

## CAPITOLUL 6. SELECTAREA DATELOR DIN TABELELE BAZEI DE DATE

### 6.1. Comanda *SELECT*

Pentru a selecta datele din una sau mai multe tabele se utilizează comanda **SELECT** a carei sintaxa este:

```
SELECT [ALL | DISTINCT]  
{[nume-tabela.]* |  
expr [sinonim], expr [sinonim],...}  
FROM nume-tabelă [@ legătură][sinonim],...  
nume-tabelă [@ legătură][sinonim],  
[WHERE condiție]  
[ CONNECT BY condiție [START WITH condiție] ]  
[ GROUP BY { expr, expr.. | CUBE ( expr, expr.. ) | ROLLUP  
( expr, expr ) } ] [ HAVING condiție ]  
[ { UNION | INTERSECT | MINUS } SELECT... ]  
[ ORDER BY {expr | număr-poziție} [ASC | DESC]  
{expr | număr-poziție}[ASC | DESC],...  
[ FOR UPDATE OF nume-col, nume-col,...[NOWAIT] ];
```

Clauzele comenzii trebuie utilizate în ordinea specificată în sintaxă, excepție făcând clauzele **CONNECT BY**, **START WITH**, **GROUP BY** și **HAVING** (care pot fi specificate în orice ordine). Clauzele **ORDER BY** și **FOR UPDATE OF** pot fi schimbate între ele. Clauza **ALL** determină afișarea tuturor rândurilor rezultate în urma cererii, spre deosebire de clauza **DISTINCT** care determină eliminarea duplicatelor, afișând doar rândurile distincte. Utilizarea caracterului asterisc (\*) are ca efect selectarea tuturor coloanelor din tabela specificată prin clauza **FROM**, în ordinea în care au fost definite la creare.

În situația finală, fiecare expresie formulată în comandă devine un nume de coloană. Totodată, orice *sinonim*, dacă este specificat, este folosit în scopul etichetării expresiei precedente din tabela afișată. Dacă *sinonimul* conține blankuri sau caractere speciale cum sunt "+" și "-", trebuie incluse între apostrofuri.

Pentru a evita ambiguitatea, în cazul în care tabelele conțin coloane cu același nume, este necesară calificarea coloanelor cu numele tabelului.

Identificarea tabelelor în care trebuie căutate datele corespunzătoare coloanelor specificați se face prin specificarea numelor lor după clauza **FROM**. În locul numelor de tabele se pot folosi sinonimele acestora.

Clauza **WHERE** specifică o condiție care este folosită pentru a selecta rândurile.

Clauza **CONNECT BY** indică faptul că rândurile formează o structură arborescentă. Prin această clauză sunt definite relațiile necesare pentru a conecta rândurile tabelului într-un arbore. Operatorul **PRIOR** folosit înaintea uneia din cele două părți ale condiției, definește nodul părinte iar în cealaltă parte nodul fiu. Clauza **START WITH**, prin specificarea unei condiții care trebuie satisfăcută, stabilește rândul folosit ca rădăcină a arborelui. Dacă se omite, comanda **SELECT** va returna o serie de arbori începând cu fiecare rând selectat. Existența clauzei **CONNECT BY** într-o comandă **SELECT** permite utilizarea pseudocoloanei **LEVEL**, care returnează valoarea 1 pentru nodul rădăcină, 2 pentru fiii nodului rădăcină, 3 pentru nepoți etc.

Clauzele **GROUP BY** și **HAVING** determină afișarea unor informații sintetice despre grupuri de rânduri care au aceeași valoare în una sau mai multe coloane. Aceste coloane sunt, în general, funcții de grup.

**ROLLUP** activează o comandă **SELECT** pentru a calcula mai multe niveluri de subtotaluri dintr-un grup specificat de dimensiuni. Calculează de asemenea și un total general. **ROLLUP** este o extensie simplă a clauzei **GROUP BY**, deci sintaxa este foarte ușor de folosit. Extensia **ROLLUP** este foarte eficientă și nu îngreunează o cerere. Rolul extensiei **ROLLUP** este foarte clar: crează subtotaluri care pornesc de la cel mai detaliat nivel până la un total general după gruparea care a fost precizată în clauza **ROLLUP**.

**CUBE** acționează asupra unui grup specificat de coloane și crează subtotaluri pentru toate combinațiile posibile între acestea. Dacă s-a specificat de exemplu: **CUBE** (timp, regiune, department), rezultatul va include toate valorile care ar fi incluse într-un **ROLLUP** plus combinații adiționale.

Pentru a combina rezultatele a două comenzi **SELECT** într-un singur rezultat, se folosesc operatorii **UNION**, **INTERSECT** și **MINUS**. **UNION** returnează rezultatele obținute de la fiecare cerere în parte, **INTERSECT** returnează doar rezultatele comune celor două cereri iar **MINUS** returnează rezultatele obținute de la prima cerere și care nu apar în urma celei de a doua selecții.

Pentru a putea folosi aceste clauze este necesar ca numărul și tipul coloanelor selectate de fiecare comandă **SELECT** să fie aceleași, lungimile lor putând fi diferite. Dacă sunt combinate mai mult de două comenzi **SELECT**, ele vor fi evaluate de la stînga la dreapta. Pentru a schimba ordinea de evaluare pot fi folosite parantezele. Totodată, cei trei operatori impun utilizarea cuvîntului **DISTINCT** în toate comenzile **SELECT**.

Clauza **ORDER BY** specifică ordinea în care trebuie returnate rîndurile distincte ale unei tabeli.

Clauza **FOR UPDATE** determină blocarea rîndurilor selectate ale tabeli astfel încât acestea nu vor mai putea fi actualizate de alți utilizatori pînă la deblocarea lor cu una din comenzile **COMMIT** sau **ROLL BACK**.

Comanda **SELECT ... FOR UPDATE** trebuie urmată de una sau mai multe comenzi **UPDATE ... WHERE**. Dacă se folosește clauza **NOWAIT**, selecția este considerată terminată chiar dacă rîndurile selectate de **FOR UPDATE** nu pot fi blocate deoarece alt utilizator lucrează cu ele.

## 6.2. Utilizarea clauzei *FROM*

Pentru a selecta una sau mai multe tabele și pentru a specifica, eventual, identificatorul proprietarului și legătura cu rețeaua se folosește secvența:

**SELECT ...**

**FROM [ident-proprietar] tabela [@LINK],...;**

*Example:*

1) Să se selecteze toate coloanele din tabela SALARIAȚI.

**SQL> SELECT \* FROM SALARIAȚI;** sau

**SQL> SELECT ALL FROM SALARIAȚI;**

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1111	AVRAM ION	VÂNZATOR	100000	21200	1000	1000
1222	BARBU DAN	VÂNZATOR	120000	20750	2000	1000
1000	COMAN RADU	ȘEF DEP	130000	35000	2500	1000
3500	DAN ION	VÂNZATOR	160000	24500	3550	2500
2500	VLAD VASILE	ȘEF DEP	160000	36500	1500	2500
3700	MANU DAN	VÂNZATOR	160000	27500	2500	2500
2650	VLAD ION	VÂNZATOR	120000	25060	3500	1000

*7 records selected.*

2) Să se selecteze coloanele MARCA, NUME, SALA, VENS din tabela SALARIAȚI.

**SQL> SELECT MARCA,NUME,SALA,VENS  
2 FROM SALARIAȚI;**

MARCA	NUME	SALA	VENS
1111	AVRAM ION	21200	1000
1222	BARBU DAN	20750	2000
1000	COMAN RADU	35000	2500
3500	DAN ION	24500	3550
2500	VLAD VASILE	36500	1500
3700	MANU DAN	27500	2500
2650	VLAD ION	25060	3500

*7 records selected.*

3) Să se selecteze coloana NUME din tabela SALARIAȚI

**SQL> SELECT NUME  
2 FROM SALARIAȚI;**

NUME  
AVRAM ION  
BARBU DAN  
COMAN RADU  
DAN ION  
VLAD VASILE  
MANU DAN  
VLAD ION

*7 records selected.*

4) Să se selecteze coloana FUNCT din tabela SALARIAȚI în variantele utilizării și neutilizării clauzei DISTINCT.

**SQL> SELECT FUNCT AFIȘARE\_FUNCȚIE  
2 FROM SALARIAȚI;**

AFISARE\_FUNCȚIE  
VÂNZATOR  
VÂNZATOR  
SEF DEP  
VÂNZATOR  
SEF DEP  
VÂNZATOR  
VÂNZATOR

*7 records selected.*

**SQL>SELECT DISTINCT FUNCT  
2 AFIȘARE\_DISTINCT\_FUNCȚIE  
3 FROM SALARIAȚI;**

AFISARE DISTINCT FUNCTIE  
VÂNZATOR  
ŞEF DEP

*2 records selected.*

### 6.3 Utilizarea operatorilor în formularea condițiilor de selecție din clauza WHERE

Operatorii SQL\*Plus (Anexa2) pot apărea în orice parte a unei comenzi și au întâietate față de orice alt tip de operatori. Operatorii aritmetici și logici sunt utilizați pentru formularea condițiilor de selecție.

#### Exemple:

1) Să se selecteze toate înregistrările privind salariații al căror salariu este mai mare de 15000 u.m.:

**SQL> SELECT \* FROM SALARIAȚI**  
**2 WHERE SALA>15000 ;**

MARCA	NUME	FUNCT	CODD SALA	VENS	CODS
1111	AVRAM ION	VÂNZATOR	100000 21200	1000	1000
1222	BARBU DAN	VÂNZATOR	120000 20750	2000	1000
1000	COMAN RADU	ŞEF DEP	130000 35000	2500	1000
3500	DAN ION	VÂNZATOR	160000 24500	3550	2500
2500	VLAD VASILE	ŞEF DEP	160000 36500	1500	2500
3700	MANU DAN	VÂNZATOR	160000 27500	2500	2500
2650	VLAD ION	VÂNZATOR	120000 25060	3500	1000

*7 records selected.*

2) Să se selecteze coloanele MARCA și NUME, precum și veniturile totale (SALA+VENS) pentru angajații care au un salariu mai mare decât 35.000 u.m.

**SQL> SELECT MARCA, NUME, SALA+VENS**  
**2 FROM SALARIAȚI**  
**3 WHERE SALA>35000 ;**

MARCA	NUME	SALA + VENS
2650	VLAD VASILE	38000

*1 record selected*

3) Să se selecteze coloanele NUME, FUNCT și SALA pentru salariații care au funcția de vânzător.

**SQL> SELECT NUME, FUNCT FROM SALARIAȚI**  
**2 WHERE FUNCT='VÂNZATOR' ;**

NUME	FUNCT	SALA
AVRAM ION	VÂNZATOR	21200

BARBU DAN	VÂNZATOR	20750
DAN ION	VÂNZATOR	24500
MANU DAN	VÂNZATOR	27500
VLAD ION	VÂNZATOR	25060

*5 records selected.*

4) Să se selecteze coloanele MARCA, NUME, CODD din tabela SALARIAȚI, pentru salariații al căror venit suplimentar depășește salariul.

**SQL> SELECT MARCA, NUME, CODD**

**2 FROM SALARIAȚI**

**3 WHERE VENS>SALA ;**

*no records selected*

5) Să se selecteze și afișeze câmpurile MARCA, NUME, SALARIU pentru salariații ale căror venituri suplimentare sunt mai mari de 1.500 u.m. și lucrează în subordinea superiorului cu Codul 1000.

**SQL> SELECT MARCA, NUME, SALA**

**2 FROM SALARIAȚI**

**3 WHERE VENS>1500**

**4 AND CODS=1000 ;**

MARCA	NUME	SALA
1222	BARBU DAN	20750
1000	COMAN RADU	35000
2650	VLAD ION	25060

*3 records selected*

6) Să se afișeze toate coloanele pentru salariații cu funcția vânzător, care au salariul mai mare ca 20.000 u.m. și lucrează în subordinea superiorului cu Codul 1000.

**SQL> SELECT \* FROM SALARIAȚI**

**2 WHERE FUNCT='VANZATOR'**

**3 AND SALA>20000**

**4 AND CODS=1000 ;**

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1111	AVRAM ION	VÂNZATOR	100000	21200	1000	1000
1222	BARBU DAN	VÂNZATOR	120000	20750	2000	1000
2650	VLAD ION	VÂNZATOR	120000	25060	3500	1000

*3 records selected.*

7) Să se afișeze coloana NUME și SALA pentru angajații care au salariul mai mic ca 30.000 u.m.

**SQL> SELECT NUME FROM SALARIAȚI**

**2        WHERE SALA<30000 ;**

NUME	SALA
AVRAM ION	21200
BARBU DAN	20750
DAN ION	24500
MANU DAN	27500
VLAD ION	25060

*5 records selected.*

8) Să se selecteze înregistrările pentru care funcția este ‘SEF DEP’ sau salariul este mai mare decât 35.000 u.m.

