

# Cuprins

<i>Prefață la ediția a II-a</i> .....	11
<b>Capitolul 1. Despre bazele de date</b> .....	13
1.1. Nevoia de baze de date .....	13
1.2. Cum ne folosim de bazele de date .....	14
1.3. La început a fost fișierul (independent) .....	16
1.4. Ce este o bază de date? .....	18
1.5. Sisteme de gestiune a bazelor de date .....	20
1.6. Module, limbaje și utilizatori ai SGBD-urilor .....	23
1.7. Utilizatorii bazelor de date .....	26
1.8. Modele de organizare a datelor în BD .....	28
1.9. Noțiuni preliminare în lucrul cu bazele de date .....	30
<b>Capitolul 2. Modelul relațional de organizare a datelor</b> .....	37
2.1. Structura .....	38
2.2. Restricții .....	42
2.3. Schema și conținutul unei baze de date relaționale .....	51
2.4. SGBDR-uri și servere de baze de date relaționale .....	63
<b>Capitolul 3. Standarde și dialecte SQL. Tipuri de date. Crearea tabelor și     modificarea conținutului</b> .....	73
3.1. Standardizarea SQL .....	75
3.2. Tipuri de date primitive SQL .....	82
3.3. Tabele și restricții în SQL .....	87

3.4. Modificarea conținutului .....	114
3.5. Tranzacții .....	133
<b>Capitolul 4. Limbaje de interogare. Algebra relațională .....</b>	<b>139</b>
4.1. Caracterizare generală a limbajelor de interogare .....	139
4.2. Operatorii algebrei relaționale .....	141
4.3. Operatorii ansambliști .....	142
4.4. Operatorii relaționali .....	145
4.5. Notații și interpretări grafice ale interogărilor .....	168
<b>Capitolul 5. Sintaxa de bază a comenzii SELECT .....</b>	<b>175</b>
5.1. Trei clauze și un rezultat .....	175
5.2. Reuniuni, intersecții, diferențe, produse carteziane .....	183
5.3. Coloane-expresii .....	187
5.4. Ordonări .....	195
5.5. Operatorii BETWEEN, OVERLAPS și IN .....	198
5.6. Comparații inexacte. LIKE și SIMILAR .....	201
5.7. Joncțiuni interne .....	207
5.8. Sinonime locale și joncțiunea unei tabele cu ea însăși .....	215
<b>Capitolul 6. Funcții SQL .....</b>	<b>219</b>
6.1. Funcții pentru numere .....	219
6.2. Funcții pentru șiruri de caractere .....	225
6.3. Funcții pentru date calendaristice .....	231
6.4. Funcții pentru intervale .....	235
6.5. Conversii între tipuri de date .....	240
6.6. Alte funcții-sistem .....	249
6.7. Funcții-agregat .....	251
<b>Capitolul 7. Grupări .....</b>	<b>269</b>
7.1. Gruparea după un criteriu .....	269

7.2. Gruparea după două sau mai multe criterii .....	275
7.3. Gruparea după funcții și expresii .....	288
7.4. Clauza HAVING .....	297
7.5. Diviziunea relațională .....	304
<b>Capitolul 8. NULL-ități, joncțiuni externe și structuri alternative .....</b>	<b>307</b>
8.1. NULL-i în tabele .....	307
8.2. Operatorul IS NULL .....	313
8.3. Transformarea NULL-ilor .....	321
8.4. Joncțiunea externă .....	326
8.5. Structuri de control alternative .....	337
<b>Capitolul 9. Subconsultări .....</b>	<b>357</b>
9.1. Subconsultări în clauza WHERE. Operatorul IN .....	357
9.2. Subconsultări și comparații în clauza WHERE .....	373
9.3. Subconsultări în clauza HAVING .....	380
9.4. Subconsultări în clauza FROM .....	394
9.5. Subconsultări scalare în clauza SELECT .....	428
9.6. Expresii-tabelă .....	433
<b>Capitolul 10. Subconsultări corelate .....</b>	<b>443</b>
10.1. Subconsultări corelate în clauza SELECT .....	443
10.2. Subconsultări corelate în clauza WHERE .....	462
10.3. Alte tipuri de corelări simple .....	475
10.4. Corelarea dublă .....	479
<b>Capitolul 11. OLAP în SQL .....</b>	<b>485</b>
11.1. Subtotaluri .....	485
11.2. Analize multidimensionale. Operatorii CUBE și GROUPING SETS .....	500
11.3. Clasamente - soluții clasice și OLAP .....	515

