menționate anterior.

## DRW – program de interfață cu o bază de date prin intermediul formatelor de ecran

### Introducere

DRW este un program utilitar cu ajutorul căruia se permite introducerea de date în câmpurile unui format de date predefinit cu ajutorul programului DSF, precum și apelul de programe precompilate aflate în bază, care vor opera cu datele introduse.

Funcțiunile DRW sînt :

- afișarea pe ecranul terminalului a câmpurilor de date și a șirurilor de caractere comentariu dintr-un format de date și acceptarea introducerii de date în câmpuri.
- apelul de programe precompilate din baza de date SOCRATE asociate cu anumite câmpuri sau cu întreg formatul de date.
- facilități de duplicare automată a anumitor câmpuri și de aliniere în câmp.
- înlănuțirea formatelor de date.

Înainte de apelul DRW sesiunea de lucru SOCRATE trebuie deschisă și biblioteca SOCLIB instalată.

### Comutatori

Lista comutatorilor este prezentată în tabelul de mai jos :

Comutator	Fișierul la care se atașează	Poate fi negat	Valoare implicită	Funcțiune
/EN	ieșire	NU	—	Permite precizarea modului de lucru cu formate imbricate.
/PRO:nome	ieșire	NU	—	Permite precizarea numelui procesorului de interfață, instalat în prealabil.
/TER:VT100 VT52	ieșire	NU	VT100	Indică tipul terminalului pe care se face afișarea formatului.
/SF:car	ieșire	NU	]	Specifică caracterul care va fi considerat terminatorul câmpului de date. SF:spațiu are ca efect afișarea fără terminator.



**Linia de comandă**

&gt;DRW nume-format/SW

unde :

- nume-format este numele formatului de date ce se dorește apelat pentru afișare pe ecranul terminalului.
- SW sint comutatori care pot lua una din valorile : SF, EN, TER, PRO.

**Introducerea de date in cimpuri**

După afișarea pe ecran a cimpurilor de date și a șirurilor de caractere comentariu, cursorul se poziționează pe rind la începutul fiecărui cimp al formatului și așteaptă introducerea datelor. Terminarea introducerii de informații într-un cimp se face prin tastarea (CR) sau prin umplerea acestuia, după care cursorul ecranului sare automat la începutul cimpului următor. Dacă un cimp avea asociat un program precompilat, înainte de poziționarea în cimpul următor se execută precompilatul.

Tastarea „↓” (săgeată în jos) are ca efect saltul necondiționat în următorul cimp al formatului (dacă există), cu abandonarea cimpului curent.

Tastarea „↑” (săgeată în sus) are ca efect saltul necondiționat în cimpul anterior al formatului (dacă există), cu abandonarea cimpului curent.

Tastarea „←” (săgeată la stînga) are ca efect mutarea cursorului ecranului cu o coloană la stînga, în limita lungimii cimpului.

Tastarea „→” (săgeată la dreapta) are ca efect mutarea cursorului ecranului cu o poziție la dreapta.

**Apelul de programe precompilate din bază**

Orice cimp de date poate avea asociat un program precompilat care va fi executat după terminarea introducerii de date în cimpul respectiv. Programul precompilat are acces la datele introduse în toate cimpurile care preced cimpul cu care este asociat, inclusiv la acesta.

Accesul la datele introduse în cimpuri se face prin atașarea la buffer-ul transmis programului a unui formal care conține o descriere de caracteristici identică cu structura de cimpuri de date din formatul de date.

Formalul atașat buffer-ului de comunicare cu utilitarul DRW (care conform convenției este BUFFER 10), trebuie să aibă obligatoriu prevăzute în plus față de descrierea cimpurilor formatului de date, la început, încă două caracteristici și anume :

```
nume-1 BINAIRE 2
nume-2 MOT 6
```

Caracteristica nume-1 este necesară pentru comunicarea erorilor de validare a cimpurilor. Precompilatul trebuie să marcheze o eroare de validare, într-un cimp de date sau în mai multe cimpuri, prin introducerea unei valori diferite de zero în această caracteristică și prin punerea pe nedefinit a caracteristicii(lor) din formal corespunzătoare cimpului(lor) eronat(e) din formatul de date. Ca efect, programul DRW va decide re-poziționarea în cimpurile eronate pentru retastarea informației corecte. Această re-poziționare este însoțită de acționarea „BELL” și de scrierea de caractere „?” în tot cimpul eronat.