**SQL> SELECT MARCA, CODS FROM SALARIAȚI**  
**2        WHERE FUNCT='SEF DEP' OR SALA>35000 ;**

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1000	COMAN RADU	ŞEF DEP	130000	35000	2500	1000
2500	VLAD VASILE	ŞEF DEP	160000	36500	1500	2500

*2 records selected.*

9) Să se selecteze datele despre salariații care au funcția de vânzător și nu lucrează în subordinea superiorului cu codul 1000,

**SQL> SELECT \* FROM SALARIAȚI**  
**2        WHERE FUNCT='VANZATOR'**  
**3        AND CODS!=1000;**

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
3500	DAN ION	VÂNZATOR	160000	24500	3550	2500
3700	MANU DAN	VÂNZATOR	160000	27500	2500	2500

*2 records selected.*

10) Să se selecteze toți salariații care lucrează în subordinea superiorului cu codul 1000 precum și cei care au salariul mai mic de 26.000 u.m. sau funcția de vânzător.

**SQL> SELECT \* FROM SALARIAȚI**  
**2        WHERE FUNCT='VÂNZATOR'**  
**3        OR SALA< 26000 AND CODS=1000 ;**

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1111	AVRAM ION	VÂNZATOR	100000	21200	1000	1000
1222	BARBU DAN	VÂNZATOR	120000	20750	2000	1000
3500	DAN ION	VÂNZATOR	160000	24500	3550	2500
2650	VLAD ION	VÂNZATOR	120000	25060	3500	1000

*4 records selected*

11) Să se selecteze MARCA și NUMELE pentru înregistrările care conțin date despre angajații a căror funcție este cea de șef de depozit sau

care au un salariu de 35.000 u.m. și lucrează în subordinea superiorului cu codul 1000.

```
SQL> SELECT MARCA, NUME
2      FROM SALARIAȚI
3      WHERE FUNCT='SEF DEP' OR
4      (SALA=35000 AND CODS=1000) ;
```

MARCA	NUME
1000	COMAN RADU
2500	VLAD VASILE

*2 records selected.*

12) Să se selecteze datele salariaților care lucrează în subordinea superiorului cu marca 1000 și au funcția șef de depozit sau salariul în valoare de 35.000 u.m.

```
SQL> SELECT * FROM SALARIAȚI
2      WHERE (FUNCT='SEF DEP' OR SALA=35000)
3      AND CODS=1000 ;
```

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1000	COMAN RADU	ȘEF DEP	130000	35000	2500	1000

*1 record selected.*

13) Să se afișeze valorile coloanelor MARCA, NUME, FUNCT privind angajații care lucrează în subordinea superiorului cu marca 1000 și au funcția de șef de depozit sau de vânzător.

```
SQL> SELECT MARCA, NUME, FUNCT, CODS
2      FROM SALARIAȚI
3      WHERE (FUNCT='SEF DEP' OR
4      FUNCT='VANZATOR')
5      AND CODS=1000 ;
```

MARCA	NUME	FUNCT	CODS
1111	AVRAM ION	VÂNZATOR	1000
1222	BARBU DAN	VÂNZATOR	1000
1000	COMAN RADU	ȘEF DEP	1000
2650	VLAD ION	VÂNZATOR	1000

*4 records selected.*

14) Să se selecteze coloanele MARCA, NUME, FUNCT pentru salariații care au funcția de șef depozit sau pentru cei care lucrează în subordinea superiorului cu marca 1000 și au funcția de vânzător.

```
SQL> SELECT MARCA, NUME, FUNCT, CODS
2      FROM SALARIAȚI
3      WHERE FUNCT='SEF DEP'
```



4      **OR (FUNCT='VÂNZATOR'**  
 5      **AND CODS=1000) ;**

MARCA	NUME	FUNCT	CODS
1111	AVRAM ION	VÂNZATOR	1000
1222	BARBU DAN	VÂNZATOR	1000
1000	COMAN RADU	ŞEF DEP	1000
2500	VLAD VASILE	ŞEF DEP	2500
2650	VLAD ION	VÂNZATOR	1000

*5 records selected.*

15) Să se selecteze toate datele privind angajații ce nu au funcția de vânzător.

**SQL> SELECT \* FROM SALARIAȚI**  
**2      WHERE NOT (FUNCT= 'VANZATOR') ;**

MARCA	NUME	FUNCT	CODD SALA	VENS	CODS
1000	COMAN RADU	ŞEF DEP	130000 35000	2500	1000
2500	VLAD VASILE	ŞEF DEP	160000 36500	1500	2500

*2 records selected.*

16) Să se selecteze valorile coloanelor MARCA, NUME, FUNCT, SALA+VENS pentru angajații care au salariul cuprins între 24.500 și 36.000 u.m.

**SQL>SELECT MARCA, NUME, FUNCT, SALA+VENS**  
**2      FROM SALARIAȚI**  
**3      WHERE SALA BETWEEN 24500 AND 36000 ;**

MARCA	NUME	FUNCT	SALA+VENS
1000	COMAN RADU	ŞEF DEP	37500
3500	DAN ION	VÂNZATOR	28050
3700	MANU DAN	VÂNZATOR	30000
2650	VLAD ION	VÂNZATOR	28560

*4 records selected.*

17) Să se selecteze câmpurile NUME, FUNCT, SALA+VENS pentru salariații care au salariul mai mic decât 24500 și mai mare decât 36000.

**SQL> SELECT NUME, FUNCT, SALA+VENS**  
**2      FROM SALARIAȚI**  
**3      WHERE SALA NOT BETWEEN 24500 AND 36000 ;**

NUME	FUNCT	SALA+VENS
AVRAM ION	VÂNZATOR	22200
BARBU DAN	VÂNZATOR	22750
VLAD VASILE	ŞEF DEP	38000

*3 records selected.*

18) Să se selecteze câmpurile MARCA, NUME și FUNCT pentru salariații care lucrează în depozitul cu codurile 130000 sau 160.000.

```
SQL> SELECT MARCA, NUME, FUNCT
2      FROM SALARIAȚI
3      WHERE CODD IN (130000,160000) ;
```

MARCA	NUME	FUNCT
1000	COMAN RADU	ȘEF DEP
3500	DAN ION	VÂNZATOR
2500	VLAD VASILE	ȘEF DEP
3700	MANU DAN	VÂNZATOR

4 records selected

19) Să se selecteze câmpurile MARCA, NUME, SALA, VENS pentru salariații care au altă funcție decât cea de vânzător.

```
SQL> SELECT MARCA, NUME, SALA, VENS
2      FROM SALARIAȚI
3      WHERE FUNCT NOT IN ('VÂNZATOR') ;
```

MARCA	NUME	FUNCT	SALA	VENS
1000	COMAN RADU	ȘEF DEP	35000	2500
2500	VLAD VASILE	ȘEF DEP	36500	1500

2 records selected

**Operatorul LIKE** face posibilă realizarea unor cereri de regăsire a înregistrărilor ce conțin un câmp a cărui valoare este comparată cu un șir de caractere dat. Pentru a evalua expresiile logice în care apar șiruri de caractere se poate folosi clauza **LIKE** în următoarea secvență:

```
SELECT ...
WHERE coloană LIKE șir ...;
```

Avantajul unei viteze mari de regăsire ca urmare a indexării este pierdut în momentul în care se caută într-o coloană indexată un șir care începe cu “-” sau “%”. Aceste caractere speciale suplinesc unul, respectiv mai multe caractere.

### Exemple:

1) Să se selecteze coloanele NUME și FUNCT precum și veniturile totale pentru salariații al căror nume începe cu litera C.

```
SQL> SELECT NUME, FUNCT, SALA+VENS
2      FROM SALARIAȚI
3      WHERE NUME LIKE 'C%' ;
```

NUME	FUNCT	SALA+VENS
------	-------	-----------

COMAN RADU

ŞEF DEP 37500

*1 record selected*

- 2) Să se selecteze coloanele NUME, FUNCT, SALA+VENS pentru salariații al căror nume se termină cu litera N.

```
SQL> SELECT NUME, FUNCT, SALA+VENS  
2 FROM SALARIAȚI  
3 WHERE NUME LIKE '%N';
```

NUME	FUNCT	SALA+VENS
AVRAM ION	VÂNZATOR	22200
BARBU DAN	VÂNZATOR	22750
DAN ION	VÂNZATOR	28050
MANU DAN	VÂNZATOR	30000
VLAD ION	VÂNZATOR	28560

*5 records selected*

- 3) Să se selecteze coloanele MARCA, NUME, MARCA, FUNCT pentru salariații al căror nume este format din nouă caractere (inclusiv spațiu), pe ultima poziție fiind N.

```
SQL> SELECT NUME, MARCA, FUNCT, SALA+VENS  
2 FROM SALARIAȚI  
3 WHERE NUME LIKE '_______N';
```

MARCA	NUME	FUNCT
1111	AVRAM ION	VÂNZATOR
1222	BARBU DAN	VÂNZATOR

*2 recors selected*

- 4) Să se selecteze salariații al căror nume are pe poziția a treia litera M.

```
SQL> SELECT NUME, FUNCT  
2 FROM SALARIAȚI  
3 WHERE NUME LIKE '__M%';
```

NUME	FUNCT
COMAN RADU	ŞEF DEP

*1 record selected*

- 5) Să se selecteze salariații al căror nume are o lungime de nouă caractere.

```
SQL> SELECT NUME  
2 FROM SALARIAȚI  
3 WHERE NUME LIKE '_____' ;
```

MARCA	NUME	NS
1111	AVRAM ION	
1222	BARBU DAN	

2 records selected

Regăsirea unor înregistrări ce conțin câmpuri cu valori nule se face cu ajutorul operatorului NULL.

**Exemple:**

1) Să se selecteze datele pentru salariații ce nu au venit suplimentar.

```
SQL> SELECT MARCA, NUME, FUNCT
2      FROM SALARIATI
3      WHERE VENS IS NULL ;
```

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
3770	CARMEN ANCA	VÂNZATOR	130000	26500	4000	

1 record selected

2) Să se selecteze toate datele salariaților care sunt șef depozit și al căror câmp VENS (venituri suplimentare) nu conține valoarea NULL.

```
SQL> SELECT * FROM SALARIATI WHERE
2      VENS IS NOT NULL AND FUNCT IS 'SEF DEP' ;
```

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1000	COMAN RADU	ŞEF DEP	130000	35000	2500	1000
2500	VLAD VASILE	ŞEF DEP	160000	36500	1500	2500

2 records selected

#### 6.4. Ordonarea liniilor rezultate în urma unei cereri

Limbajul SQL\*Plus are posibilitatea ordonării crescătoare sau descrescătoare a liniilor rezultate în urma unei cereri. Această operație se realizează prin intermediul secvenței:

```
SELECT ...
FROM ...
WHERE ...
ORDER BY {expr | număr-poziție} [ASC | DESC], ...;
```

unde:

*expr* reprezintă o expresie care face referire la una sau mai multe coloane; *număr-poziție* este un număr care identifică poziția coloanei din comanda **SELECT**, după care se dorește sortarea. Utilizarea operatorului de tipul **UNION**, **INTERSECT** sau **MINUS** impune prezența argumentului *număr-poziție*. **ASC** sau **DESC** precizează modul

de ordonare *ascendent*, respectiv *descendent*. În cazul în care clauzele **ORDER BY** și **DISTINCT** sunt utilizate împreună, clauza **ORDER BY** trebuie să se refere la *coloane* care n-au fost menționate în comanda **SELECT**.

**Exemple:**

1) Să se selecteze toate coloanele tabeli SALARIAȚI privind salariații care sunt vânzători și pentru care marca superiorului este 1000. Afișarea să se facă după valorile coloanei MARCA, descrescător.

```
SQL>SELECT * FROM SALARIAȚI
2   WHERE CODS=1000
3   AND FUNCT='VÂNZATOR'
4   ORDER BY MARCA DESC ;
```

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
2650	VLAD ION	VÂNZATOR	120000	25060	3500	1000
1222	BARBU DAN	VÂNZATOR	120000	20750	2000	1000
1111	AVRAM ION	VÂNZATOR	100000	21200	1000	1000

3 records selected

2) Să se selecteze crescător după salariu, angajații cu funcția vânzător pentru care marca superiorului este 1000.

```
SQL>SELECT * FROM SALARIAȚI
2   WHERE CODS=1000
3   AND FUNCT='VÂNZATOR'
4   ORDER BY SALA ;
```

MARCA	NUME	FUNCT	CODD	SALA	VENS	CODS
1222	BARBU DAN	VÂNZATOR	120000	20750	2000	1000
1111	AVRAM ION	VÂNZATOR	100000	21200	1000	1000
2650	VLAD ION	VÂNZATOR	120000	25060	3500	1000

3 records selected.

3) Să se selecteze crescător, după salariu, coloanele MARCA, NUME, SALA, VENS, SALA+VENS pentru acei salariați cu funcția de vânzător și pentru care marca superiorului este 1000

```
SQL> SELECT MARCA, NUME, SALA, VENS
2   FROM SALARIAȚI
3   WHERE CODS=1000 AND FUNCT='VÂNZATOR'
4   ORDER BY SALA ;
```

MARCA	NUME	SALA	VENS	SALA+VENS
1222	BARBU DAN	20750	2000	22750
1111	AVRAM ION	21200	1000	22200
2650	VLAD ION	25060	3500	28560

3 records selected.