11.4. Ferestre pentru funcțiile analitice .....	532
11.5. Comparații și ponderi .....	540
<b>Capitolul 12. SQL, ierarhii și recursivitate .....</b>	<b>547</b>
12.1. Autojuncțiuni și subconsultări .....	547
12.2. Interogări arborescente în Oracle .....	557
12.3. Interogări arborescente în MS SQL Server .....	567
12.4. Interogări arborescente în DB2 .....	574
12.5. Interogări arborescente în PostgreSQL .....	577
12.6. Interogări mai puțin arborescente, dar tot recursive .....	580
<b>Capitolul 13. Tabele, restricții și actualizări prin interogări SQL .....</b>	<b>591</b>
13.1. Crearea tabelelor folosind interogări .....	591
13.2. Restricții și aserțiuni pe bază de interogări .....	603
<b>Capitolul 14. Tabele temporare și virtuale .....</b>	<b>633</b>
14.1. Tabele temporare .....	633
14.2. Tabele virtuale în interogări .....	639
14.3. Probleme ale actualizărilor tabelelor-sursă pe baza modificării tabellelor virtuale .....	641
14.4. Restricții în tabele virtuale .....	651
<b>Capitolul 15. Scheme, obiecte și utilizatori .....</b>	<b>653</b>
15.1. Crearea de subscheme și acordarea de drepturi .....	653
15.2. Informații despre obiectele din (sub)scheme .....	664
<b>Capitolul 16. Funcții, proceduri și pachete stocate .....</b>	<b>689</b>
16.1. Funcții stocate .....	690
16.2. Curse .....	712
16.3. Funcții ce furnizează seturi de înregistrări .....	721

16.4. Funcții folosite ca valori implicite și restricții la nivel de atribut/înregistrare .....	730
16.5. Proceduri stocate .....	733
16.6. Pachete .....	748
<b>Capitolul 17. Declanșatoare</b> .....	<b>753</b>
17.1. Tipologia declanșatoarelor .....	753
17.2. Declanșatoare în PL/pgSQL .....	755
17.3. Declanșatoare în PL/SQL .....	766
17.4. Declanșatoare în Transact-SQL .....	778
17.5. Declanșatoare în SQL PL .....	787
<b>Capitolul 18. SQL dinamic</b> .....	<b>793</b>
18.1. SQL dinamic în PostgreSQL PL/pgSQL .....	793
18.2. SQL dinamic în PL/SQL .....	802
18.3. SQL dinamic în Transact-SQL .....	809
18.4. SQL dinamic în IBM DB2 SQL PL .....	816
<b>Capitolul 19. Obiectual-relațional în SQL</b> .....	<b>825</b>
19.1. Tipuri distincte și structurate de date .....	826
19.2. Colecții .....	838
19.3. Stocarea tipurilor-utilizator în tabele .....	846
19.4. Metode asociate tipurilor structurate în Oracle .....	854
<i>Bibliografie</i> .....	869

Iar dacă dorim să extragem clienții cu adresele nenule, în niciun caz nu folosim operatorul NOT IN, deoarece interogarea:

```
SELECT *
```

```
FROM clienti
```

```
WHERE Telefon NOT IN (NULL)
```

nu extrage nicio linie. După cum vom vedea în paragraful 8.2, în SQL putem folosi în predicatul de selecție clauza IS NULL, prin care aceste probleme sunt rezolvate destul de ușor.

Tot în paragraful 5.5 am văzut că DB2-ul nu acceptă includerea unui NULL într-o listă-argument a operatorului IN (figura 5.23)<sup>1</sup>.

