

## **Funcții grup și clauzele GROUP BY, HAVING.**

### **I. [Funcții grup și clauza GROUP BY]**

- Clauza *GROUP BY* este utilizată pentru a diviza liniile unui tabel în grupuri. Pentru a returna informația corespunzătoare fiecărui astfel de grup, pot fi utilizate funcțiile agregat. Ele pot apărea în clauzele:
  - *SELECT*
  - *ORDER BY*
  - *HAVING*.

Server-ul *Oracle* aplică aceste funcții fiecărui grup de linii și returnează un singur rezultat pentru fiecare mulțime.

- Dintre funcțiile grup definite în sistemul *Oracle*, se pot enumera: *AVG*, *SUM*, *MAX*, *MIN*, *COUNT*, *STDDEV*, *VARIANCE* etc. Tipurile de date ale argumentelor funcțiilor grup pot fi *CHAR*, *VARCHAR2*, *NUMBER* sau *DATE*.
  - Funcțiile *AVG*, *SUM*, *STDDEV* și *VARIANCE* operează numai asupra valorilor numerice.
  - Funcțiile *MAX* și *MIN* pot opera asupra valorilor numerice, caracter sau dată calendaristică.
- Absența clauzei *GROUP BY* conduce la aplicarea funcției grup pe mulțimea tuturor liniilor tabelului.
- Toate funcțiile grup, cu excepția lui *COUNT(\*)*, ignoră valorile *null*. *COUNT(expresie)* returnează numărul de linii pentru care expresia dată nu are valoarea *null*. Funcția *COUNT* returnează 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.
- Expresiile din clauza *SELECT* a unei cereri care conține opțiunea *GROUP BY* trebuie să reprezinte o proprietate unică de grup, adică fie un atribut de grupare, fie o funcție de agregare aplicată tuplurilor unui grup, fie o expresie formată pe baza primelor două. Toate expresiile din clauza *SELECT*, cu excepția funcțiilor de agregare, se trec în clauza *GROUP BY* (unde pot apărea cel mult 255 expresii).

### **II. [Clauza HAVING]**

Opțiunea *HAVING* permite restricționarea grupurilor de linii returnate, la cele care îndeplinesc o anumită condiție.

Dacă această clauză este folosită în absența unei clauze *GROUP BY*, aceasta presupune că gruparea se aplică întregului tabel, deci este returnată o singură linie, care este reținută în rezultat doar dacă este îndeplinită condiția din clauza *HAVING*.

### **III. [Exerciții – funcții grup și clauzele GROUP BY, HAVING]**

1. a) Funcțiile grup includ valorile *NULL* în calcule?  
b) Care este deosebirea dintre clauzele *WHERE* și *HAVING*?
2. Să se afișeze cel mai mare salariu, cel mai mic salariu, suma și media salariilor tuturor angajaților. Etichetați coloanele Maxim, Minim, Suma, respectiv Media. Sa se rotunjeasca rezultatele.

```
SELECT MAX(salary) Maxim, _____, _____, _____
FROM employees;
```

3. Să se modifice cererea anterioara pentru a se afișa minimul, maximul, suma și media salariilor pentru fiecare job.

```
SELECT job_id, MAX(salary) Maxim, _____, _____, _____
FROM employees
GROUP BY job_id;
```

4. Să se afișeze numărul de angajați pentru fiecare job.

```
SELECT job_id, COUNT(*)
FROM employees
GROUP BY _____;
```

5. Să se determine numărul de angajați care sunt șefi. Etichetați coloana "Nr. manageri".

```
SELECT COUNT(DISTINCT manager_id) "Nr. Manageri"
FROM employees;
```

? De ce am folosit cuvântul cheie DISTINCT? Ce am fi obținut dacă îl omiteam?

6. Să se afișeze diferența dintre cel mai mare și cel mai mic salariu. Etichetați coloana "Diferența".

7. Scrieți o cerere pentru a se afișa numele departamentului, locația, numărul de angajați și salariul mediu pentru angajații din acel departament. Coloanele vor fi etichetate corespunzător.

```
SELECT d.department_name "Departament", l.city "Locatie",
COUNT(*) "Nr angajati", AVG(salary) "Salariu mediu"
FROM locations l, departments d, employees e
WHERE l.location_id = d.location_id
AND d.department_id = e.department_id
GROUP BY department_name, l.city;
```

**!!!Obs:** În clauza GROUP BY se trec obligatoriu toate coloanele prezente în clauza SELECT, care nu sunt argument al funcțiilor grup (a se vedea ultima observație de la punctul I).

8. Să se afișeze codul și numele angajaților care câștiga mai mult decât salariul mediu din firmă. Se va sorta rezultatul în ordine descrescătoare a salariilor.