4) Să se selecteze coloanele MARCA, NUME, CODD, VENS ordonate crescător după codul depozitului și veniturile suplimentare.

```
SQL> SELECT MARCA, NUME, CODD, VENS
2      FROM SALARIAȚI
3      WHERE FUNCT='VÂNZATOR'
4      ORDER BY CODD, VENS;
```

MARCA	NUME	CODD	VENS
1111	AVRAM ION	100000	1000
1222	BARBU DAN	120000	2000
2650	VLAD ION	120000	3500
3700	MANU DAN	160000	2500
3500	DAN ION	160000	3550

5 records selected

### 6.5. Selecții din mai multe tabele

Operația prin care se selectează și se grupează coloanele din tabele diferite, în scopul obținerii unor informații coerente, poartă numele de joncțiune (JOIN). Pentru a realiza o *joncțiune* și pentru a preciza corespondența între rândurile tabelor, se utilizează următoarea secvență:

```
SELECT ...
      FROM nume-tab1, nume-tab2, ...
WHERE condiție ...;
```

unde:

*condiție* reprezintă orice expresie care compară câmpurile diferitelor tabele. Vor fi selectate rândurile pentru care condiția este îndeplinită. În condiție poate fi folosit operatorul (+) care simulează existența unor rânduri vide în tabelele pentru care nu se găsesc corespondențe.

#### Exemple:

1) Din tabela COMENZI (creată anterior de utilizator, pe structura NrCom, DataCom, CodP, Cant, PrețMax, PretMin, Pret(plătit), să se selecteze codul produsului, cantitatea și diferența dintre valoarea mărfurilor comandate la prețul maxim și cel efectiv negociat (pentru un câmp definit DIFERENTA). Ordonarea să se facă ascendent după CODP.

```
SQL> SELECT COMENZI.CODP, CANT,
2      PRETMAX*CANT-PRET*CANT DIFERENTA
```

```

3      FROM COMENZI,PRETURI
4      WHERE PRETURI.CODP=COMENZI.CODP
5      ORDER BY COMENZI.CODP ;

```

CODP	CANT DIFERENTA	
13333	4	8000
14444	6	3000
16686	15	75000

3 records selected.

2) Să se afișeze din tabela COMENZI codul produsului, diferența dintre valoarea mărfurilor comandate la prețul maxim și cel efectiv negociat (pentru DIF\_MAX), diferența dintre valoarea mărfurilor comandate la prețul minim și cel efectiv negociat (pentru DIF\_MIN) selectate după criteriul egalității EQUI-JOIN (JOIN pe condiție de egalitate). Liniile vor fi ordonate crescător după valoarea totală a comenzii.

```

SQL> SELECT COMENZI.CODP,
2      PRETMAX*CANT-*PRET*CANT DIF_MAX,
3      PRETMIN*CANT-PRET*CANT DIF_MIN
4      FROM COMENZI.PRETURI
5      WHERE PRETURI.CODP=COMENZI.CODP
6      ORDER BY PRET*CANT;

```

CODP	DIF_MAX	DIF_MIN
14444	3000	0
13333	8000	-4000
16666	75000	-15000

3 records selected.

3) Să se afișeze câmpurile CODP, DENP, STOC și CANT, utilizând criteriul egalității OUTER-JOIN, pe câmpul comun CODP (se afișează datele despre acele produse pentru care există comenzi dar nu sunt în tabela Produse). Liniile vor fi ordonate crescător după câmpul CODP.

```

SQL> SELECT COMENZI.CODP, DENP, STOC, CANT
2      FROM PRODUSE, COMENZI
3      WHERE COMENZI.CODP=PRODUSE.CODP (+)
4      ORDER BY COMENZI.CODP ;

```

CODP	DENP	STOC	CANT
13333	CANAPEA A7	6	4
14444	SCAUN D4	36	6
16666	PLACAJ 2/2	100	15

3 records selected.

4) Să se afișeze în modul distinct (fără a se repeta aceleași linii), Codul depozitului și funcțiile care conțin valori nule în câmpul veniturilor

suplimentare. Selectarea să se facă pe criteriul egalității pe câmpul comun CODD și în condițiile în care acel depozit există.

```
SQL> SELECT DISTINCT DEPOZITE.CODD, NUME,
FUNCT
2 FROM SALARIATI, DEPOZITE
3 WHERE SALARIATI.CODD=DEPOZITE.CODD (+)
4 AND VENS IS NULL;
```

CODD	NUME	FUNCT
130000	CARMEN ANCA	VÂNZATOR

1 record selected.

5) Să se afișeze în modul distinct codul depozitului și funcțiile care conțin valori nule în câmpul veniturilor suplimentare. Selectarea se face pe criteriul egalității în câmpul comun CODD și în condițiile în care acel depozit există. Liniile vor fi ordonate descrescător după codul depozitului (din tabela SALARIAȚI). Se va adăuga în tabela salariați un nou tuplu cuprinzând datele vânzătorului Alexe Ioan, cod depozit 160000 și fără venituri suplimentare, pentru eficiența ordonării.

```
SQL> SELECT DISTINCT DEPOZITE.CODD, FUNCT
2 FROM SALARIATLDEPOZITE
3 WHERE SALARIATLCODD=DEPOZITE.CODD(+)
4 AND VENS IS NULL
5 ORDER BY SALARIATI.CODD DESC
```

CODD	NUME	FUNCT
160000	ALEXE IOAN	VÂNZĂTOR
130000	CARMEN ANCA	VÂNZATOR

2 records selected

Într-o cerere se pot înlocui numele de tabele sau coloane prin etichete.

6) Să se selecteze câmpurile NUME, FUNCT, DEND din tabelele DEPOZITE (pentru care este utilizat numele D), CodD, DenD, Capac, NrSal, și SALARIAȚI (pentru care este utilizat numele S). Liniile vor fi ordonate crescător după câmpul NUME.

```
SQL> SELECT NUME, FUNCT, D.CODD, DEND
2 FROM DEPOZITE D, SALARIAȚI S
3 WHERE D.CODD=S.CODD
4 ORDER BY NUME
```

NUME	FUNCT	CODD	DEND
ALEXE IOAN	VÂNZATOR	160000	SPORT
AVRAM ION	VÂNZATOR	100000	MOBILA
BARBU DAN	VÂNZATOR	120000	ALIMENTAR



CARMEN ANCA	VÂNZATOR	130000	AUTO
COMANRADU	SEF DEP	130000	AUTO
DAN ION	VÂNZATOR	160000	SPORT
DORU DAN	SEF DEP	130000	AUTO
FRINCUI ION	SEF DEPR	160000	SPORT
RADU IOANA	VÂNZATOR	130000	AUTO
SANDU ION	VÂNZATOR	130000	AUTO
VLAD ION	VÂNZATOR	120000	ALIMENTAR
VLAD VASILE	SEF DEP	160000	SPORT

13 records selected.

Pentru a pune în evidență posibilitatea schimbării denumirilor de tabele, în exemplele care urmează se va folosi, cu preponderență, adresarea prin calificare.

7) Să se selecteze, în condițiile redenumirii tabelelor COMENZI cu CONTRACT și PRODUSE cu COS, a coloanelor CODP, CANT, PREȚ din CONTRACT și coloanelor DENP, STOC din COS. Să fie selectate doar câmpurile care au unitatea de măsură "buc". Liniile vor fi ordonate crescător după valorile comenzilor.

```
SQL> SELECT CONTRACT.CODP, CONTRACT.CANT,
2      CONTRACT.PRET, DENP, COS.STOC
3      FROM COMENZI CONTRACT, PRODUSE COS
4      WHERE CONTRACT.CODP=COS.CODP
5      AND UM='BUC'
6      ORDER BY CONTRACT.CANT*COS.PRET
```

CODP	CANT	PREȚ	DENP	STOC
14444	6	4500	SCAUN D4	36
13333	4	80000	CANAPEA A	76

2 records selected.

8) Să se selecteze, în condițiile redenumirii tabelelor COMENZI cu CONTRACT și PRODUSE cu COS, a coloanelor CODP, CANT, PREȚ din CONTRACT și a coloanelor DENP, STOC din COS. Vor fi selectate doar câmpurile care au unitatea de măsură "buc" iar liniile vor fi ordonate crescător după valorile comenzilor. La afișare se va extrage o coloană numită 'DIFERENȚA' care să reflecte stocul rămas în urma onorării comenzii.

```
SQL> SELECT CONTRACT.CODP,
2      CONTRACT.CANT, CONTRACT.PRET,
3      DENP, COS.STOC,
4      COS.STOC-CONTRACT.CANT DIFERENȚA
5      FROM COMENZI CONTRACT, PRODUSE COS
6      WHERE CONTRACT.CODP=COS.CODP
7      AND UM="BUC"
```

## 8 ORDER BY CONTRACT.CANT \* CONTRACT.PRET

CODP	CANT	PREȚ	DENP	STOC	DIFERENȚA
14444	6	4500	SCAUN D4	36	30
13333	4	80000	CANAPEA A7	6	2

2 records selected.

9) Din tabela COMENZI (definită prin etichetele T1 și T2) să se selecteze CODP, CODC, în condițiile în care valoarea comenzii este mai mare ca 50.000 u.m. (JOIN-ul unei tabele pe ea însăși).

```
SQL> SELECT DISTINCT T1.CODP, T2.CODC
2 FROM COMENZI T1,COMENZI T2
3 WHERE T1.CANT * T2.PRET >60000
```

CODP	CODC
13333	121111
14444	121111
16666	121111
22222	121111

4 records selected

10) Să se selecteze campurile CODP, DENP, PREȚ, PRETMAX, PRETMIN pentru produsele al căror preț (negociat) este cuprins între prețul maxim și prețul minim. Ordonarea să se facă crescător după câmpul CODP.

```
SQL> SELECT PRODUSE.CODP, DENP,
2 COMENZI.PRET,
3 PRETMIN,PRETMAX
4 FROM PRODUSE, PRETURI, COMENZI
5 WHERE COMENZI.PRET BETWEEN PRETMIN
AND PRETMAX
6 ORDER BY PRODUSE.CODP
```

CODP	DENP	PREȚ	PRETMIN	PRETMAX
13333	CANAPEA A7	80000	79000	82000
14444	SCAUN D4	4500	4500	5000
16666	PLACAJ 2/2	25000	24000	30000

3 records selected.

## 6.6. Realizarea cererilor incluse

**Subcererile** reprezintă cereri incluse în clauzele unor comenzi SQL. Rândurile selectate de o subcerere nu sunt afișate, ele fiind utilizate în continuare de o comandă SQL.

Dacă subcererea este folosită în partea dreaptă a unei expresii logice sau a unei expresii de atribuire, ea va returna o singură valoare sau o coloană de valori. Compunerea valorii rezultate cu cea din stânga expresiei se face în conformitate cu operatorul care face legătura între cele două părți.

În cazul în care subcererea este folosită pentru a specifica valori în comenzi ca INSERT, CREATE TABLE, UPDATE, ea va returna câte o valoare pentru fiecare coloană specificată în comandă. Clauze ca ORDER BY, FOR nu pot fi folosite în subcereri.

Subcererile apar, în general, în următoarele comenzi:

```
COPY [FROM nume-utilizator/parolă@bază-de-date]  
[TO nume-utilizator/parolă@bază-de-date]  
{APPEND | CREATE | INSERT | REPLACE}  
Tabelă (col1,col2,...) USING cerere;...;
```

```
CREATE TABLE tabelă ...  
AS cerere ;
```

```
INSERT INTO tabelă | [(col1,col2,...)]  
[VALUES (val1,val2,...) | cerere];
```

```
UPDATE tabelă [sinonim]  
SET (col1, col2, ...) = (cerere)  
[WHERE condiție];
```

Există și *subcereri corelate* cu cererile din comanda principală, care apar doar în clauza WHERE a comenzii SELECT. Ele pot utiliza sinonime pentru tabela precizată în comanda SELECT și sunt evaluate câte o dată pentru fiecare rând selectat în comanda principală. Subcererile corelate pot apare în formule ca:

```
1). SELECT  
coloana1, coloana2,...  
FROM tabela1, tabela2 tab2,...  
WHERE coloana1 IN  
    (SELECT coloana1  
      FROM tabela1  
      WHERE condiție)  
.... ;
```

```

2) SELECT
coloana1, coloana2,...
FROM tabela1 tab1, tabela2 tab2,...
WHERE coloana2 IN
      (SELECT funcție (coloană)
       FROM tabela2 tab2
       WHERE tab2.coloana=coloana)
.... ;

```

**Exemple:**

1) Să se selecteze câmpurile NUME și FUNCT ale salariaților cu funcția identică cu a lui RADU IOANA.

```

SQL> SELECT NUME, FUNCT, FUNCȚIE
2      FROM SALARIAȚI
3      WHERE FUNCT=
4      (SELECT FUNCT FROM SALARIAȚI
5      WHERE NUME='RADU IOANA');

```

NUME	FUNCȚIE
AVRAM ION	VÂNZATOR
BARBU DAN	VÂNZATOR
DAN ION	VÂNZATOR
MANU DAN	VÂNZATOR
VLAD ION	VÂNZATOR
SANDU ION	VÂNZATOR
CARMEN ANA	VÂNZATOR
RADU IOANA	VÂNZATOR
ALEXE IOAN	VÂNZATOR

9 records selected.

