

LABORATOR 4 SQL**GRUPAREA DATELOR. FUNCȚII MULTIPLE-ROW (GRUP SAU AGREGAT)**

- Funcțiile grup pot fi utilizate pentru a întoarce informația corespunzătoare fiecăruia dintre grupurile obținute în urma divizării liniilor tabelului cu ajutorul clauzei GROUP BY.
 - Server-ul Oracle aplică aceste funcții fiecărui grup de linii și întoarce un singur rezultat pentru fiecare mulțime.
 - Pot apărea în clauzele SELECT, ORDER BY și HAVING.
 - Exemple de funcții grup: AVG, SUM, MAX, MIN, COUNT etc.
 - Tipurile de date ale argumentelor funcțiilor grup pot fi CHAR, VARCHAR2, NUMBER sau DATE.
 - Funcțiile AVG și SUM operează numai asupra valorilor numerice.
 - Funcțiile MAX și MIN pot opera asupra valorilor numerice, caracter sau dată calendaristică.
 - **Toate funcțiile grup, cu excepția funcției COUNT(*), ignoră valorile *null*.**
 - COUNT(expresie) întoarce numărul de linii pentru care expresia dată nu are valoarea *null*.
 - Funcția COUNT întoarce un număr mai mare sau egal cu zero și nu întoarce niciodată valoarea *null*.
 - Când este utilizată clauza GROUP BY, server-ul sortează implicit mulțimea rezultată în ordinea crescătoare a valorilor coloanelor după care se realizează gruparea.
 - Absența clauzei GROUP BY conduce la aplicarea funcției grup pe mulțimea tuturor liniilor tabelului.
 - În clauza GROUP BY se trec obligatoriu toate coloanele prezente în clauza SELECT, care nu sunt argument al funcțiilor grup.
1. Afișați cel mai mare salariu, cel mai mic salariu, suma și media salariilor tuturor angajaților. Etichetați coloanele Maxim, Minim, Suma, respectiv Media.


```
SELECT MIN(salary) min, MAX(salary) max,  
       SUM(salary) suma, ROUND(AVG(salary)) media  
FROM   employees;
```
 2. Utilizând funcția grup COUNT determinați:
 - a. numărul total de angajați;
 - b. numărul de angajați care au manager;
 - c. numărul de manageri.
 3. Afișați diferența dintre cel mai mare și cel mai mic salariu. Etichetați coloana "Diferenta".
 4. Obțineți numărul de angajați din departamentul având codul 50.
 5. Câți angajați din departamentul 80 câștigă comision?

6. Determinați valoarea medie și suma salariilor pentru toți angajații care sunt reprezentanți de vânzări (codul jobului este SA_MAN, SA_REP).
7. a. Selectați data angajării primei persoane care a fost angajată de companie (ținând cont și de istoricul angajaților).
b. Selectați numele persoanelor care au fost angajate primele în companie.

8. Afișați numărul de angajați pentru fiecare job.

```
SELECT job_id, COUNT(employee_id) nr_angajati
FROM employees
GROUP BY job_id;
```

9. Afișați minimul, maximul, suma și media salariilor pentru fiecare departament.

10. Afișați codul departamentului și media salariilor pentru fiecare job din cadrul acestuia.

```
SELECT department_id, job_id, AVG(salary)
FROM employees
GROUP BY department_id, job_id;
```

11. a. Afișați codul departamentelor pentru care salariul minim depășește 5000\$.

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) > 5000;
```

- b. Modificați cererea anterioară astfel încât să afișați numele acestor departamente.

- c. Modificați cererea anterioară astfel încât să afișați și orașul în care funcționează departamentele.

12. Obțineți codul departamentelor și numărul de angajați al acestora pentru departamentele în care lucrează cel puțin 10 angajați.

13. Câte departamente au cel puțin 10 angajați?

```
SELECT COUNT(COUNT(employee_id))
FROM employees
GROUP BY department_id
HAVING COUNT(*) >= 10;
```

14. Obțineți codul departamentelor și suma salariilor angajaților care lucrează în acestea, în ordine crescătoare după suma salariilor. Se consideră angajații care au comision și departamentele care au mai mult de 5 angajați.

15. Determinați numele angajaților care au mai avut cel puțin două joburi.

16. Afișați job-ul pentru care salariul mediu este minim.

```

SELECT job_id
FROM employees
GROUP BY job_id
HAVING AVG(salary) = (SELECT MIN(AVG(salary))
                      FROM employees
                      GROUP BY job_id);

```

17. Afișați cel mai mare dintre salariile medii pe departamente.

18. a. Afișați codul, numele departamentului și suma salariilor pe departamente.

```

SELECT d.department_id, department_name, a.suma
FROM departments d, (SELECT department_id ,SUM(salary) suma
                     FROM employees
                     GROUP BY department_id) a
WHERE d.department_id =a.department_id;

```

b. Dați o altă metodă de rezolvare a acestei probleme.

19. a. Scrieți o cerere pentru a afișa numele departamentului, numărul de angajați și salariul mediu pentru angajații din acel departament. Coloanele vor fi etichetate Departament, Nr. angajati, Salariu Mediu.