Programul precompilat asociat cu întreg formatul de date are acces la toate cimpurile acestuia, în rest, se respectă cele menționate mai sus. Programul precompilat asociat cu întreg formatul de date se execută după terminarea tastării informației în ultimul cimp al formatului.

Caracteristica nume-2 este necesară pentru a preciza numele (max. 6 caractere) unui format de date care va fi imediat afișat pe ecran. La întâlnirea unei valori nenule în caracteristica nume-2 programul DRW decide abandonarea imediată a formatului curent și încărcarea formatului de date cerut prin programul precompilat.

**Înlănțuiri de formate de date**

Există două moduri de a realiza înlănțuiri de formate de date.

De la terminal, prin apăsarea tastei CTRL/A pe ultima linie a ecranului apare întrebarea:

```
FORMAT
```

Ca răspuns se introduce numele unui format de date care va fi imediat afișat pe ecran. Comutatorul EN nu are efect pentru formatele afișate în modul descris mai sus.

Prin program, cu ajutorul cimpului descris în capitoul anterior, se poate de asemenea preciza numele formatului de date următor în modul descris.

Dacă în linia de comandă DRW a fost specificat comutatorul EN, atunci este permis apelul de formate de date imbricate. În această situație, la tastarea CTRL/Z, va fi afișat pe ecran formatul de date anterior (dacă există). Apelul de formate imbricate nu conservă informațiile introduse în cimpurile de date din momentul schimbării de format.

### Facilități de duplicare automată

Tastarea în orice moment a CTRL/B are ca efect trecerea în regim de duplicare automată a tuturor cimpurilor formatului. În acest mod de lucru tastarea (CR) într-un cimp are ca efect conservarea valorii introduse anterior în acel cimp.

Tastarea CTRL/D, în contextul tastării anterioare a CTRL/B, are ca efect intrarea în regim de salt automat (duplicare automată). În acest mod de lucru în cimpurile de date din format care au fost definite cu atribut de SALT AUTOMAT, și numai în acestea, nu se mai așteaptă introducerea de date de la terminal; ele păstrind valoarea introdusă anterior.

Anularea efectului CTRL/D se face tastind CTRL/E. Dacă se tastează CTRL/C se anulează efectele CTRL/B și CTRL/D.

Efectele tastelor CTRL/B și CTRL/D se limitează la formatul de date curent afișat pe ecran.

### Terminarea sesiunii de lucru

Sesiunea de lucru DRW, lansată fără comutatorul EN, se încheie prin tastarea CTRL/Z. Dacă în linia de comandă DRW a fost folosit comutatorul EN, apăsarea CTRL/Z va avea ca efect revenirea în formatul de date anterior. Dacă formatul de date curent este de primul nivel, atunci CTRL/Z termină sesiunea de lucru cu DRW.

## 3. Alte extensii ale produsului

### Noi facilități ale procesorului DFS

#### Extragerea structurii

Noua formă a liniei de comandă pentru apelul procesorului DFS este

```
>DFS [fiș-listare/sw1] [fiș-extras] [=fiș-int1/sw2 [,fiș-int2/sw2...]]
```

cu cazul particular

```
>DFS, fiș-extras
```

„Fiș-extras” este fișierul în care sint extrase, sub formă de fișier sursă, toate caracteristicile din domeniul STR.

Funcția este utilă în cazurile în care s-au pierdut fișierele sursă originale ce au stat la bază catalogării structurii actuale, sau când nu se mai cunoaște ordinea lor de catalogare ori opțiunile de aliniere (JSA sau /-SA). Fișierul sursă extras poate fi utilizat pentru modificarea sau adăugarea de noi caracteristici la entități deja existente prin recatalogarea structurii.

#### Afișarea informațiilor despre dimensionarea spațiului real

Fișierului de listare de la apelul procesorului DFS i-a fost asociat un nou comutator /REZ, care are ca efect afișarea unei liste rezumate relative la ocuparea spațiului virtual la fiecărui domeniu, precum și la modul de pliere al acestuia peste spațiul real corespunzător.

Lista este utilă pentru dimensionarea spațiului real al fiecărui domeniu, atât din punct de vedere al ocupării spațiului cit și din acela al creșterii performanțelor.