2) Să se selecteze, în modul distinct, valorile timpurilor MARCA, SALA, NUME, CODS ale angajaților care au salariul mai mare decât unul dintre subordonații superiorului cu codul 1000. Rezultatele sunt cerute ordonate descrescător, după valorile câmpului SALA.

```

SQL> SELECT DISTINCT MARCA, SALA, NUME, CODS
2      FROM SALARIAȚI
3      WHERE SALA> ANY
4      (SELECT SALA FROM SALARIAȚI
5      WHERE CODS=1000)
6      ORDER BY SALA DESC;

```

MARCA	SALA	NUME	CODS
2500	36500	VLAD VASILE	2500
3755	36500	DORU DAN	4000
2550	36000	FRINCU ION	2500
1000	35000	COMAN RADU	1000
8700	27500	MÂNU DAN	2500
8770	26500	CARMEN ANA	4000
8755	25700	ALEXE IOAN	2500
8760	25600	SANDU ION	4000

8650	25060	VLAD ION	1000
8600	24500	DAN ION	2500
1111	21200	AVRAM ION	1000

*11 records selected.*

3) Să se selecteze valorile câmpurilor MARCA, NUME, SALA, CODS ale angajaților care au salariul mai mare decât al oricărui salariat din subordinea angajatului cu codul 1000. Datele să fie ordonate descrescător după valorile câmpului MARCA.

```
SQL> SELECT MARCA, NUME, SALA, CODS
2 FROM SALARIAȚI
3 WHERE SALA>ALL
4 (SELECT SALA FROM SALARIAȚI
5 WHERE CODS=1000)
6 ORDER BY MARCA DESC;
```

MARCA	NUME	SALA	CODS
3755	DORU DAN	36500	4000
2550	FRINCU ION	36000	2500
2500	VLAD VASILE	36500	2500

*3 records selected.*

4) Să se selecteze câmpurile CODD, DEND, NRSAL pentru acele depozite care au numărul de salariați mai mare ca 5 și codul cuprins în intervalul 100000 și 130000.

```
SQL> SELECT CODD, DEND, NRSAL
2 FROM DEPOZITE
3 WHERE NRSAL>5 AND CODD IN
4 (SELECT CODD FROM DEPOZITE
5 WHERE CODD BETWEEN 100000 AND 130000);
```

CODD	DEND NRSAL
120000	ALIMENTAR 11

*1 record selected.*

5) Să se selecteze CODD, DEND, NRSAL pentru acele depozite care au numărul de salariați mai mare ca 5 și codul în afara intervalului 100000 și 130000.

```
SQL> SELECT CODD.DEND,NRSAL
2 FROM DEPOZITE
3 WHERE NRSAL>5 AND CODD NOT IN
4 (SELECT CODD FROM DEPOZITE
5 WHERE CODD BETWEEN 100000 AND 130000);
```

CODD	DEND NRSAL
140000	TEXTILE 7

*1 record selected.*

6) Să se selecteze următoarele informații: marca, funcția și veniturile totale ale salariaților care au funcția și salariul lui VLAD VASILE.

```
SQL> SELECT MARCA, NUME, SALA+VENS
2      FROM SALARIAȚI
3      WHERE (FUNCT, SALA)=
4      (SELECT FUNCT,SALA FROM SALARIAȚI
5      WHERE NUME=' VLAD VASILE ');
```

MARCA	NUME	SALA+VENS
1500	VLAD VASILE	38000
3755	DORU DAN	42000

*2 records selected.*

7) Să se selecteze MARCA, NUME, SALA+VENS pentru acei angajați care au funcția lui VLAD VASILE sau salariul mai mare sau egal cu cel pe care îl are RADU IOANA. Ordonarea este crescătoare după valorile câmpului NUME.

```
SQL> SELECT MARCA, NUME, SALA+VENS
2      FROM SALARIAȚI
3      WHERE FUNCT IN
4      (SELECT FUNCT FROM SALARIAȚI
5      WHERE NUME='VLAD VASILE')
6      OR SALA>=
7      (SELECT SALA FROM SALARIAȚI
8      WHERE NUME='RADU IOANA')
9      ORDER BY NUME ;
```

NUME	FUNCT	SALA+VENS
ALEXE IOAN	VÂNZATOR	22200
AVRAM ION	VÂNZATOR	22750
BARBU DAN	VÂNZATOR	22200
CARMEN ANCA	VÂNZATOR	23456
COMANRADU	SEF DEP	37500
DAN ION	VÂNZATOR	24850
DORU DAN	SEF DEP	42000
FRINCU ION	SEF DEP	73000
MANU DAN	VÂNZATOR	30000
RADU IOANA	VÂNZATOR	23750
SANDU ION	VÂNZATOR	25600
VLAD ION	VÂNZATOR	28560
VLAD VASILE	SEF DEP	38000

*13 records selected.*

Dacă se dorește evitarea afișării valorilor NULE ale câmpului sumă venituri totale (SALA+VENS), atunci se va proceda astfel:

```

SQL> SELECT MARCA, NUME, SALA+VENS
2 FROM SALARIAȚI
3 WHERE FUNCT IN
4 (SELECT FUNCT FROM SALARIAȚI
5 WHERE NUME='VLAD VASILE')
6 OR SALA>=
7 (SELECT SALA FROM SALARIAȚI
8 WHERE NUME='RADU IOANA')
9 AND VENS IS NOT NULL
10 ORDER BY NUME ;

```

8) Să se selecteze câmpurile NUME și CODD ale angajaților cu codul superiorului 1000 și care au aceeași Funcție cu DORU DAN. Să se afișeze numai salariații pentru care există corespondență de CODD în tabelele DEPOZITE și SALARIAȚI. Datele să fie ordonate crescător după valorile câmpurilor CODD și NUME.

```

SQL> SELECT NUME, DEPOZITE.CODD
2 FROM SALARIAȚI, DEPOZITE
3 WHERE CODS = 1000
4 AND DEPOZITE.CODD=SALARIAȚI.CODD
5 AND FUNCT IN
6 (SELECT FUNCT FROM SALARIAȚI
7 WHERE NUME='DORU DAN')
8 ORDER BY DEPOZITE.CODD, NUME;

```

NUME	CODD
COMAN RADU	130000

1 record selected.

9) Să se selecteze câmpurile MARCA, NUME, CODD, DEND, SALA+VENS pentru angajații care au salariul mai mare decât media salariilor realizate în depozitul din care fac ei parte.

```

SQL> SELECT MARCA, NUME, SALARIAȚI.CODD,
2 DEND, SALA+VENS
3 FROM SALARIAȚI, DEPOZITE
4 WHERE SALA>
5 (SELECT AVG(SALA) FROM SALARIAȚI
6 WHERE SALARIAȚI.CODD=DEPOZITE.CODD)
7 AND DEPOZITE.CODD=SALARIAȚI.CODD
8 ORDER BY MARCA;

```

NUME	CODD	DEND	SALA+VENS
COMAN RADU	130000	AUTO	37500

VLAD VASILE	160000	SPORT	38000
FRINCU ION	160000	SPORT	73000
DORU DAN	130000	AUTO	42000

## 6.7. Utilizarea expresiilor, funcțiilor, variabilelor sistem și pseudo-coloanelor în selectarea datelor

*Expresiile aritmetice* pot fi utilizate în comenzile de selectare a datelor. Ele se construiesc cu ajutorul operatorilor aritmetici, numerelor de coloane și constantelor.

### Exemple:

1) Să se selecteze câmpurile CODD, CODP, CODC, precum și CANT\*PRET denumit ca valoare totală, din tabela COMENZI, pentru toate înregistrările al căror cod de depozit este 100000.

```
SQL> SELECT CODD, CODP, CODC, CANT*PRET
2      FROM COMENZI, DEPOZITE
3      WHERE CODD=100000 ;
```

CODD	CODP	CODC	VALOARE TOTALĂ
100000	13333	121111	320000
100000	14444	121111	27000
100000	16666	121111	375000
100000	22222	121111	282000

4 records selected

2) Să se selecteze codurile produselor și data până la care prețurile sunt admise, pentru produsele din tabela COMENZI care au prețul negociat mai mare decât prețul mediu stabilit.

```
SQL> SELECT DISTINCT PRETURI.CODP, DATASF
2      FROM PRETURI, COMENZI
3      WHERE COMENZI.PRET > (PRETMIN+PRETMAX)/2
```

CODP	DATASF
14444	01-NOV-05
11111	30-AUG-05
12222	30-SEP-05
16666	01-NOV-05
13333	01-GCT-05

5 records selected.

3) Să se afișeze numele, marca, raportul VENS/SALA și veniturile totale pentru șefii de depozite. Ordonarea datelor să fie făcută crescător după valorile raportului menționat.

```
SQL> SELECT NUME, MARCA, VENS/SALA, VENS+SALA
```



```

2      FROM SALARIATI
3      WHERE FUNCT='SEF DEP'
4      ORDER BY VENS/SALA DESC ;

```

NUME	MARCA	VENS/SALA	VENS+SALA
FRINCU ION	2550	1.0277777777777778	73000
DORU DAN	3755	.15068493150684932	42000
COMAN RADU	1000	.07142857142857143	37500
VLAD VASILE	2500	.0410958904109589	38000

4 records selected

4) Să se selecteze numele, codul depozitului, venitul total lunar și anual prognozat din tabela SALARIAȚI, pentru vânzători. Ordonarea să fie făcută crescător după valorile câmpului NUME.

```

SQL> SELECT      NUME,CODD,      SALA+VENS,
(SALA+VENS)*12

```

```

2      FROM SALARIAȚI
3      WHERE FUNCT='VÂNZATOR'
4      ORDER BY NUME;

```

NUME	CODD	SALA+VENS	(SALA+VENS)*12
ALEXE IOAN	160000	26500	318000
AVRAM ION	100000	22200	266400
BARBU DAN	120000	22750	273000
CARMEN ANCA	130000	26500	318000
DAN ION	160000	24850	298200
MANU DAN	160000	30000	360000
RADU IOANA	130000	23750	285000
SANDU ION	130000	25600	307200
VLAD ION	120000	28560	342720

9 records selected

O expresie aritmetică în care un operand este nul (valoarea NULL) are o valoare nulă (NULL). De aceea, de multe ori, construirea corectă a expresiilor aritmetice presupune transformarea valorii NULL într-o altă valoare, eventual nulă față de operația aritmetică (cum ar fi, de exemplu, zero la adunare, unu la înmulțire etc.).

Se presupune că în câmpul VENS s-au introdus și valori NULL. Pentru astfel de valori, expresia SALA+VENS are valoarea NULL, indiferent ce valoare are SALA, ceea ce matematic nu este corect. De aceea, pentru exemplul anterior, se recomandă ca valoarea NULL a câmpului VENS să fie transformată în zero (fiind o sumă) înaintea evaluării expresiei aritmetice.

### **Exemple:**

1) Să se selecteze coloanele NUME, MARCA, VENS, SALA+VENS din tabela SALARIAȚI, în condițiile în care codul superiorului este 1000.

```
SQL> SELECT NUME, MARCA, VENS, SALA+VENS  
2      FROM SALARIAȚI  
3      WHERE CODS= 1000 ;
```

NUME	MARCA	VENIS	SALA+VENS
AVRAM ION	1111	1000	22200
BARBU DAN	1222	2000	22750
COMAN RADU	1000	2500	36000
VLAD ION	2650		

Deoarece venitul suplimentar al salariatului VLAD ION este NULL, venitul lui total (SALA+VENS) a rezultat tot NULL. Utilizînd funcția NVL, ca în exemplul de mai jos, VENITUL\_TOTAL este acum calculat corect (pentru toți salariații).

```
SQL> SELECT NUME,  
2      SALA+NVL(VENS,0) VENIT_TOTAL  
3      FROM SALARIAȚI  
4      ORDER BY NUME ;
```

NUME	VENIT_TOTAL
ALEXE IOAN	25700
AVRAM ION	22200
BARBU DAN	22750
CARMEN ANCA	26500
COMAN RADU	35000
DAN ION	24850
DORU DAN	42000
FRINCU ION	73000
MANU DAN	30000
RADU IOANA	23750
SANDU ION	25600
VLAD ION	28560
VLAD VASILE	38000

*13 records selected.*

În capul de tabel, rezultat în urma cererilor, apar numele coloanelor din baza de date. În locul acestora pot fi afișate *etichete de coloană* declarate în comenzile de definire a tabelelor.

Sintaxa de declarare este:

### **SELECT**

**coloana1 nume-etichetă1, coloana2 nume-etichetă2, ...;**

Notățiile pentru coloane cu etichete au fost utilizate și în exemplele anterioare, cum ar fi cele pentru "Diferența", "Valoare Totală" sau "Diferență MIN sau MAX" etc.

**Exemplu:**

Să se selecteze denumirea depozitelor, codul acestora și numărul de salariați ce își desfășoară activitatea în cadrul lor, ordonate crescător după denumire.

```
SQL> SELECT DEND "PROFILUL",  
2      CODD "CODUL",  
3      NRSAL "NUMĂR DE SALARIAȚI"  
4      FROM DEPOZITE  
5      ORDER BY DEND;
```

PROFILUL	CODUL	NUMĂR DE SALARIAȚI
ALIMENTAR	120000	11
AUTO	130000	5
MOBILA	100000	3
SPORT	160000	4
TEXTILE	140000	7

Funcțiile aritmetice sunt utilizate în cererile de selecție.