### 8.1.2. Două tabele suplimentare

Deși baza de date VÂNZĂRI este alcătuită dintr-un număr considerabil de tabele și atribute, introducem încă două tabele cu scopul de a gestiona o parte din datele privind drepturile bănești (salariu negociat și sporuri) ale angajaților firmei. Ambele tabele nu au nimic în comun cu vânzările, ci doar cu salarizarea. Prima tabelă se numește PERSONAL2 și conține date generale despre angajați: marcă; nume și prenume; data nașterii; compartiment; marca șefului (direct); salariu tarifar. A doua, SPORURI, evidențiază sporurile lunare primite de fiecare angajat: sporul de vechime (SporVechime), sporul pentru schimbul de noapte (SporNoapte), sporuri pentru condiții deosebite (SporCD) și sporuri diverse (AlteSpor).

Cu ajutorul valorii NULL se poate face diferența între angajații pentru care nu s-a calculat valoarea sporului pe luna curentă (și care vor avea în câmpul corespunzător valoarea NULL) și cei care nu au dreptul la un asemenea spor, adică valoarea este 0. Scriptul de creare a celor două tabele este prezentat în listingul 8.1.

Listing 8.1. Script DB2/PostgreSQL de creare a tabelor SPORURI și PERSONAL2<sup>2</sup>

```
DROP TABLE sporuri;
DROP TABLE personal2;

CREATE TABLE personal2(
  Marca NUMERIC(5) NOT NULL
    CONSTRAINT pk_personal2 PRIMARY KEY,
  NumePren VARCHAR(40),
  DataNast DATE,
  Compart VARCHAR(20),
  MarcaSef NUMERIC(5)
    CONSTRAINT fk_personal2 REFERENCES personal2(marca),
  SalTarifar NUMERIC(12,2));

CREATE TABLE sporuri(
  An NUMERIC(4) NOT NULL,
  Luna NUMERIC(2) NOT NULL,
```

1. Mesajul de eroare DB2 este, în cazul lansării în execuție a ultimelor două interogări, *SQL0206N "NULL" is not valid in the context where it is used. SQLSTATE = 42703.*
2. În Oracle, în loc de VARCHAR se folosește VARCHAR2, iar în SQL Server, în loc de DATE se folosește SMALLDATETIME.

Marca NUMERIC(5) NOT NULL  
 CONSTRAINT fk\_sporuri\_pers2 REFERENCES personal2 (marca),  
 SporVechime NUMERIC(12,2),  
 SporNoapte NUMERIC(12,2),  
 SporCD NUMERIC(12,2),  
 AlteSpor NUMERIC(12,2),  
 CONSTRAINT pk\_sporuri PRIMARY KEY (an,luna,marca);

Figurile 8.4 și 8.5 ilustrează câteva înregistrări de test ale celor două tabele. Tabela PERSONAL2 conține zece linii, deci, la momentul curent, firma are zece angajați. Pentru doi (angajații 4 și 7) nu li se cunoaște data nașterii. Cinci angajați (2, 4, 5, 6 și 7) lucrează în compartimentul Financiar, trei (3, 8 și 9) la compartimentul Marketing, iar unul (10) la Resurse umane. Există și un angajat care lucrează în cadrul compartimentului Direcțiune. Bănuim că este vorba despre directorul general, bănuială confirmată de faptul că, în linia corespunzătoare acestui angajat (prima), atributul MarcaŞef, care indică marca șefului direct al fiecărui angajat, are valoarea NULL. Cu alte cuvinte, Angajat 1 nu are șef.