```

SELECT department_name "Departament ",
       (SELECT COUNT(employee_id)
        FROM employees
        WHERE department_id = d.department_id ) "Nr. angajati",
       (SELECT AVG(salary)
        FROM employees
        WHERE department_id = d.department_id) "Salariu mediu"
FROM departments d;

```

b. Dați o altă metodă de rezolvare pentru problema anterioară.

20. a. Scrieți o cerere pentru a afișa job-ul, salariul total pentru job-ul respectiv pe departamentele 10, 20 și 30, respectiv salariul total pentru job-ul respectiv pe toate cele 3 departamente. Etichetați coloanele corespunzător. Datele vor fi afișate în următoarea formă:

Job	Dep10	Dep20	Dep30	Total
J1	10	5	25	40
J2	15		10	25

```

SELECT DISTINCT job_id,
       (SELECT SUM(salary)
        FROM employees
        WHERE job_id=e.job_id AND department_id =10
        GROUP BY job_id) dep10,

```

```

(SELECT SUM(salary)
FROM   employees
WHERE  job_id=e.job_id AND department_id =20
GROUP BY job_id) dep20,
(SELECT SUM(salary)
FROM   employees
WHERE  job_id=e.job_id AND department_id =30
GROUP BY job_id) dep30,
(SELECT SUM(salary)
FROM   employees
WHERE  job_id=e.job_id AND department_id IN (10, 20,30)
GROUP BY job_id) total
FROM   employees e;

```

- b. Clauzele GROUP BY din subcererile anterioare sunt necesare?
- c. Este necesară o clauză GROUP BY în cererea principală?
- d. Clauza SELECT a cererii principale trebuie să utilizeze opțiunea DISTINCT?
- e. Dați o altă metodă de rezolvare utilizând funcția DECODE.

Indicație: SUM(DECODE(department_id,10,salary))

21. Să se creeze o cerere prin care să se afișeze numărul total de angajați și, din acest total, numărul celor care au fost angajați în 1997, 1998, 1999 și 2000. Datele vor fi afișate în forma următoare:

1997	1998	1999	2000	Total
10	5	25	10	50

Indicație: SUM(DECODE(TO_CHAR(hire_date,'yyyy'),1997,1,0))

Operatorii ROLLUP și CUBE

- Clauza *GROUP BY* permite gruparea liniilor selectate după valorile expresiilor precizate în aceasta. Pentru fiecare grup, va fi întoarsă o singură linie de informație.
- Clauza *GROUP BY* poate produce grupări superagregat utilizând extensiile *CUBE* sau *ROLLUP*.
- **ROLLUP** grupează liniile selectate pe baza valorilor primelor n , $n - 1$, ..., 0 expresii din specificația *GROUP BY* și întoarce o singură linie pentru fiecare grup.
 - **ROLLUP** creează grupări prin deplasarea într-o singură direcție, de la dreapta la stânga, de-a lungul listei de coloane specificate în clauza **GROUP BY**. Apoi, se aplică funcția agregat acestor grupări. Dacă sunt specificate n expresii în operatorul **ROLLUP**, numărul de grupări generate va fi $n+1$. Liniile care se bazează pe valoarea primelor n expresii se numesc linii obișnuite, iar celelalte se numesc linii superagregat.
 - **GROUP BY ROLLUP (expr_1, expr_2, ..., expr_n)** generează $n+1$ tipuri de linii, corespunzătoare următoarelor grupări:

```

GROUP BY (expr_1, expr_2, ..., expr_n-1, expr_n)
GROUP BY (expr_1, expr_2, ..., expr_n-1)
...
GROUP BY (expr_1, expr_2)
GROUP BY (expr_1)
GROUP BY () - corespunzător absenței clauzei GROUP BY și
deci, calculului funcțiilor grup din cerere pentru întreg
tabelul.

```

- **CUBE** grupează liniile selectate pe baza valorilor tuturor combinațiilor posibile ale expresiilor specificate și întoarce câte o linie totalizatoare pentru fiecare grup.
 - Acest operator este folosit pentru a produce mulțimi de rezultate care sunt utilizate în rapoarte. În vreme ce *ROLLUP* produce subtotalurile doar pentru o parte dintre combinațiile posibile, operatorul *CUBE* produce subtotaluri pentru toate combinațiile posibile de grupări specificate în clauza *GROUP BY*, precum și un total general.
 - Dacă există n coloane sau expresii în clauza *GROUP BY*, vor exista 2^n combinații posibile superagregat.

22. Afișați codurile departamentelor în care lucrează cel puțin un angajat, iar pentru fiecare dintre acestea și pentru fiecare manager care lucrează în departamentul respectiv afișați numărul de salariați. De asemenea, afișați numărul de salariați pentru fiecare departament indiferent de manager și numărul total de angajați din companie.