## Noi funcțiuni ale procesorului SOC

### Extensii SGF : append și update

În funcția ATTACHE, parametrul „prm” are ca valori posibile :

- I pentru input
  - O pentru output
  - citire cu LIRE ;
  - modificare ;
  - A pentru append
  - U pentru update
  - rescriere cu ECRIRE ;
- Valoarea A are ca efect adăugarea articolelor scrise cu ECRIRE în coadă vechiului fișier.  
Valoarea U permite modificarea articolelor unui fișier prin :

### APPEL și EXEC cu Zi

Noua sintaxă a instrucțiunilor APPEL și EXEC este următoarea:

```

APPEL {nume-subprogram} [ { BUFFER n } ]
APPEL { Zi } [ AVEC { contantă-Z } ]
      { identifiator } [ { Zj } ]
EXEC { Zi } [ (Xi Xj ... ) ]

```

Această nouă sintaxă permite utilizarea dinamică a instrucțiunilor APPEL și EXEC, prin introducerea prealabilă a numelui task-ului sau precompilatului dorit într-o variabilă Zi.

### M Zi parțial cu Yi

În sintaxa

$$Zi [ (p1) ] = Zj (p2 \lg)$$

oricare din valorile p1, p2 sau lg pot fi specificate fie prin constante numerice întregi, fie prin variabile de tip Yi. Aceasta permite o prelucrare mai facilă a șirurilor de caractere.

La execuția instrucțiunii se face verificarea variabilelor Yi implicate (să nu conțină nedefinit, zero, valori negative sau prea mari). și în caz contrar este poziționat codul 23 în variabila ERREUR.

## Noi facilități ale procesorului MGS

În comenzile procesorului MGS :SUPMAC, :SUPPRO, :LISMAC, :LISPRO și LISALL, identifiatorul din corpul comenzii poate fi de forma :

șir-caractere-1\*șir-caractere-2

Comanda acționează asupra tuturor macroinstrucțiunilor și/sau precompilatelor ale căror nume încep cu șir-caractere-1 și se termină cu șir-caractere-2. Atât șir-caractere-1 cât și șir-caractere-2 pot lipsi.

În particular, comenzile în care identifiatorul este înlocuit cu „\*” acționează asupra tuturor macroinstrucțiunilor/precompilatelor.

## **Sporirea confidențialității : drepturi de compilare**

Pentru sporirea confidențialității, deziderat solicitat de utilizatorii SOCRATE-MINI, a fost introdus la apelul utilitarului GFM comutatorul /CP:n prin intermediul căruia poate fi definit un utilizator unic cu drepturi de compilare și de catalogare în bază. Opțional, se poate interzice tuturor compilarea și catalogarea în bază. Singură instrucțiune exceptată de la prevederile de mai sus este instrucțiunea EXEC.

## **Extinderea setului de caractere pentru procesoare**

Setul de caractere permis în analiza lexicală, realizată de procesoarele DFS, SOC și MGS a fost extins astfel :

- identificatorii și cuvintele cheie pot fi scrise și cu litere mici ;
- printre separatorii de limbaj a fost introdus și <TAB> ;
- în constantele alfanumerice de forma 'șir-de-caractere' sînt permise toate caracterele din setul ASCII, mai puțin „'” și „?”.

## CĂRȚI APĂRUTE ÎN 1986-88 CARE SE MAI POT AFLA ÎN REȚEAUA DE LIBRĂRII

### Biblioteca de automată informatică, electronică, management

1. Adrian Davidovicu  
Boldur Bărbat
  2. Marius Guran  
Florin Filip
  - 3-4. Cr. Giumale, D. Prețeșcu,  
L. D. Serbănași, D. Tușiș,  
Gh. Tecuci, D. Cristea
  5. M. Suciș, D. Popescu  
Tr. Ionescu
  - 6-7. Baron, Al. Isaic-Maniu,  
L. Tövissi, D. Niculescu,  
C. Baron, V. Antonescu,  
I. Roman
  8. Gh. Turbuș, I. Boicu,  
E. Spirea, M. Hușanu,  
I. Tomescu și colectiv 100 specialiști din MTTc, ITCI, IPA
  - 9-12. Colective largi
  13. T. Geber, V. Cristea,  
V. Săvescu, I. Miu,  
R. Bulgakov, M. Vuici
- Limbaje de programare pentru sisteme în timp real, 224, pag. 23 lei  
Sisteme ierarhizate în timp real, cu prelucrare distribuită a datelor, 296, pag. 29 lei  
LISP. Programare (DA-LISP). Probleme rezolvate. Aplicații complexe LISP 86 pentru micro (Felix PC, M 216) și TC LISP pentru minicalculatoare (CORAL Independent). Sisteme de I. A pe TC Lisp 2 vol., 44 lei  
Microprocesoare, microcalculatoare și roboți în automatizări industriale, 384 pagini, 28 lei  
Calitate și fiabilitate, manual practic, 2 vol., 1100 pag. (cu 300 aplicații, exemple, studii de caz, 4 rigle de calcul, 5 standarde, 2000 teste-întrebări rezolvate) 139 lei  
Inginerie de sistem, automatizări și informatică în transporturi feroviare, navale, aeriene, rutiere, vol. 1, 758 pag. 75 lei  
Automatizări, management, calculatoare (AMC) volumele 52-55 ~ 2000 pag. ~ 180 lei (a se căuta și volumele anterioare, în curs de epuizare).  
Echipamente periferice vol. 3, 260 pag., 19 lei.

## CĂRȚI ÎN CURS DE APARIȚIE

### Biblioteca de automată informatică, electronică, management

- 1-2. Gh. Sabău, Al. Sotir și colectiv 11 specialiști ASE, CSP, ITCI, Centrul de calcul teritorial Constanța
  3. Gh. Turbuș, E. Spirea, M. Hușanu, I. Ionescu, I. Boicu și colectiv 100 specialiști MTTc, ITCI, IPA, IPB, Centru de calcul Teritorial Constanța
  - 4-5. L. Dumitrașcu
  - 6-7. N. Patrubani
- Practica bazelor de date. Totul despre... SOCRATE și SOCRATE-MINI pe Felix C, CORAL, INDEPENDENT. Volumele 1 și 2, 768 pagini, Seria Practică, Lei 62, apare în trim. I.  
Inginerie de sistem, automatizări și informatică în transporturi feroviare, navale, aeriene, rutiere. Volumul 2, 1000 pagini, Seria Fundamente, 85 lei ; apare în trim. I  
Învățăm microelectronica interactivă conversând în BASIC. Totul despre... BASIC în 14 conversații și 7 sinteze pe Felix C, CORAL, INDEPENDENT, Felix PC, M118, TPD, HC.85, aMIC, COMMODORE, AMSTRAD și compatibile, volumele 1 și 2, 980 pag. lei 100, apare în trim. II.  
Totul despre... microprocesorul Z80, vol. 1 și vol. 2, 700 pagini, (o parte a tirajului însoțită de o casetă pentru un simulator al funcționării microprocesorului pe calculatoarele personale PRAE și aMIC), 160 lei fără casetă, cu casetă 70 lei

- 8- 9. A. Petrescu și colectiv IPB, ITCI, Fabrica de calculatoare, Liceul Dimitrie Cantemir, CNOP
- 10-11. A. Tănăsescu și colectiv IPB, ISPIF
- 12-17. Colective largi
18. A. Davidoviciu și colectiv ITCI, ASE
- 19-20. I. Văduva, V. Baltac, Florescu V. și colectiv ASE, ITCI
21. O. Rusu, I. Brudaru
22. P. Constantinescu
- ABC de calcul electronic. Totul despre... ..HC 85, vol. 1 și vol. 2, 700 pagini (o parte a tirajului cu 2 casete cu programe, acționând calculatoare personale HC85 și compatibile SINCLAIR SPECTRUM), 250 lei cu casete, 70 lei fără casete.**
- Grafică asistată de calculator. Programe Fortran pe minicalculatoare, pentru reprezentări geometrice, vol. 1 și vol. 2, 800 pag. 80 lei, apare în trim. III.**
- Automatică, management, calculatoare (AMC). Serie continuă de instruire, informare, sinteze, cercetări aplicative în sisteme electronice, automate, informatice, de conducere. Volumele 56-61. Volume de ~ 350 pag. și ~ 35 lei fiecare. Echipamente electronice și tehnică de calcul-manuale de utilizare. Calculatoare personale, programe. Aplicații informatice în ramuri industriale. Proiectarea asistată de calculator. Grafică interactivă. Automatizarea și informatizarea proceselor. Conducere și organizare asistată. Limbaje și produse program, ș.a.m.d., apar în trim. II-IV.**
- Sistemul de operare MIX și limbajul MACRO pentru minicalculatoarele CORAL/INDEPENDENT, 2 volume, 800 pagini, 90 lei**
- Informatică economică, 2 volume, 800 pag. (o parte a tirajului cu dischetă). Preț 80 lei fără dischetă**
- Echilibrarea liniilor flexibile, 300 pag. 30 lei, apare în trim. III Sinergia, informația și geneza sistemelor 350 pag., 35 lei, apare în trim. III**

Cărțile prevăzute pentru apariția eșalonată în trimestrele I-IV 1989 (cea mai mare parte au avut aparițiile amânate din 1988).

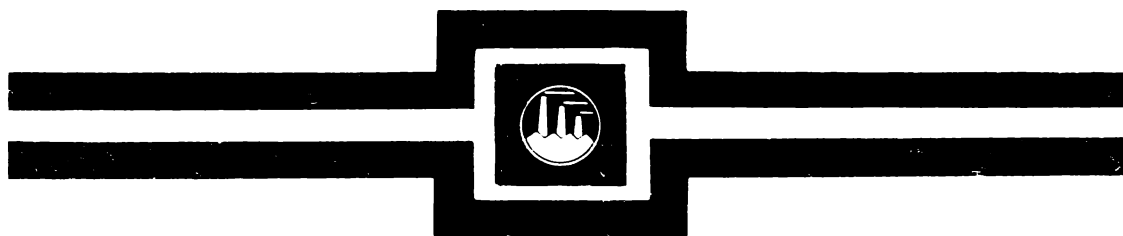
Se difuzează prin unitățile centrelor de librării, spre care se îndrumă întreprinderile și cititorii.

**PENTRU ACESTE CĂRȚI SE POT FACE TOTUȘI, ȘI COMENZI FERME LA EDITURA TEHNICĂ, PIAȚA SCINTEII 1, BUCUREȘTI.**

Comenzile întreprinderilor se semnează de director și contabil șef, cele ale cititorilor individuali au indicată adresa exactă. Comenzile se trimit de editură la centrele de librării, cu indicarea unor priorități de satisfacere a lor. Plata nu se face decât la primirea exemplarelor de la rețeaua de librării.



- **SOCRATE** este sistemul de gestiune al bazelor de date pe calculatoarele românești Felix C, provenit de la familia franceză IRIS-50. Acest SGBD este cunoscut și disponibil de mult timp în toate unitățile de calcul electronic din țară. **SOCRATE-MINI** este un SGBD de același tip, realizat în ultimii ani de specialiști ai Centrului Teritorial de Calcul Constanța pentru minicalculatoarele CORAL/INDEPENDENT și cele compatibile cu familia americană PDP 11.
- Cartea îmbină preocupările a două grupuri de specialiști: primul compus din cadre didactice, analiști și programatori de sistem și aplicații de la Academia de Studii Economice, Institutul de Tehnică de Calcul și Informatică, Comitetul de Stat al Planificării și al doilea — din elaboratori ai produsului-program **SOCRATE-MINI**.



**EDITURA TEHNICĂ**

- Au rezultat două volume care își merită subtitlul „Totul despre...”, deoarece, pe de o parte, **SOCRATE**, acest SGBD — de tip rețea, mai complex decît unul relațional — este tratat complet, pornind de la limbajele de descriere și de manipulare a datelor, copios exemplificate, pînă la optimizări și dezvoltări **SOCRATE** și alte SGBD-uri, iar pe de altă parte, **SOCRATE-MINI** este reprezentat prin manualele de lansare-utilizare-operare pînă la ultima versiune (V 4.0), cu multiple aplicații. Și nu numai atît; fiecare volum se încheie cu cite un studiu de caz, în extenso: o aplicație de personal, respectiv una de transport naval.
- Datorită acestei structurări, spectrul celor interesați este deosebit de larg: informaticieni, cadre didactice, studenți, utilizatori actuali și potențiali de baze de date, din toate domeniile economiei naționale.

aze