MARCA	NUMEPREN	DATANAST	COMPART	MARCASEF	SALTARIFAR
1 ANGAJAT 1		01-07-1962	DIRECTIUNE	(null)	1600
2 ANGAJAT 2		11-10-1977	FINANCIAR	1	1450
3 ANGAJAT 3		02-08-1962	MARKETING	1	1450
4 ANGAJAT 4		(null)	FINANCIAR	2	1380
5 ANGAJAT 5		30-04-1965	FINANCIAR	2	1420
6 ANGAJAT 6		09-11-1965	FINANCIAR	5	1350
7 ANGAJAT 7		(null)	FINANCIAR	5	1280
8 ANGAJAT 8		31-12-1960	MARKETING	3	1290
9 ANGAJAT 9		28-02-1976	MARKETING	3	1410
10 ANGAJAT 10		29-01-1972	RESURSE UMANE	1	1370

Figura 8.4. Câteva înregistrări în tabela PERSONAL2

Dacă privim cu atenție și figura 8.5, mai putem bănuși că firma s-a înființat în aprilie 2007, când avea numai trei angajați. Lucrul acesta ne este sugerat de existența, pentru această lună, a numai trei angajați pentru care au fost introduse sporuri. La momentul curent (primul trimestru al anului 2008) sunt 10 angajați.

AN	LUNA	MARCA	SPORVECHIME	SPORNOAPTE	SPORCD	ALTESPOR
2007	4	1	160	0	0	132
2007	4	2	130	45	0	70
2007	4	3	145	156	420	157
2007	5	1	160	0	0	0
2007	5	2	80	45	0	70
2007	5	3	145	0	0	0
2007	5	10	137	0	0	430
2007	6	1	160	0	0	0
2007	6	2	80	0	0	150
2007	6	4	50	15	88	120
2007	6	5	130	15	0	20
2007	6	10	200	12	0	6
2007	7	1	160	0	(null)	(null)
2007	7	2	80	0	0	158
2007	7	3	145	0	0	0
2007	7	4	50	15	(null)	15
2007	7	5	130	0	0	120
2007	7	6	110	147	0	0
2007	7	7	60	210	0	0
2007	7	8	130	0	15	0
2007	7	9	140	100	77	0
2007	7	10	200	0	0	120

Figura 8.5. Câteva înregistrări în tabela SPORURI

Popularea celor trei tabele este conținută în listingul 8.2.

Listing 8.2. Script Oracle/PostgreSQL de populare a tabelor SPORURI și PERSONAL2<sup>3</sup>

```

DELETE FROM personal2;
INSERT INTO personal2 VALUES (1, 'ANGAJAT 1', DATE'1962-07-01',
'DIRECTIUNE', NULL, 1600);
INSERT INTO personal2 VALUES (2, 'ANGAJAT 2', DATE'1977-10-11',
'FINANCIAR', 1, 1450);
INSERT INTO personal2 VALUES (3, 'ANGAJAT 3', DATE'1962-08-02',
'MARKETING', 1, 1450);
INSERT INTO personal2 VALUES (4, 'ANGAJAT 4', NULL,
'FINANCIAR', 2, 1380);
INSERT INTO personal2 VALUES (5, 'ANGAJAT 5', DATE'1965-04-30',
'FINANCIAR', 2, 1420);
INSERT INTO personal2 VALUES (6, 'ANGAJAT 6', DATE'1965-11-09',
'FINANCIAR', 5, 1350);
INSERT INTO personal2 VALUES (7, 'ANGAJAT 7', NULL, 'FINANCIAR', 5, 1280);
INSERT INTO personal2 VALUES (8, 'ANGAJAT 8', DATE'1960-12-31',
'MARKETING', 3, 1290);
INSERT INTO personal2 VALUES (9, 'ANGAJAT 9', DATE'1976-02-28',
'MARKETING', 3, 1410);
INSERT INTO personal2 VALUES (10, 'ANGAJAT 10', DATE'1972-01-29',
'RESURSE UMANE', 1, 1370);

DELETE FROM sporuri ;
INSERT INTO sporuri VALUES (2007, 4, 1, 160, 0, 0, 132);