```

SELECT department_id, manager_id, COUNT(employee_id)
FROM   employees
WHERE  manager_id IS NOT NULL AND department_id IS NOT NULL
GROUP BY ROLLUP (department_id, manager_id);

```

department_id	manager_id	COUNT(employee_id)

10	7782	1
10	7839	1
10		2

20	7566	2
20	7788	1
20	7839	1
20	7902	1
20		5

30	7698	5
30	7839	1
30		6

În rezultatul prezentat anterior se pot distinge 3 tipuri de linii.

- a. Prima linie reprezintă numărul de angajați care sunt subordonați managerului 7782 și lucrează în departamentul 10. În mod similar se interpretează liniile 2, 4, 5, 6, 7, 9 și 10 din rezultat.
 - b. Linia 3, 8 și 11 din rezultat conțin numărul salariaților care lucrează în departamentele 10, 20, respectiv 30 și sunt subordonați unui manager. Aceste linii se disting prin faptul că valoarea coloanei mgr este *null*.
 - c. Ultima linie conține numărul total al salariaților care lucrează în departamentele 10, 20 și 30 și sunt subordonați unui manager. Valoarea afișată este obținută prin însumarea valorilor de pe liniile 3, 8 și 11.
- 23.** Afișați codurile departamentelor în care lucrează cel puțin un angajat, iar pentru fiecare dintre acestea și pentru fiecare manager care lucrează în departamentul respectiv afișați numărul de salariați. De asemenea, afișați numărul de salariați pentru fiecare departament indiferent de manager, numărul de angajați subordonați unui manager indiferent de departament și numărul total de angajați din companie.

```
SELECT department_id, manager_id, COUNT(employee_id)
FROM employees
WHERE manager_id IS NOT NULL AND department_id IS NOT NULL
GROUP BY CUBE (department_id, manager_id);
```

department_id	manager_id	COUNT(employee_id)

10	7782	1
10	7839	1
10		2

20	7566	2
20	7788	1
20	7839	1
20	7902	1
20		5

30	7698	5
30	7839	1
30		6

	7566	2
	7698	5
	7782	1
	7788	1
	7839	3
	7902	1

Indicație: În plus față de rezultatul corespunzător operației *ROLLUP*, operatorul *CUBE* a produs linii care reprezintă numărul de angajați subordonați fiecărui manager. Aceste linii se disting prin faptul că valoarea coloanei *department_id* este *null*.

24. Pentru fiecare departament, job, respectiv an al angajării afișați numărul de salariați. De asemenea, afișați numărul de angajați:

- pentru fiecare departament și job, indiferent de anul angajării;
- pentru fiecare departament, indiferent de job și de anul angajării;
- la nivel de companie.

25. Afișați suma alocată pentru plata salariilor pe joburi, în cadrul departamentului. De asemenea, afișați valoarea totală necesară pentru plata salariilor la nivel de departament, valoarea totală necesară pentru plata salariilor la nivel de job, indiferent de departament și valoarea totală necesară pentru plata salariilor la nivel de companie.

26. Afișați numele departamentelor, titlurile job-urilor și valoarea medie a salariilor, pentru:

- fiecare departament și, în cadrul său pentru fiecare job;
- fiecare departament (indiferent de job);
- întreg tabelul.

De asemenea, afișați și o coloană care indică intervenția coloanelor *department_name* și *job_title* în obținerea rezultatului.

Observație: Funcția *GROUPING*(expresie) întoarce:

- valoarea 0, dacă expresia a fost utilizată pentru calculul valorii agregat
- valoarea 1, dacă expresia nu a fost utilizată.

27. Afișați numele departamentelor, numele job-urilor și valoarea medie a salariilor, pentru:

- fiecare departament și, în cadrul său pentru fiecare job;
- fiecare departament (indiferent de job);
- fiecare job(indiferent de departament)
- întreg tabelul.

Cum intervin coloanele în obținerea rezultatului?

Afișați cuvântul *Dept*, dacă departamentul a intervenit în agregare, respectiv cuvântul *Job*, dacă job-ul a intervenit în agregare.

Indicație: `DECODE (GROUPING (department_name) , 0, 'Dept')`

28. Se dă următoarea cerere:

```
SELECT department_id, manager_id, COUNT(employee_id)
FROM employees
WHERE manager_id IS NOT NULL AND department_id IS NOT NULL
GROUP BY CUBE (department_id, manager_id);
```

- a. Eliminați clauza *WHERE* din această cerere. Analizați rezultatul obținut.
- b. Modificați cererea obținută astfel încât să se poată identifica dacă o valoare *null* din rezultat este stocată pe una dintre coloanele *manager_id* sau *department_id* sau este produsă de operatorul CUBE.

29. Afișați numele departamentelor, numele job-urilor, codurile managerilor, maximul și suma salariilor pentru:

- fiecare departament și, în cadrul său, fiecare job;
- fiecare job și, în cadrul său, pentru fiecare manager;
- întreg tabelul.

Observație: Clauza GROUP BY GROUPING SETS permite obținerea numai a anumitor grupări superagregat.

```
GROUP BY GROUPING SETS ((expr_11, expr_12, ..., expr_1n),  
                          (expr_21, expr_22, ...expr_2m), ...,  
                          ( ) )
```