

LABORATOR 6: GRUPAREA DATELOR

CLAUZELE GROUP BY HAVING

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. Dacă o cerere conține clauza *GROUP BY* lista select poate conține doar funcții agregat și expresiile după care s-a realizat gruparea.

Exemplul 1: Să se obțină media salariilor pentru fiecare departament.

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

NUME	DEP_ID	SALARY	DEP_ID	AVG(salary)
Raphaely	30	11000	30	4150
Khoo	30	3100		
Baida	30	2900		
Tobias	30	2800		
Himuro	30	2600		
Colmenares	30	2500		
Hartstein	20	13000	20	9500
Fay	20	6000		

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.

Clauza *HAVING* nu are sens fără existența clauzei *GROUP BY*. Această opțiune permite restricționarea grupurilor de linii returnate, la cele care îndeplinesc o anumită condiție.

Exemplul 2: Să se obțină media salariilor pentru angajații departamentului 10

```
SELECT department_id, AVG(salary) medie_sal
FROM employees
GROUP BY department_id
HAVING department_id=10;
```

DEPARTMENT_ID	MEDIE_SAL
10	4400

Exemplul 3: Să se afișeze cel mai mare salariu, cel mai mic salariu, suma și media salariilor tuturor angajaților. Etichetați coloanele Maxim, Minim, Suma, respectiv Media. Să se rotunjeasca rezultatele.

```
SELECT MAX(salary) "Maxim", MIN(salary) "Minim", SUM(salary) "Suma",
AVG(salary) "Media"
FROM employees;
```

Exercițiul 4: Să se modifice cererea anterioară pentru a se afișa minimul, maximul, suma și media salariilor pentru fiecare job.

Exercițiul 5: 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.

Exercițiul 6: 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 1000\$. Sortați rezultatul în ordine descrescătoare a salariilor.

Exercițiul 7: 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.

Exercițiul 8: Managerul companiei are nevoie de următoarele informații: numele, prenumele, numărul de telefon și emailul celor care lucrează în același oraș cu el. Pentru fiecare angajat din lista dorește să știe și numărul de angajați din subordine. Nu este interesat să afle propriul email și nu este interesat de cei care conduc direct mai puțin de 5 salariați.

Exercițiul 9: Să se obțină codul, titlul și salariul mediu al job-ului pentru care salariul mediu este minim.

Exercițiul 10: Să se obțină numele angajaților care lucrează în departamentul cu cea mai mare medie a salariilor.

Exercițiul 11: Să se afișeze codul și numele angajatului, numele departamentului și suma salariilor pe departamente.

Exercițiul 12: Să se afișeze pentru fiecare angajat numele, salariul și salariul mediu al celor care au același șef direct cu el.

Exercițiul 13: Afișați pentru fiecare salariat următoarele informații:

- durata primului job (daca nu a avut anterior alt job se va considera jobul actual) exprimată în numărul de luni lucrate;
- numărul joburilor avute anterior.

Exercițiul 14: Să se determine numărul angajaților care nu câștigă comision.

Exercițiul 15: Să se determine numărul de manageri din fiecare departament. Etichetați coloana "Nr. Manageri".

Exercițiul 16: 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 departamentele. Se vor eticheta coloanele corespunzător.

Job Dep10 Dep20 Dep30 Total

Exercițiul 17: 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. Denumiți capetele de tabel în mod corespunzător.

Exercițiul 18: Obțineți pentru fiecare angajat următoarele informații:

- numele sau;
- numărul de departamente pe care le conduce și în care lucrează mai puțin de 10 angajați
- numărul de departamente pe care le conduce și în care lucrează mai mult de 10 angajați.

Exercițiul 19: Afișați un tabel cu numele angajaților stabilindu-se pentru fiecare angajat dacă a fost promovat/penalizat/a pastrat nivelul.

Se considera ca un angajat a fost promovat dacă media dintre min_salary și max_salary pentru ultimul job detinut anterior este mai mică decât media dintre min_salary și max_salary pentru jobul detinut în prezent. Se vor afișa și angajații care nu au schimbat jobul.

OPERATORUL EXISTS

În instrucțiunile *SELECT* imbricate, este permisă utilizarea oricărui operator logic. Pentru a se testa dacă valoarea recuperată de cererea externă există în mulțimea valorilor regăsite de cererea internă corelată, se va putea utiliza operatorul *EXISTS*. Dacă subcererea returnează cel puțin o linie, operatorul returnează valoarea *TRUE*. În caz contrar, va fi returnată valoarea *FALSE*.

Operatorul *EXISTS* asigură că nu mai este continuată căutarea în cererea internă după ce aceasta regăsește o linie.

Exemplul 20: Să se determine locațiile în care se află cel puțin un departament.

```
SELECT location_id, city
FROM locations l
WHERE EXISTS (SELECT 'x'
FROM departments
WHERE location_id = l.location_id);
```

Exercițiul 21: Să se determine departamentele în care nu există nici un angajat. Utilizați NOT EXISTS. Propune-ți alte variante de rezolvare.

Exercițiul 22: Să se afișeze numele departamentelor în care lucrează cel puțin un angajat care are salariul > 5000.

Operații cu mulțimi

Exercițiul 23: Să se afișeze numele angajaților care au lucrat cel puțin în toate departamentele în care a lucrat angajatul cu codul 206.

Division

Exercițiul 24: a. Să se afișeze numele angajaților care au lucrat cel puțin la aceleași proiecte la care a lucrat angajatul cu codul 202.

b. Să se afișeze numele angajaților care au lucrat cel mult la aceleași proiecte la care a lucrat angajatul cu codul 202.

c. Să se afișeze numele angajaților care au lucrat cel exact la aceleași proiecte la care a lucrat angajatul cu codul 202.