```

3. Versiunea DB2/SQL Server a scriptului se obține prin eliminarea cuvintelor DATE.



```

INSERT INTO sporuri VALUES (2007, 4, 2, 130, 45, 0, 70);
INSERT INTO sporuri VALUES (2007, 4, 3, 145, 156, 420, 157);
INSERT INTO sporuri VALUES (2007, 5, 1, 160, 0, 0, 0);
INSERT INTO sporuri VALUES (2007, 5, 2, 80, 45, 0, 70);
INSERT INTO sporuri VALUES (2007, 5, 3, 145, 0, 0, 0);
INSERT INTO sporuri VALUES (2007, 5, 10, 137, 0, 0, 430);
INSERT INTO sporuri VALUES (2007, 6, 1, 160, 0, 0, 0);
INSERT INTO sporuri VALUES (2007, 6, 2, 80, 0, 0, 150);
INSERT INTO sporuri VALUES (2007, 6, 4, 50, 15, 88, 120);
INSERT INTO sporuri VALUES (2007, 6, 5, 130, 15, 0, 20);
INSERT INTO sporuri VALUES (2007, 6, 10, 200, 12, 0, 6);
INSERT INTO sporuri VALUES (2007, 7, 1, 160, 0, NULL, NULL);
INSERT INTO sporuri VALUES (2007, 7, 2, 80, 0, 0, 158);
INSERT INTO sporuri VALUES (2007, 7, 3, 145, 0, 0, 0);
INSERT INTO sporuri VALUES (2007, 7, 4, 50, 15, NULL, 15);
INSERT INTO sporuri VALUES (2007, 7, 5, 130, 0, 0, 120);
INSERT INTO sporuri VALUES (2007, 7, 6, 110, 147, 0, 0);
INSERT INTO sporuri VALUES (2007, 7, 7, 60, 210, 0, 0);
INSERT INTO sporuri VALUES (2007, 7, 8, 130, 0, 15, 0);
INSERT INTO sporuri VALUES (2007, 7, 9, 140, 100, 77, 0);
INSERT INTO sporuri VALUES (2007, 7, 10, 200, 0, 0, 120);

```

## 8.2. Operatorul IS NULL

Valoarea NULL nu se confundă cu valoarea 0 – pentru atributele numerice – sau cu valoarea ' ' – spațiu – pentru atributele de tip șir de caractere. Este important de notat că, în vederea identificării valorilor nule, operatorul are forma *IS NULL* și nu *= NULL*.

*Pentru care dintre clienți nu se cunoaște adresa?*

Soluția se bazează pe utilizarea operatorului IS NULL care extrage toate valorile NULL pentru un atribut (figura 8.6):

```

SELECT *
FROM clienti
WHERE Adresa IS NULL

```

CODCL	DENCL	CODFISCAL	ADRESA	CODPOST	TELEFON
1002	Client 2 SA	R.1002	(null)	700505	0232212121
1005	Client 5 SRL	R.1005	(null)	701900	0256111111

Figura 8.6. Clienții „fără adresă”

Prin execuția frazei SQL:

```

SELECT *
FROM clienti
WHERE Adresa = NULL

```

se va obține o tabelă cu 0 (zero) linii sau chiar o eroare (în DB2<sup>4</sup>). Rezultatul evaluării `Adresa = NULL` va fi întotdeauna FALSE.

În unele SGBD-uri, precum Oracle, există o sursă de confuzie între șirul de caractere de lungime zero (" , adică două apostrofuli lipite, fără spațiu) și NULL-ul propriu-zis. Astfel, dacă vrem să NULL-izăm adresa clientului cu codul 1007 putem recurge la varianta următoare a comenzii UPDATE:

```
UPDATE clienti
SET Adresa = "
WHERE CodCl=1007
```

Atribuirea s-a făcut, deci, cu `Adresa = "`, și nu cu `Adresa = NULL`. Rezultatul este același, dacă examinăm linia acestui client după comanda UPDATE (vezi figura 8.7).

CODCL	DENCL	CODFISCAL	ADRESA	CODPOST	TELEFON
1007	Client 7 SRL	R1007	(null)	701900	0256121212

Figura 8.7. NULL-itatea adresei clientului 1007 (în Oracle)

Prin urmare, dacă șirul vid " (de lungime zero) este echivalentul valorii NULL, la atribuire, nu la fel stau lucrurile în clauza WHERE, întrucât interogarea Oracle:

```
SELECT *
FROM CLIENTI
WHERE Adresa = "
```

„continuă” să nu extragă nicio linie. Lucrurile nu stau la fel în DB2 și în MS SQL Server, întrucât interogarea extrage linia corespunzătoare clientului 1007, valoarea adresei pentru acest client nefiind considerată nulă. În concluzie, dacă pentru DB2, PostgreSQL și SQL Server, șirul de caractere de lungime zero - " (două apostrofuli, nu ghilimele!) - este diferit de NULL, pentru Oracle uneori este diferit (în clauza WHERE), altelei același lucru (în clauza SET a comenzii UPDATE).