**Exemple:**

1) Să se afișeze codul, denumirea și cantitatea - ridicată la pătrat, pentru produsele cu unitatea de măsură 'BUC'.

```
SQL> SELECT CODP, CODUL,  
2      DENP, DENUMIRE,  
3      POWER (CANT,2) CANT_PATRAT  
4      FROM PRODUSE  
5      WHERE UM='BUC'  
6      ORDER BY CODP;
```

CODUL	DENUMIRE	CANT_PATRAT
14444	SCAUN D4	1296
11111	MESE 15/20	49
12222	FOTOLIU A3	144
13333	CANAPEA A7	36

*4 records selected.*

2) Să se selecteze în modul distinct câmpurile CODP, DENP, CODC.DENC și să se calculeze câmpul  $[(CANT*PRET)/2]$ , rotunjit la două zecimale.

```
SQL> SELECT DISTINCT  
2      COMENZI.CODP, DENP,  
3      COMENZI.CODC, DENC,
```

```

4      ROUND ( (COMENZI.CANT*COMENZI.PRET)/2,2 )
ROTUNJIRE
5      FROM COMENZI, CLIENTI, PRODUSE
6      WHERE COMENZI.CODP = PRODUSE.CODP
7      AND COMENZI.CODC = CLIENTI.CODC;

```

CODP	DENP	CODC	DENC	ROTUNJIRE
13333	CANAPEA A7	121111	UNIT-2	160000
16666	PLACAJ 2/2	121111	UNIT-2	187500
14444	SCAUN D4	121111	UNIT-2	13500

*3 records selected.*

*Funcțiile caracter* operează asupra șirurilor de caractere. Ele se utilizează pentru transformarea literelor mari în mici sau invers, extragerea unui subșir dintr-un șir, începând cu o anumită poziție, selectarea cuvintelor care se pronunță asemănător cu un șir dat etc.

**Exemple:**

1) Să se afișeze o situație finală prin care să fie redată câmpurile NUME și MARCA angajatului, reunite într-un câmp comun denumit "INFORMAȚIE", iar câmpul venituri totale anuale să fie denumit "VENIT\_ANUAL". Selecția este cerută pentru angajații cu codul superiorului egal cu 1000.

```

SQL>SELECT      NUME      ||      '-'      ||      MARCA
INFORMAȚIE
2      (SALA+VENS)*12  VENIT_ANUAL
3      FROM SALARIAȚI
4      WHERE CODS =1000;

```

INFORMAȚIE	VENIT_ANUAL
AVRAM ION - 1111	266400
BARBU DAN - 1222	273000
COMAN RADU - 1000	450000
VLAD ION - 2650	342720

*4 records selected.*

2) Să se selecteze câmpurile NUME și FUNCT din tabela SALARIAȚI și să se atribuie un cod de clasificare fiecărei funcții. Codul este format dintr-o singură cifră și are valorile: 1 pentru vânzător, 2 pentru director, 3 pentru restul funcțiilor.

```

SQL> SELECT NUME, FUNCT,
2      DECODE(FUNCT, 'VÂNZATOR',1,'DIRECTOR',2,3)
3      CLASIFIC_FUNCT
4      FROM SALARIAȚI
5      ORDER BY FUNCT, NUME ;

```

NUME	FUNCT CLASIFIC_FUNCT
------	----------------------

COMAN RADU	SEF DEP	3
DORU DAN	SEF DEP	3
FRINCU ION	SEF DEP	3
VLAD VASILE	SEF DEP	3
ALEXE IOAN	VÂNZATOR	1
AVRAM ION	VÂNZATOR	1
BARBU DAN	VÂNZATOR	1
CARMEN ANCA	VÂNZATOR	1
DAN ION	VÂNZATOR	1
MANU DAN	VÂNZATOR	1
RADU IOANA	VÂNZATOR	1
SANDU ION	VÂNZATOR	1
VLAD ION	VÂNZATOR	1

13 records selected

3) Să se afișeze salariul și veniturile suplimentare ale tuturor angajaților, cu specificarea numelui numai pentru salariații vânzători. Pentru restul angajaților să se afișeze mesajul: \*\*\* Nu interesează \*\*\*. Ordonarea să se facă după funcție, în mod descrescător.

```
SQL> SELECT DECODE
2      (
3      FUNCT,'VÂNZATOR',      NUME,      '***Nu
interesează***'
4      ) "NUMELE",
5      SALA, NVL (VENS, 0)
6      FROM SALARIAȚI
7      ORDER BY FUNCT DESC;
```

NUMELE	SALA	VENS
AVRAM ION	21200	1000
BARBU DAN	20750	2000
DAN ION	24500	350
MANU DAN	27500	2500
VLAD ION	25060	3500
SANDU ION	25600	0
CARMEN ANCA	26500	0
RADU IOANA	20750	3000
ALEXE IOAN	25700	0
*** Nu interesează ***	35000	2500
*** Nu interesează ***	36500	1500
*** Nu interesează ***	36000	37000
*** Nu interesează ***	36500	5500

13 records selected.

4) Să se afișeze primele 5 caractere din NUME, MARCA și primul caracter din funcție, pentru toți angajații.

```
SQL> SELECT      SUBSTR (NUME, 1, 5) 5_din_Nume,
2      MARCA, Marca_Sal,
3      SUBSTR (FUNCT, 1, 1) 1_din_Funcție
4      FROM SALARIAȚI;
```

5_din_Nume	Marca_Sal	1_din_Funcție
AVRAM	1111	V
BARBU	1222	V
COMAN	1000	S
DAN I	3500	V
VLAD	2500	S
MANU	3700	V
FRINC	2550	S
VLAD	2650	V
DORU	3755	S
SANDU	3760	V
CARME	3770	V
RADU	1680	V
ALEXE	3755	V

*13 records selected.*

5) Să se selecteze și afișeze numele, funcția și salariul total pentru toți angajații care au numele terminat cu litera N. Situația trebuie aibă următoarea formă: Persoana\_cu\_Funcția SALA+VENS NUME SALARIAT - funcție

```
SQL> SELECT NUME || '-' || LOWER (FUNCT)
Persoana_cu_Funcția,
2 SALA+VENS SAL_TOTAL
3 FROM SALARIAȚI
4 WHERE
5 UPPER (NUME) LIKE ' %N '
```

Persoana_cu_Funcția	SAL_TOTAL
AVRAM ION-vânzător	22200
BARBU DAN- vânzător	22750
DAN ION- vânzător	24850
MĂNU DAN- vânzător	30000
FRINCUION-sefdep	73000
VLAD ION- vânzător	28560
DORUDAN-sefdep	42000
SANDU ION- vânzător	25600
ALEXE IOAN-vânzator	

5) Să se selecteze câmpurile CODC, DENC, STR și NR din tabela CLIEȚI, pentru clienții cu cifra 1 pe prima poziție a contului lor.

```
SQL> SELECT CODC, DENC, STR, NR
2 FROM CLIEȚI WHERE
3 INSTR (CONT,'1',1)=1 ;
```

CODC	DENC	STR	NR
121111	UNIT-1	Moșilor	104
211111	UNIT-2	Dorobanți	18

6) Să se afișeze numele și marca acelor angajați al căror nume se pronunță asemănător cu DORU DAN.

```
SQL> SELECT NUME, MARCA  
2 FROM SALARIAȚI  
3 WHERE SOUNDEX (NUME) = SOUNDEX ('DORU  
DAN');
```

NUME	MARCA
DORU DAN	3755
DORU DANIEL	5565

*2 records selected.*

7) Să se selecteze din tabela SALARIAȚI coloanele NUME și MARCA, pentru angajații a căror funcție este asemănătoare fonetic cu șirul de caractere: 'vânzător'.

```
SQL> SELECT NUME, MARCA  
2 FROM SALARIAȚI  
3 WHERE SOUNDEX (FUNCT)= SOUNDEX ('VÂNZATOR')
```

NUME	MARCA
AVRAM ION	1111
BARBU DAN	1222
DAN ION	3500
MANU DAN	3700

*4 records selected.*

8) Să se selecteze constanta "NUMELE SI CODUL CLIENȚILOR:" și valorile câmpurilor DENC, CODC pentru clienții din strada Calea Moșilor și cu un cod mai mare ca 100000.

```
SQL> SELECT "NUMELE SI CODUL CLIENȚILOR:",  
2 DENC, CODC  
3 FROM CLIENȚI  
4 WHERE STR LIKE 'MOȘILOR%'  
5 AND CODC> 100000 ;
```

NUMELE ȘI CODUL CLIENȚILOR:	DENC	CODC
NUMELE ȘI CODUL CLIENȚILOR:	UNIT-2	121111

Pentru afișarea câmpurilor de tip **dată calendaristică** sau pentru calcule în care sunt implicate aceste câmpuri, există funcții specifice.

**Exemple:**

1) Să se selecteze în modul distinct codurile și denumirile produselor, precum și a datei până la care prețurile actuale sunt admise. Ordonarea să se facă crescător după valorile cimpului CODP.

```
SQL> SELECT DISTINCT PRODUSE.CODP,
```

```

2  PRODUSE.DENP.DATASF
3  FROM PRETURI,PRODUSE
4  WHERE PRODUSE.CODP=PRETURI.CODP
5  ORDER BY PRETURI.CODP ;

```

CODP	DENP	DATASF
11111	MESE 15/20	30-AUG-05
12222	FOTOLIU A3	30-SEP-05
13333	CANAPEA A7	01-OCT-05
15555	BIROU C6X4	01-OCT-05
16668	PLACAJ 2/2	01-NOV-05
14444	SCAUN D4	01-NOV-05

*6 records selected.*

2) Să se selecteze câmpurile CODP și DATASF scriindu-se codul pe un rând și data pe următorul. Data va fi scrisă sub forma LL/ZZ-AA :

Se vor folosi operatorul TRUNC care asigură trecerea la rândul următor, funcția TO\_CHAR (expr [fmt] ) care efectuează conversia câmpului dată *expr* într-un șir de caractere în formatul specificat în *[fmt]*; comanda COLUMN care definește un alt format de afișare a coloanei DATASF.

```

SQL> COLUMN DATASF FORMAT A21 TRUNC
SQL> SELECT      CODP,
2  TO_CHAR (DATASF, 'MM/DD-YY') DATASF
3  FROM PRETURI;

```

CODP	DATASF
11111	08/30-05
12222	09/30-05
13333	10/01-05
15555	10/01-05
16666	11/01-05
14444	11/01-05

*6 records selected.*

3) Să se selecteze coloanele CODP, DATASF la produsele cu codul mai mare ca 13333, afișând ziua și luna în litere iar anul cu patru cifre.

```

SQL> COLUMN DATASF FORMAT A26
SQL> SELECT CODP,
2  TO_CHAR (DATASF, 'DAY MONTH YYYY')
DATASFÂRȘIT
3  FROM PRETURI
4  WHERE CODP>13333 ;

```

CODP	DATASFÂRȘIT	
15555	THURSDAY OCTOBER	2005
16666	SUNDAY NOVEMBER	2005
14444	SUNDAY NOVEMBER	2005



*3 records selected*

4) Să se selecteze CODP, DATASF la produsele cu codul mai mare ca 13333, afișând luna în litere și pentru an ultimele două cifre.

```
SQL> SELECT CODP,  
2 TO_CHAR (DATASF, 'DAY MONTH YY') DATASFÂRȘIT  
3 FROM PRETURI WHERE CODP> 13333 ;
```

CODP	DATASFÂRȘIT	
15555	THURSDAY OCTOBER	05
16666	SUNDAY NOVEMBER	05
14444	SUNDAY NOVEMBER	05

*3 records selected*

5) Să se selecteze coloanele CODP, DATASF la produsele cu codul mai mare ca 13333, afișând ziua în cifre, luna în litere și anul cu patru cifre.

```
SQL> SELECT CODP,  
2 TO_CHAR (DATASF, 'DD MONTH YYYY')  
DATASFÂRȘIT  
3 FROM PRETURI  
4 WHERE CODP> 13333 ;
```

CODP	DATASFÂRȘIT	
15555	01 OCTOBER	2005
16666	01 NOVEMBER	2005
14444	01 NOVEMBER	2005

*3 records selected*

6) Să se selecteze coloanele CODP, DATASF la produsele cu codul mai mare ca 13333, afișând primele trei litere de la zi, luna în litere și anul cu patru cifre.

```
SQL> SELECT CODP,  
2 TO_CHAR (DATASF, 'DY MONTH YYYY')  
DATASFÂRȘIT  
3 FROM PRETURI  
4 WHERE CODP> 13333 ;
```

CODP	DATASFÂRȘIT	
15555	THU OCTOBER	2005
16666	SUN NOVEMBER	2005
14444	SUN NOVEMBER	2005

*3 records selected*

7) Să se selecteze câmpurile CODP, DATASF la produsele cu codul mai mare ca 13333, afișând ziua în litere, luna în cifre și anul cu patru cifre.

```

SQL>SELECT CODP,
2      TO_CHAR (DATASF, 'DAY MM YYYY')
DATASFÂRȘIT
3      FROM PRETURI
4      WHERE CODP> 13333 ;

```

CODP	DATASFÂRȘIT
15555	THURSDAY 10 2005
16668	SUNDAY 11 2005
14444	SUNDAY 11 2005

*3 records selected.*

8) Să se selecteze câmpurile CODP, DATASF la produsele cu codul mai mare ca 13333, afișând primele trei litere pentru zi și lună și ultimele două cifre de la an.

```

SQL>SELECT CODP,
2      TO_CHAR (DATASF, 'DY-MON-YY') DATASFÂRȘIT
3      FROM PRETURI
4      WHERE CODP>13333 ;

```

CODP	DATASFÂRȘIT
15555	THU-OCT-05
16666	SUN-NOV-05
14444	SUN-NOV-05

*3 records selected.*

9) Să se selecteze câmpurile CODP, DATASF pentru produsele cu codul mai mare decât 13333, afișându-se primele trei litere de la zi, luna în litere și ultimele două cifre de la an.

```

SQL>SELECT CODP,
2      TO_CHAR (DATASF, 'DY-MONTH-YY')
DATASFÂRȘIT
3      FROM PRETURI
4      WHERE CODP>13333 ;

```

CODP	DATA SFÂRȘIT
16556	THU-OCTOBER -05
16666	SUN-NOVEMBER -05
14444	SUN-NOVEMBER -05

*3 records selected.*

10) Să se selecteze coloanele CODP, DATASF pentru toate produsele, afișând ziua în cifre urmate de sufixul *-th*, primele trei litere ale lunii și ultimele două cifre ale anului, despărțite prin *liniuță*.

```

SQL>SELECT CODP,

```

**2 TO CHAR (DATASF, 'DDth-MON-YY')**  
**DATASFÂRȘIT**

**3 FROM PRETURI ;**

CODP	DATASFÂRȘIT
11111	30TH-AUG-05
12222	30TH-SEP-05
13333	01ST-OCT-05
16555	01ST-OCT-05
16666	01ST-NOV-05
14444	01ST-NOV-05

*6 records selected.*

11) Să se selecteze din tabela PRETURI valorile câmpurilor CODP și DATASF. Data de sfârșit (DATASF) să se prezinte însoțită de timpul intern, exprimat în diverse forme de afișare.

**SQL>SELECT CODP,**  
**2 TO\_CHAR (DATASF, 'DDth-MON-YY HH:MIPM')**  
**DATASFÂRȘIT**

**3 FROM PRETURI ;**

CODP	DATASFÂRȘIT	
11111	30TH-AUG-05	12:00AM
12222	30TH-SEP-05	12:00AM
13333	01ST-OCT-05	12:00AM
15555	01ST-OCT-05	12:00AM
16666	01ST-NOV-05	12:00AM
14444	01ST-NOV-05	12:00AM

*6 records selected.*

Sau :

**SQL> SELECT CODP,**  
**2 TO\_CHAR (DATASF, 'DDth-MON-YY HH:MI')**  
**DATASFÂRȘIT**

**3 FROM PRETURI ;**

CODP	DATASFÂRȘIT	
11111	30TH-AUG-05	12:00
12222	30TH-SEP-05	12:00
13333	01ST-OCT-05	12:00
15555	01ST-OCT-05	12:00
16666	01ST-NOV-05	12:00
14444	01ST-NOV-05	12:00

*6 records selected.*

12) Să se afișeze câmpurile CODP, DATASF și data de sfârșit a valabilității unui preț (considerată la o distanță de 90 de zile față de DATASF).

**SQL> SELECT CODP, DATASF, DATASF+90**

**2. FROM PRETURI;**

CODP	DATASF	DATASF+90
11111	30-AUG-05	28-NOV-05
12222	30-SEP-05	29-DEC-05
13333	01-OCT-05	30-DEC-05
15555	01-OCT-05	30-DEC-05
16666	01-NOV-05	30-JAN-06
14444	01-NOV-05	30-JAN-06

*6 records selected*

13) Să se selecteze codul produsului, data maximă admisă de practicare a unui preț și data curentă pentru acele produse care îndeplinesc condiția ca DATASF+10 să fie mai mare decât SYSDATE.

**SQL>SELECT CODP, DATASF,**

**2 SYSDATE DATA\_CURENTĂ**

**3 FROM PRETURI**

**4 WHERE DATASF+10 > SYSDATE ;**

CODP	DATASF	DATA_CURENTĂ
12222	30-SEP-05	13-SEP-05
13333	01-OCT-05	13-SEP-05
15555	01-OCT-05	13-SEP-05
16666	01-NOV-05	13-SEP-05
14444	01-NOV-05	13-SEP-05

*5 records selected.*

14) Să se selecteze codul produsului, data de sfârșit, data curentă, valorile expresiilor TRUNC(DATASF+90) - TRUNC (SYSDATE) și DATASF+90. Selecția să se realizeze pentru produsele cu codul mai mare ca 13333 și data de sfârșit plus 90 de zile mai mare decât data curentă.

**SQL>SELECT CODP, DATASF,**

**2 SYSDATE DATA\_CURENTA,**

**3 TRUNC(DATASF+90)-TRUNC(SYSDATE)**

**DIFERENȚA**

**4 DATASF+90 DATA\_90**

**5 FROM PRETURI**

**6 WHERE DATASF+90 > SYSDATE**

**7 AND CODP>13333 ;**

CODP	DATASF	DATA_CURENTA	DIFERENȚA	DATA_90
15555	01-OCT-05	02-OCT-05	89	30-DEC-05
16666	01-NOV-05	02-OCT-05	120	30-JAN-06
14444	01-NOV-05	02-OCT-05	120	30-JAN-06

*3 records selected.*

Operațiile de calcul cu data calendaristică sunt posibile în cadrul unei comenzi de selecție. Structura afișării câmpurilor rezultate se poate stabili prin comanda COLUMN.

15) Să se selecteze și afișeze coloanele CODP, DATASF, RDATE și RSDATE (utilizând funcțiile NEXT\_DAY și LAST\_DAY).

```
SQL>COLUMN    DATASF    FORMAT A21
SQL>COLUMN    RDATE     FORMAT A21
SQL>COLUMN    RSDATE    FORMAT A20
SQL>SELECT CODP,
TO_CHAR (DATASF, 'DAY MON - YY') DATASFÂRȘIT
TO_CHAR (NEXT_DAY (DATASF+90, 'VINERI'), 'DAY
MON - YY') RDATE
TO_CHAR (LAST_DAY (DATASF+90), 'DAY MON -
YY') RSDATE
FROM PREȚURI ;
```

CODP	DATASF	RDATE	RSDATE
11111	WEDNESDAY AUG-05	VINERI NOV-05	WEDNESDAY NOV-05
12222	FRIDAY SEP-05	VINERI DEC-05	SATURDAY DEC-05

2 records selected

O operație posibil de realizat este și cea de a aduna algebric un număr de luni la o dată calendaristică: funcția ADD\_MONTHS (d,n), care adună *n* luni la data *d*.

16) Să se afișeze codul produsului, data de sfârșit și a unui nou termen de valabilitate a unui preț dat, calculat prin adăugarea a trei luni. Să se selecteze doar produsele al căror cod este mai mic decât 13333, iar data de sfârșit este mai mare decât data curentă plus trei luni.

```
SQL>SELECT CODP, DATASF,
2      ADD_MONTHS (DATASF,3) RDATE
3      FROM PREȚURI
4      WHERE
5      DATASF > SYSDATE+90
6      AND CODP<13333 ;
```

CODP	DATASF	RDATE
11111	30-AUG-05	30-NOV-05
12222	30-SEP-05	30-DEC-05

2 records selected

17) Să se selecteze codul, denumirea produselor și data de sfârșit, pentru acele produse care îndeplinesc următoarele condiții: data de sfârșit este cuprinsă în intervalul: 30/Aug/05 - 1/Oct/05; prețurile sunt mai mari ca 30.000 u.m.. Datele să se ordoneze crescător după codul produsului.

**SQL>SELECT DISTINCT PRODUSE.CODP, DENP, DATASF**

**2 FROM PRETURI, PRODUSE**

**3 WHERE**

**4 DATASF BETWEEN '30-AUG-05' AND '01-OCT-05'**

**5 AND PRET > 30000**

**6 AND PRODUSE.CODP=PRETURI.CODP**

**7 ORDER BY PRODUSE.CODP ;**

CODP	DENP	DATASF
12222	FOTOLIU A3	30-SEP-05
13333	CANAPEA A7	01-OCT-05
15555	BIROU C6X4	01-OCT-05

*3 records selected*

#### **6.8.Utilizarea funcțiilor de grupare și a clauzei GROUP BY în selectarea datelor**

Rezultatele obținute în urma selectărilor pot fi grupate cu ajutorul clauzei *GROUP BY*. Secvența utilizată pentru această operație este:

**SELECT ...**

**GROUP BY tabelă.coloana1, tabelă.coloană2,...**

**HAVING condiție**

**...;**

Prin parcurgerea secvenței se obține câte un rând pentru înregistrările care au aceleași valori în coloanele specificate în clauza *GROUP BY*. Prezența clauzei *HAVING* determină obținerea acelor grupuri care îndeplinesc condițiile specificate. Este de reținut faptul că *GROUP BY* și *HAVING* trebuie să fie declarate după clauzele *WHERE*, *CONNECT BY* și *START WITH*, în cazul când acestea există în comandă.

#### **Exemple:**

1) Să se selecteze din tabela *PRETURI* valorile din coloana *DATASF* și să se contorizeze numărul aparițiilor acestora.

```

SQL> SELECT TO_CHAR (DATASF, ' "DATA - " DD MON
YYYY')
2 DATA_MAXIMA,
3 COUNT (*) NUMAR_PRODUSE
4 FROM PRETURI
5 GROUP BY TO_CHAR (DATASF, ' "DATA - " DD
MON YYYY');

```

DATA_MAXIMA	NUMAR_PRODUSE
DATA - 01 NOV 2005	2
DATA - 01 OCT 2005	2
DATA - 30 AUG 2005	1
DATA - 30 SEP 2005	1

2) Să se selecteze și afișeze valoarea medie zilnică a comenzilor ce trebuie onorate în perioada 01-30 Iulie 2005

```

SQL> SELECT AVG (CANT*PRET) MEDIA
2 FROM COMENZI
3 WHERE
4 DATAL >'01-JUL-05'AND DATAL < '30-JUL-05';

```

MEDIA
25100

3) Să se afișeze valoarea totală a salariilor și veniturilor suplimentare pentru salariații cu funcția vânzător.

```

SQL> SELECT SUM (SALA) TOTAL_SAL ,
2 SUM (VENS) TOTAL_VEN
3 FROM SALARIAȚI
4 WHERE FUNCT = 'VÂNZATOR';

```

TOTAL_SAL	TOTAL_VEN
268560	14750

4) Să se afișeze media anuală a veniturilor totale (SALA+VENS) pentru salariații cu funcția vânzător.

```

SQL> SELECT AVG (SALA+VENS)*12 MEDIA_ANUALA
FROM SALARIAȚI
WHERE FUNCT='VÂNZATOR';

```

MEDIA_ANUALA
308147

5) Să se afișeze valoarea maximă și minimă a salariului precum și diferența max-min. Selecția se face din tabela SALARIAȚI.

```
SQL> SELECT MAX (SALA) MAX_SAL,
2      MIN (SALA) MIN_SAL,
3      MAX (SALA)-MIN (SALA) DIF_MAX_MIN
4      FROM SALARIAȚI ;
```

MAX_SAL	MIN_SAL	DIF_MAX_MIN
45000	24250	25750

6) Să se determine lungimea maximă a șirurilor de caractere din coloana DENP a tabelului PRODUSE.

```
SQL> SELECT MAX (LENGTH (DENP)) LUNG_MAX_DENP
2 FROM PRODUSE ;
```

LUNG_MAX_DENP
10

7) Să se selecteze coloanele NUME, FUNCT, SALA+VENS din tabela SALARIAȚI pentru angajații care au salariul egal cu salariul maxim.

```
SQL> SELECT NUME, FUNCT, SALA+VENS
2      FROM SALARIAȚI
3      WHERE      SALA =
4      (SELECT MAX (SALA) FROM SALARIAȚI) ;
```

NUME	FUNCT	SALA+VENS
ION ION	DIRECTOR	85000

*1 record selected*

8) Să se selecteze coloanele DENP, CODP, CODD din tabela PRODUSE pentru care stocul existent este mai mare sau egal cu cantitatea comandată.

```
SQL> SELECT DENP, CODP, CODD
2      FROM PRODUSE
3      WHERE STOC >=
4      (SELECT SUM (CANT) FROM COMENZI);
```

DENP	CODP	CODD
PLACAJ 2/2	166666	100000
SCAUN D4	144444	100000
CANAPEA A7	133333	100000

9) Să se afișeze numărul de valori nenule înregistrate în coloana VENS din tabela SALARIAȚI.

```
SQL> SELECT COUNT (VENS)
2      FROM SALARIAȚI ;
```

COUNT (VENS)
11



10) Să se selecteze din tabela SALARIAȚI coloanele NUME, FUNCT, SALA+VENS pentru șefii de depozite care au salariul mai mare sau egal cu jumătate din salariul maxim.

```
SQL>SELECT NUME, FUNCT, SALA+NVL (VENS,0)  
2      FROM SALARIAȚI  
3      WHERE SALA >=  
4      (SELECT MAX (SALA)/2 FROM SALARIAȚI)  
5      AND FUNCT= 'SEF DEP' ;
```

NUME	FUNCT	SALA+NVL(VENS,0)
COMAN RADU	SEF DEP	37500
VLAD VASILE	SEF DEP	38000
FRINCUI ION	SEF DEP	73000
DORU DAN	SEF DEP	42000
PAUL STEFAN	SEF DEP	40600

11) Să se afișeze numărul de produse distincte care au ca unitate de măsură 'BUC'

```
SQL>SELECT COUNT(DISTINCT CODP) NR_PROD  
2      FROM PRODUSE  
3      WHERE UM = 'BUC';
```

NR_PROD
5

12) Să se afișeze numărul de subordonați ai salariatului cu marca 2500.

```
SQL> SELECT COUNT(*) NR_SUBORD  
FROM SALARIAȚI  
WHERE CODS=2500;
```

NR_SUBORD
3

13) Să se afișeze valoarea medie a salariilor și valoare medie a veniturilor suplimentare pentru fiecare depozit fie utilizînd comanda SELECT de mai multe ori, fie clauza GROUP BY o singură dată.

Folosind comanda SELECT:

Pentru salariații depozitului 100000 (codd=100000)

```
SQL> SELECT AVG(SALA), AVG(VENS)  
FROM SALARIAȚI  
WHERE CODD = 100000;
```

AVG(SALA)	AVG(VENS)
-----------	-----------

33100                      20500

Pentru salariații depozitului 120000 (codd=120000)

```
SQL>SELECT AVG(SALA), AVG(VENS)  
2 FROM SALARIAȚI  
3 WHERE CODD = 120000;
```

AVG(SALA)	AVG(VENS)
2402.5	1975

Pentru salariații depozitului 130000 (codd=130000)

```
SQL> SELECT AVG(SALA), AVG(VENS)  
2 FROM SALARIAȚI  
3 WHERE CODD = 130000;
```

AVG(SALA)	AVG(VENS)
28870	2750

Pentru salariații depozitului 160000 (codd=160000)

```
SQL> SELECT AVG(SALA), AVG(VENS)  
2 FROM SALARIAȚI  
3 WHERE CODD = 160000;
```

AVG(SALA)	AVG(VENS)
30866.7	9390

Folosind clauza GROUP BY:

```
SQL> SELECT CODD, AVG(SALA), AVG(VENS)  
2 FROM SALARIAȚI  
3 WHERE CODD = 160000;  
4 GROUP BY CODD ;
```

CODD	AVG(SALA)	AVG(VENS)
100000	33100	20500
120000	24202.5	1975
130000	28870	2750
160000	30866	9390

14) Să se calculeze media salariului pentru fiecare grup de angajați care au același superior.

```
SQL> SELECT CODS, AVG(SALA)  
2 FROM SALARIAȚI  
3 GROUP BY CODS ;
```

CODS	AVG (SALA)
1000	22336.7
2500	27083,3
2550	25500
3755	26600
4000	26050
7000	35750
8000	4500

15) Să se calculeze media salariului anual prognozat pentru salariații care au același superior și nu au funcția de șef depozit.

```
SQL> SELECT CODS,AVG(SALA)*12 MEDIE_SAL_ANUAL
2 FROM SALARIAȚI
3 WHERE FUNCT NOT IN ('SEF DEP')
4 GROUP BY CODS ;
```

CODS	MEDIE_SAL_ANUAL
1000	268040
2500	271500
2550	306000
3755	319200
4000	312600
8000	540000

16) Să se calculeze și afișeze valoarea medie a comenzilor pentru fiecare depozit.

```
SQL> SELECT AVG (CANT*PRET) VAL_MEDIE,
2 DEPOZITE.CODD COD_DEP
3 FROM COMENZI, DEPOZITE
4 WHERE COMENZI.CODD=DEPOZITE.CODD
5 GROUP BY DEPOZITE.CODD ;
```

VAL_MEDIE	COD_DEP
251000	100000

17) Să se calculeze și afișeze valoarea medie a comenzilor pe produse.

```
SQL> SELECT AVG (CANT*PRET) VAL_MEDIE,
2 PRODUSE.CODP COD_PROD
3 FROM PRODUSE, COMENZI
4 WHERE COMENZI.CODP=PRODUSE.CODP
5 GROUP BY PRODUSE.CODP ;
```

VAL_MEDIE	COD_PROD
320000	133333
27000	144444
375000	166666

18) Să se determine media salariului anual pentru fiecare funcție dacă există mai mult de doi salariați angajați pe aceeași funcție. Se vor afișa funcția, numărul de salariați cu funcția respectivă și media calculată.

```
SQL> SELECT FUNCT FUNCȚIE,
2      COUNT(*) NR_SAL,
3      AVG(SALA)*12 MEDIE_SAL
4      FROM SALARIAȚI
5      GROUP BY FUNCT
6      HAVING COUNT(*)>2 ;
```

FUNCȚIE	NR_SAL	MEDIE_SAL
SEF DEP	5	429600
VÂNZATOR	11	292975

2 records selected.

19) Să se selecteze codurile superiorilor (CODS) care au doi sau mai mulți subordonați.

```
SQL> SELECT CODS
2      FROM SALARIAȚI
3      WHERE FUNCT='VÂNZATOR'
4      GROUP BY CODS
5      HAVING COUNT(*)>=2 ;
```

CODS
1000
2500
2550
3755
4000

5 records selected.

20) Să se selecteze codurile superiorilor care au media veniturilor suplimentare ale subordonaților mai mare decât 10% din salariul mediu. Se vor afișa codurile superiorilor și media veniturilor totale anuale ale subordonaților.

```
SQL> SELECT CODS,
2      AVG (SALA+NVL (VENS,0) ) *12 MED_VEN_TOT_AN
3      FROM SALARIAȚI
4      GROUP BY CODS
5      HAVING AVG (VENS) > AVG (SALA)*0.10;
```

CODS	MED_VEN_TOT_AN
2500	486400
7000	474300
8000	1020000

21) Să se selecteze funcțiile pentru care salariile medii sunt mai mari decât salariul mediu al unui vânzător. Se vor afișa funcțiile și salariile medii pentru fiecare funcție.

```
SQL> SELECT FUNCT, AVG(SALA) MEDIE_SAL
2      FROM SALARIAȚI
3      GROUP BY FUNCT
4      HAVING AVG (SALA) >
5      (SELECT AVG(SALA) FROM SALARIAȚI
6      WHERE FUNCT= 'VÂNZATOR');
```

FUNCT	MEDIE_SAL
DIRECTOR	45000
SEF DEP	35800

22) Să se afișeze numărul salariaților din depozitul cu codul 100000 și să se determine câți dintre ei au venituri suplimentare.

```
SQL> SELECT COUNT (*) NR_SALARIAȚI_DEP_100000,
2      COUNT(VENS) NR_SAL_VEN_SUPL
3      FROM SALARIAȚI
4      WHERE CODS=100000 ;
```

NR_SALARIAȚI_DEP_100000	NR_SAL_VEN_SUPL
4	3

23) Să se determine numărul salariaților din depozitul cu codul 100000 precum și suma și media veniturilor lor suplimentare.

```
SQL> SELECT SUM (NVL (VENS,0)) SUMA_VEN_SUPL,
2      COUNT (NVL (VENS,0)) NR_SAL,
3      AVG (NVL (VENS,0)) MEDIA_VEN_SUPL
4      FROM SALARIAȚI
5      WHERE CODS=100000 ;
```

SUMA_VEN_SUPL	NR_SAL	MEDIA_VEN_SUPL
9000	4	2250

24) Pentru a afla numărul de angajați și salariul mediu anual în toate combinațiile posibile de departamente și funcții, se va folosi următoarea cerere:

```
SELECT      DECODE(GROUPING(ume_dept),      1,      'Toate
Departamentele', ume_dept) AS ume_dept,
```

```

DECODE(GROUPING(funcția), 1, 'Toate Funcțiile', funcția) AS
funcția,
COUNT(*) "Total Angajați",
AVG(sal) * 12 "Sal Mediu"
FROM ang, dept
WHERE dept.nume_dept = emp.nume_dept
GROUP BY CUBE (nume_dept, funcția);

```

Nume dept.	funcția	total angajați	sal mediu
CONTABILITATE	FUNCTIONAR	1	15600
CONTABILITATE	MANAGER	1	29400
CONTABILITATE	PRESEDINTE	1	60000
CONTABILITATE	Toate Funcțiile	3	35000
CERCETARE	ANALYST	2	36000
CERCETARE	FUNCTIONAR	2	11400
CERCETARE	MANAGER	1	35700
CERCETARE	Toate Funcțiile	5	26100
VANZARI	FUNCTIONAR	1	11400
VANZARI	MANAGER	1	34200
VANZARI	VANZATOR	4	16800
VANZARI	Toate Funcțiile	6	18800
Toate Departamentele	ANALYST	2	36000
Toate Departamentele	FUNCTIONAR	4	12450
Toate Departamentele	MANAGER	3	33100
Toate Departamentele	PRESEDINTE	1	60000
Toate Departamentele	VANZATOR	4	16800
Toate Departamentele	Toate Funcțiile	14	24878.5714

### 6.9. Operații pe tabele structurate arborescent

Limbajul SQL\*Plus permite explorarea *structurilor arborescente* existente în baza de date. Operația se realizează cu ajutorul clauzelor START WITH și CONNECT BY din comanda SELECT.

```

SELECT ...
FROM ...
CONNECT BY [PRIOR] col1 = [PRIOR] col2
START WITH col = valoare
...;

```

**Example:**

1) Să se afișeze câmpurile NUME, FUNCT, CODS, MARCA din tabela SALARIAȚI. Datele să fie ordonate crescător după codul superiorului.

**SQL>SELECT NUME, FUNCT, CODS, MARCA  
FROM SALARIAȚI  
ORDER BY CODS;**

NUME	FUNCT	CODS	MARCA
AVRAM ION	VÂNZĂTOR	1000	1111
BARBU DAN	VÂNZĂTOR	1000	1222
COMAN RADU	SEF DEP	7000	1000
VLAD ION	VÂNZĂTOR	1000	2650
AILENEI FLORIN	VÂNZĂTOR	2500	2553
DAN ION	VÂNZĂTOR	2500	3500
DARIAN GEO	VÂNZĂTOR	2500	2554
FRINCU ION	SEF DEP	2500	2550
RADU ION	VÂNZĂTOR	2500	1680
VLAD VASILE	SEF DEP	7000	2500
ALEXE IOAN	VÂNZĂTOR	3755	3759
MANU DAN	VÂNZĂTOR	3755	3700
DORU DAN	SEF DEP	7000	3755
CARMEN ANCA	VÂNZĂTOR	4000	3770
PAUL ȘTEFAN	SEF DEP	7000	4000
SANDU ION	VÂNZĂTOR	4000	3760
ION ION	DIRECTOR	7000	7000

2) Să se selecteze NUMELE, MARCA, codul superiorului și codul depozitului pentru subordonații direcți și indirecti ai salariatului cu numele DORU DAN.

**SQL> SELECT NUME, MARCA, CODS, CODD  
2 FROM SALARIAȚI  
3 CONNECT BY PRIOR MARCA = CODS  
4 START WITH NUME = 'DORU DAN';**

NUME	MARCA	CODS	CODD
DORU DAN	3755	7000	130000
MANU DAN	3700	3755	160000
ALEXE IOAN	3759	3755	160000

*3 records selected*

3) Să se afișeze câmpurile MARCA, NUME, FUNCT, CODS, CODD din tabela SALARIAȚI, ordonate crescător după marca și codul superiorului, pentru subordonații direcți și indirecti ai salariatului cu numele VLAD VASILE.

**SQL> SELECT MARCA, NUME, FUNCT, CODS, CODD  
2 FROM SALARIAȚI  
3 CONNECT BY PRIOR MARCA = CODS  
4 START WITH NUME = 'VLAD VASILE'**

### **5 ORDER BY MARCA, CODS;**

MARCA	NUME	FUNCT	CODS	CODD
1680	RADU ION	VÂNZĂTOR	2500	2553
2500	VLAD VASILE	SEF DEP	7000	3500
2550	FRINCUI ION	SEF DEP	2500	2554
2553	AILENEI FLORIN	VÂNZĂTOR	2550	2550
2554	DARIAN GEOR	VÂNZĂTOR	2550	2500
2556	DAN ION	VÂNZĂTOR	2500	2500
7000	ION ION	DIRECTOR	7000	7000

*7 records selected.*

4) Să se afișeze câmpurile MARCA, NUME, FUNCT, CODS, CODD din. tabela SALARIAȚI, ordonate crescător după marca.

```
SQL> SELECT MARCA,NUME,FUNCT,CODS,CODD  
2 FROM SALARIAȚI  
3 ORDER BY MARCA;
```

MARCA	NUME	FUNCT	CODS	CODD
1000	COMAN RADU	SEF DEP	7000	130000
1111	AVRAM ION	VINZATOR	1000	100000
1222	BARBU DAN	VINZATOR	1000	120000
1680	RADU ION	VINZATOR	2500	130000
2500	VLAD VASILE	SEF DEP	7000	160000
2550	FRINCUI ION	SEF DEP	2500	160000
2553	AILENEI FLORIN	VINZATOR	2550	120000
2554	DARIAN GEO	VINZATOR	2550	120000
2650	VLAD ION	VINZATOR	1000	120000
3500	DAN ION	VINZATOR	2500	120000
3700	MĂNU DAN	VINZATOR	3755	160000
3755	DORU DAN	SEF DEP	7000	130000
3759	ALEXE IOAN	VINZATOR	3755	160000
3760	SANDU ION	VINZATOR	4000	130000
3770	CARMEN ANA	VINZATOR	4000	130000
4000	PAUL ȘTEFAN	SEF DEP	7000	160000
7000	ION ION	DIRECTOR	7000	100000

*17 records selected.*

5) Să se selecteze numele, marca, codul superiorului, codul depozitului pentru subordonații direcți și indirecti ai salariatului cu numele DORU DAN.

```
SQL> SELECT NUME,MARCA,CODS,CODD  
2 FROM SALARIAȚI  
3 CONNECT BY PRIOR MARCA=CODS  
4 START WITH NUME='DORU DAN';
```

NUME	MARCA	CODS	CODD
DORU DAN	3755	7000	130000



MÂNU DAN	3700	3755	160000
ALEXE IOAN	3759	3755	160000

3 records selected.

6) Să se afișeze MARCA, NUME, FUNCT, CODS și CODD ordonate crescător după marcă și cod superior pentru subordonații direcți și indirecți ai salariatului cu numele VLAD VASILE.

```
SQL> SELECT MARCA,NUME,FUNCT,CODS,CODD
2 FROM SALARIAȚI
3 CONNECT BY PRIOR MARCA=CODS
4 START WITH NUME='VLAD VASILE'
5 ORDER BY MARCA.CODS
```

MARCA	NUME	FUNCT	CODS	CODD
1680	RADU ION	VINZATOR	2500	130000
2500	VLAD VASILE	SEF DEP	7000	160000
2550	FRINCU ION	SEF DEP	2500	160000
2553	AILENEI FLORIN	VINZATOR	2550	120000
2554	DARIAN GEO	VINZATOR	2550	120000
3500	DAN ION	VINZATOR	2500	160000

6 records selected.

7) Să se afișeze câmpurile NUME și MARCA pentru subordonații direcți și indirecți ai salariatului ION ION precum și nivelul lor de subordonare.

```
SQL> SELECT LEVEL,JWME.MARCA
2 FROM SALARIAȚI
3 CONNEOT BY PRIOR MARCA=CODS
4 START WITH NUME='ION ION';
```

LEVEL	NUME	MARCA
1	ION ION	7000
2	COMAN RADU	1000
3	AVRAM ION	1111
3	BARBU DAN	1222
3	VLAD ION	2650
2	VLAD VASILE	2500
3	FRINCU ION	2550
4	AILENEI FLORIN	2553
4	DARIAN GEO	2554
3	RADU ION	1680
3	DAN ION	3500
2	DORU DAN	3755
3	MÂNU DAN	3700
3	ALEXE IOAN	3759
2	PAUL ȘTEFAN	4000
3	SANDU ION	3760
3	CARMEN ANA	3770

8) Să se selecteze câmpurile LEVEL, NUME și MARCA aparținând subordonaților lui ION ION. Nivelul de subordonare va fi pus în evidență și prin afișarea deplasată a numelor salariaților subordonați față de numele superiorului corespunzător.

**SQL> COLUMN ORG.CHART FORMAT A21**

**SQL> SELECT LEVEL,**

**2 LPADC (' ',LEVEL\*2) || NUME NUME\_SI\_PRENUME,**

**3 MARCA**

**4 FROM SALARIAȚI**

**5 CONNECT BY PRIOR MARCA=CODS**

**6 START WITH NUME='ION ION»;**

LEVEL	NUME.SI.PRENUME	MARCA
1	ION ION	7000
2	COMAN RADU	1000
3	AVRAM ION	1111
3	BARBU DAN	1222
3	VLAD ION	2650
2	VLAD VASILE	2500
3	FRINCU ION	2550
4	AILENEI FLORIN	2553
4	DARIAN GEO	2554
3	RADU ION	1680
3	DAN ION	3500
2	DORU DAN	3755
3	MĂNU DAN	3700
3	ALEXEIOAN	3759
2	PAUL ȘTEFAN	4000
3	SANDU ION	3760
3	CARMEN ANA	3770

*17 records selected.*

9) Să se afișeze câmpurile LEVEL, NUME, MARCA aparținând subordonaților salariaților PAUL ȘTEFAN și COMAN RADU, punând în evidență, prin scriere decalată, modul de subordonare.

**SQL> SELECT LEVEL,**

**LPAD(' ',LEVEL\*2) || NUME**

**3 NUME\_SI\_PRENUME,MARCA**

**4 FROM SALARIAȚI**

**5 CONNECT BY PRIOR MARCA=CODS**

**6 START WITH NUME='PAUL ȘTEFAN' 7 OR  
NUME='COMAN RADU'**

LEVEL	NUME	MARCA
1	COMAN RADU	1000
2	AVRAM ION	1111
2	BARBU DAN	1222
2	VLAD ION	2650

1	PAUL ȘTEFAN	4000
2	SANDU ION	3760
2	CARMEN ANA	3770

*7 records selected.*

10) Să se afișeze câmpurile LEVEL, NUME și MARCA, în structură arborescentă, aparținând salariaților din subordinea celor cu aceeași funcție cu a salariatului COMAN RADU.

```
SQL> SELECT LEVEL,LPADC ('LEVEL*2') || NUME
2     NUME_SI_PRENUME, MARCA
3     FROM SALARIAȚI
4     CONNECT BY PRIOR MARCA=CODS
5     START WITH FUNCT IN
6     (SELECT FUNCT FROM SALARIAȚI
7     WHERE NUMES"COMAN RADU");
```

LEVEL	NUME_SI_PRENUME	MARCA
2	COMAN RADU	1000
3	AVRAM ION	1111
3	BARBU DAN	1222
3	VLAD ION	2650
2	VLAD VASILE	2500
3	FRINCUI ION	2550
4	AILENEI FLORIN	2553
4	DARIAN GEO	2554
3	RADU ION	1680
3	DAN ION	3500
2	DORU DAN	3755
3	MÂNU DAN	3700
3	ALEXEIOAN	3759
2	PAUL ȘTEFAN	4000
3	SANDU ION	3760
3	CARMEN ANA	3770

11) Să se selecteze câmpurile LEVEL, NUME, MARCA, FUNCT, CODD aparținând subordonaților angajaților care lucrează în același depozit cu RADU ION.

```
SQL> SELECT LEVEL,
2     LPADC (' ',LEVEL*2) || NUME
3     NUME_SI_PRENUME,
4     MARCA,FUNCT,CODD
5     FROM SALARIAȚI
6     CONNECT BY PRIOR MARCA=CODS
7     START WITH CODD IN
8     (SELECT CODD
9     FROM SALARIAȚI
10    WHERE NUME='RADU ION')
```

LEVEL	NUME, SI PRENUME	MARCA	FUNCT	CODD
1	RADU ION	1680	VINZATOR	130000
1	COMAN RADU	1000	SEF DEP	130000
2	AVRAM ION	1111	VINZATOR	100000
2	BARBU DAN	1222	VINZATOR	120000
2	VLAD ION	2650	VINZATOR	120000
1	DORU DAN	3755	SEF DEP	130000
2	MÂNU DAN	3700	VINZATOR	160000
2	ALEXE IOAN	3759	VINZATOR	160000
1	CARMEN ANA	3770	VINZATOR	130000
1	SANDU ION	3760	VINZATOR	130000

*10 records selected.*

12) Să se selecteze câmpurile NUME, MARCA, FUNCT, CODD aparținând superiorilor salariatului VLAD ION.

```
SQL> SELECT NUME,MARCA,CODS,FUNCT,CODD
2 FROM SALARIAȚI
3 CONNECT BY MARCA=PRIOR CODS
4 START WITH NUME='VLAD ION';
```

NUME	MARCA	CODS	FUNCT	CODD
VLAD ION	2650	1000	VINZATOR	120000
COMAN RADU 1000	7000	SEF	DEP	130000
ION ION	7000	8000	DIRECTOR	100000

*3 records selected.*

13) Să se afișeze câmpurile NUME, MARCA, FUNCT aparținând subordonaților salariatului VLAD VASILE, mai puțin datele salariatului AILENEI FLORIN.

```
SQL> SELECT NUME,MARCA,FUNCT
2 FROM SALARIAȚI
3 WHERE NUME != 'AILENEI FLORIN'
4 CONNECT BY PRIOR MARCA=CODS
5 START WITH NUME='VLAD VASILE';
```

NUME	MARCA	FUNCT
VLAD VASILE	2500	SEF DEP
FRINCU ION	2550	SEF DEP
DARIAN GEO	2554	VINZATOR
RADU ION	1680	VINZATOR
DAN ION	3500	VINZATOR

14) Să se afișeze o serie de date despre subordonații salariatului ION ION, mai puțin datele despre VLAD VASILE și despre salariații din subordinea lui.

```
SQL> SELECT MARCA,LEVEL,NUME,PUNCT,CODS
2 FROM SALARIAȚI
```

```

3   CONNECT BY PRIOR MARCA=CODS
4   AND NUME <> 'VLAD VASILE'
5   START WITH NUME='ION ION';

```

MARCA	LEVEL	NUME	FUNCT	CODS
7000	1	ION ION	DIRECTOR	8000
1000	2	COMAN RADU	SEF DEP	7000
1111	3	AVRAM ION	VINZATOR	1000
1222	3	BARBU DAN	VINZATOR	1000
2650	3	VLAD ION	VINZATOR	1000
3755	2	DORU DAN	SEF DEP	7000
3700	3	MÂNU DAN	VINZATOR	3755
3759	3	ALEXE IOAN	VINZATOR	3755
4000	2	PAUL ȘTEFAN	SEF DEP	7000
3760	3	SANDU ION	VINZATOR	4000
3770	3	CARMEN ANA	VINZATOR	4000

*11 records selected*

15) Să se afișeze o serie de date ale salariaților subordonați lui ION ION, mai puțin datele salariaților COMAN RADU și VLAD VASILE precum și ale subordonaților lui VLAD VASILE.

```

SQL> SELECT MARCA,NUME,LEVEL,FUNCT,
2 CODS.CODD
3 FROM SALARIAȚI
4 WHERE NUME <> 'COMAN RADU'
5 CONNECT BY PRIOR MARCA=CODS
6 AND NUME != 'VLAD VASILE'
7 START WITH NUME='ION ION'

```

MARCA	NUME	LEVEL	FUNCT	CODS
7000	ION ION	1DIRECTOR	8000	100000
1111	AVRAM ION	3 VINZATOR	1000	100000
1222	BARBU DAN	3 VINZATOR	1000	120000
2650	VLAD ION	3 VINZATOR	1000	120000
3755	DORU DAN	2 SEF DEP	1000	130000
3700	MÂNU DAN	3 VINZATOR	3755	160000
3759	ALEXE IOAN	3 VINZATOR	3755	160000
4000	PAUL ȘTEFAN	2 SEF DEP	7000	160000
3760	SANDU ION	3 VINZATOR	4000	130000
3770	CARMEN ANA	3 VINZATOR	4000	130000

*10 records selected.*

16) Să se afișeze datele salariaților subordonați direct salariatului ION ION.

```

SQL> SELECT MAJRCA,NUME,LEVEL,
2 FUNCT,CODS,CODD

```

**3 FROM SALARIAȚI**  
**4 WHERE LEVEL=2**  
**5 CONNECT BY PRIOR MARCA=CODS**  
**6 START WITH NUME='ION ION'**

MARCA	NUME	LEVEL	FUNCT	CODS	CODD
1000	COMAN RADU	2	SEF DEP	7000	130000
2500	VLAD VASILE	2	SEF DEP	7000	160000
3755	DORU DAN	2	SEF DEP	7000	130000
4000	PAUL ȘTEFAN	2	SEF DEP	7000	160000

*4 records selected.*

CAPITOLUL 6. SELECTAREA DATELOR DIN TABELELE BAZEI DE DATE.....	1
6.1. Comanda SELECT.....	1
6.2. Utilizarea clauzei FROM .....	3
6.3 Utilizarea operatorilor în formularea condițiilor de selecție din clauza WHERE .....	5
6.3.Ordonarea liniilor rezultate în urma unei cereri.....	12
6.4. Selecții din mai multe tabele .....	14
6.5. Realizarea cererilor incluse .....	18
6.6. Utilizarea expresiilor, funcțiilor, variabilelor sistem și pseudo-coloanelor în selectarea datelor .....	24
6.7.Utilizarea funcțiilor de grupare și a clauzei GROUP BY în selectarea datelor.....	38
6.8. Operații pe tabele structurate arborescent .....	46