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE salary > (SELECT AVG(salary)
FROM employees)
ORDER BY salary DESC;
```

9. Pentru fiecare șef, să se afișeze codul său și salariul celui mai prost platit subordonat. Se vor exclude cei pentru care codul managerului nu este cunoscut. De asemenea, se vor exclude grupurile în care salariul minim este mai mic de 7000\$. Sortați rezultatul în ordine descrescătoare a salariilor.

```
SELECT manager_id, MIN(salary)
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
HAVING MIN(salary) > 7000
ORDER BY 2 DESC;
```

10. Pentru departamentele in care salariul maxim depășește 7000\$, să se obțină codul, numele acestor departamente și salariul maxim pe departament.

11. Care este salariul mediu minim al job-urilor existente? Salariul mediu al unui job va fi considerat drept media aritmetică a salariilor celor care îl practică.

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

12. Să se afișeze codul, numele departamentului și media salariilor pe departamente.

13. Să se afișeze maximul salariilor medii pe departamente.

14. Sa se obtina codul, titlul și salariul mediu al job-ului pentru care salariul mediu este minim.

15. Să se afișeze salariul mediu din firmă doar dacă acesta este mai mare decât 2500. (clauza *HAVING* fără *GROUP BY*)

16. Să se afișeze suma salariilor pe departamente și, în cadrul acestora, pe job-uri.

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

! Modificați această cerere astfel încât să se afișeze numele departamentelor și titlul job-urilor.

17. Să se afișeze numele departamentului si cel mai mic salariu din departamentul avand cel mai mare salariu mediu.

18. Sa se afiseze codul, numele departamentului si numarul de angajati care lucreaza in acel departament pentru:

a) departamentele in care lucreaza mai putin de 4 angajati;

b) departamentul care are numarul maxim de angajati.

```
a) SELECT e.department_id, d.department_name, COUNT(*)
FROM departments d JOIN employees e
ON (d.department_id = e.department_id )
WHERE e.department_id IN (SELECT department_id
FROM employees
GROUP BY department_id
HAVING COUNT(*) < 4)
GROUP BY e.department_id, d.department_name;
```

Sau

```
SELECT e.department_id, d.department_name, COUNT(*)
FROM employees e JOIN departments d
ON (d.department_id = e.department_id )
GROUP BY e.department_id, d.department_name
HAVING COUNT(*) < 4;
```

19. Sa se afiseze salariatii care au fost angajati în aceeași zi a lunii în care cei mai multi dintre salariați au fost angajati.

```
SELECT employee_id, last_name, TO_CHAR(hire_date, 'dd')
FROM employees
WHERE TO_CHAR(hire_date, 'dd') IN
(SELECT TO_CHAR(hire_date, 'dd')
FROM employees
```

```

GROUP BY TO_CHAR(hire_date,'dd')
HAVING COUNT(*)=(SELECT MAX(COUNT(*))
FROM employees
GROUP BY TO_CHAR(hire_date,'dd')));

```

20. Să se obțină numărul departamentelor care au cel puțin 15 angajați.

```

SELECT COUNT(COUNT(department_id))
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 15;

```

Ce reprezintă rezultatul returnat de cererea:

```

SELECT COUNT(department_id)
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 15;

```

21. Să se obțină codul departamentelor și suma salariilor angajaților care lucrează în acestea, în ordine crescătoare. Se consideră departamentele care au mai mult de 10 angajați și al căror cod este diferit de 30.

22. Sa se afiseze codul, numele departamentului, numarul de angajati si salariul mediu din departamentul respectiv. Se vor afișa și departamentele fără angajați (outer join).

23. Scrieti o cerere pentru a afisa, pentru departamentele avand codul > 80, salariul total pentru fiecare job din cadrul departamentului. Se vor afisa orasul, numele departamentului, jobul si suma salariilor. Se vor eticheta coloanele corespunzator.

24. Care sunt angajatii care au mai avut cel putin doua joburi?

25. Să se calculeze comisionul mediu din firmă, luând în considerare toate liniile din tabel.

**Obs:** Funcțiile grup ignoră valorile *null*. Prin urmare, instrucțiunea

```

SELECT AVG(commission_pct)
FROM employees;

```

va returna media valorilor pe baza liniilor din tabel pentru care există o valoare diferită de *null*. Astfel, reiese că suma valorilor se împarte la numărul de valori diferite de *null*. Calculul mediei pe baza tuturor liniilor din tabel se poate realiza utilizând funcțiile *NVL*, *NVL2* sau *COALESCE*:

```

SELECT AVG(NVL(commission_pct, 0))
FROM employees;

```

O altă variantă este dată de o cerere de forma:

```

SELECT SUM(commission_pct)/COUNT(*)
FROM employees;

```