*Care sunt persoanele și lunile pentru care nu s-a calculat (nu se cunoaște) sporul pentru condiții deosebite?*

Prin interogarea:

```
SELECT sporuri.Marca, NumePren, Compart, An, Luna
FROM personal2
INNER JOIN sporuri ON personal2.Marca = sporuri.Marca
WHERE SporCD IS NULL
```

se obține situația din figura 8.8.

MARCA	NUMEPREN	COMP...	AN	LUNA
1	ANGAJAT 1	DIRECTIUNE	2007	7
4	ANGAJAT 4	FINANCIAR	2007	7

Figura 8.8. Angajații pentru care nu s-au operat sporurile pentru condiții deosebite

4. Mesajul de eroare din DB2 la execuția comenzii este: `SQL0401N The data types of the operands for the operation "=" are not compatible. SQLSTATE=42818.`

Care sunt angajații și lunile în care aceștia nu au primit spor pentru condiții deosebite?

Atât soluția, cât și rezultatul sunt sensibil diferite (vezi figura 8.9), întrucât acum ne interesează valorile zero (nu NULL) ale SporCD:

```
SELECT sporuri.Marca, NumePren, Compart, An, Luna
```

```
FROM personal2 INNER JOIN sporuri
```

```
ON personal2.Marca = sporuri.Marca
```

```
WHERE SporCD = 0
```

```
ORDER BY An, Luna, NumePren
```

MARCA	NUMEPREN	COMPART	AN	LUNA
1 ANGAJAT 1	DIRECTIUNE	2007	4	
2 ANGAJAT 2	FINANCIAR	2007	4	
1 ANGAJAT 1	DIRECTIUNE	2007	5	
10 ANGAJAT 10	RESURSE UMANE	2007	5	
2 ANGAJAT 2	FINANCIAR	2007	5	
3 ANGAJAT 3	MARKETING	2007	5	
1 ANGAJAT 1	DIRECTIUNE	2007	6	
10 ANGAJAT 10	RESURSE UMANE	2007	6	
2 ANGAJAT 2	FINANCIAR	2007	6	
5 ANGAJAT 5	FINANCIAR	2007	6	
10 ANGAJAT 10	RESURSE UMANE	2007	7	
2 ANGAJAT 2	FINANCIAR	2007	7	
3 ANGAJAT 3	MARKETING	2007	7	
5 ANGAJAT 5	FINANCIAR	2007	7	
6 ANGAJAT 6	FINANCIAR	2007	7	
7 ANGAJAT 7	FINANCIAR	2007	7	

Figura 8.9. Angajații și lunile pentru care SporCD este zero (neNULL)

Care dintre angajați sunt născuți înainte de 1 ianuarie 1970 și care după această dată? Persoanele născute înainte (figura 8.10) se obțin prin:

```
SELECT *
```

```
FROM personal2
```

```
WHERE DataNast < DATE'1970-01-01'
```

MARCA	NUMEPREN	DATANAST	COMPART	MARCASEF	SALTARIFAR
1 ANGAJAT 1	01-07-1962	DIRECTIUNE	(null)	1600	
3 ANGAJAT 3	02-08-1962	MARKETING	1	1450	
5 ANGAJAT 5	30-04-1965	FINANCIAR	2	1420	
6 ANGAJAT 6	09-11-1965	FINANCIAR	5	1350	
8 ANGAJAT 8	31-12-1960	MARKETING	3	1290	

Figura 8.10. Persoane născute înainte de 1 ianuarie 1970

iar cele născute după 1 ianuarie 1970 (figura 8.11) prin: