

Baze de Date

Solutii – Laborator 7 – Saptamana 10

LABORATOR 7 - SAPTAMANA 10

GRUPARI

--exemplu grupare

```
create table grupare (id number(5) primary key,  
                    nume varchar2(10) not null,  
                    salariu number(10) not null,  
                    manager_id number(5) not null  
                    );
```

```
select * from grupare;
```

```
insert into grupare values (1, 'user1', 1000, 1);
```

```
insert into grupare values (2, 'user2', 1400, 1);
```

```
insert into grupare values (3, 'user3', 700, 2);
```

```
insert into grupare values (4, 'user4', 300, 2);
```

```
insert into grupare values (5, 'user5', 1600, 2);
```

```
insert into grupare values (6, 'user6', 1200, 2);
```

```
commit;
```

```
--exemplu folosind clauza where
```

```
select *
```

```
from grupare
```

```
where salariu < 1100;
```

```
--exemplu folosind where si grupare
```

```
select manager_id, salariu
```

```
from grupare
```

```
where salariu < 1100
```

```
group by manager_id, salariu;
```

--exemplu folosind where, iar gruparea realizata doar dupa coloana manager_id

```
select manager_id
```

```
from grupare
```

```
where salariu < 1100
```

```
group by manager_id
```

```
having count(*) < 2;
```

--exemplu folosind having

```
select max(salariu)
```

```
from grupare
```

```
having max(salariu) < 2000;
```

--group by si having

```
select * from grupare;
```

```
select manager_id, min(salariu)
```

```
from grupare
```

```
group by manager_id
```

```
having min(salariu) < 1000;
```

1.

a) Funcțiile grup includ valorile NULL în calcule?

--funcțiile grup (sum, min, max, avg) ignoră valorile null

count(expresie) - ignoră și el valorile null

count(*) - ia în considerare și valorile null

create table test (salary number(10));

select * from test;

insert into test values (10);

insert into test values (20);

insert into test values (30);

insert into test values (null);

insert into test values (null);

select avg(salary)

from test;

select sum(salary) / count(salary)

from test;

```
select sum(salary) / count(*)
```

```
from test;
```

count(*) numara si valorile null

count(expresie) nu numara valorile null

b) Care este deosebirea dintre clauzele WHERE și HAVING?

where salary >

group by

having max()

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 media salariilor.

```
SELECT MAX(salary) Maxim, MIN(salary), SUM(salary),  
round(AVG(salary))
```

```
FROM employees;
```

3. Să se modifice problema 2 pentru a se afișa minimul, maximul, suma și media salariilor
pentru FIECARE job.

IT_PROG MAX, MIN, SUM, AVG

SA_REP MAX, MIN, SUM, AVG

... pt toate joburile ...

select * from employees;

SELECT job_id, job_title, MAX(salary) Maxim, MIN(salary), SUM(salary),
AVG(salary)

FROM employees join jobs using (job_id)

GROUP BY job_id, job_title;

4. Să se afișeze numărul de angajați pentru FIECARE departament.

SELECT nvl(to_char(department_id), 'Angajati fara departament'),
COUNT(employee_id)

FROM employees

--WHERE department_id is not null

GROUP BY department_id;

5. Să se determine numărul de angajați care sunt șefi.

Etichetati coloana "Nr. manageri".

```
select count(distinct manager_id) "Nr. manageri"
from employees;
```

6. Să se afișeze diferența dintre cel mai mare si cel mai mic salariu.

Etichetati coloana "Diferenta".

```
SELECT max(salary)- min(salary) Diferenta
FROM employees;
```

-- pentru fiecare departament in parte

```
SELECT department_id, max(salary)- min(salary) Diferenta
FROM employees
WHERE department_id is not null
GROUP BY department_id;
```

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.

Se vor afisa si angajatii care nu au departament

!!!Obs: În clauza GROUP BY se trec obligatoriu toate coloanele prezente în clauza SELECT,

care nu sunt argument al funcțiilor grup.

```
SELECT department_name, location_id, count(employee_id),
       round(avg(salary))
```

```
FROM employees e left join departments d on (e.department_id =
d.department_id)
```

```
GROUP BY department_name, location_id;
```

```
select * from departments;
```

8. Să se afișeze codul și numele angajaților care au salariul mai mare 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 round(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 1000\$.

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) > 5800
order by 2 desc;
```

10. Pentru departamentele în care salariul maxim depășește 3000\$, să se obțină codul, numele acestor departamente și salariul maxim pe departament.

```
SELECT department_id, department_name, MAX(salary)
FROM departments JOIN employees USING(department_id)
GROUP BY department_id, department_name
HAVING MAX(salary) >= 3000;
```

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(round(avg(salary)))
FROM employees
GROUP BY job_id;
```

12. Să se afișeze maximul salariilor medii pe departamente.

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

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

este minim.

-- Rezolvati

```
select e.job_id, j.job_title, min(sal)
from employees e join jobs j on(e.job_id = j.job_id)
    join (select job_id, avg(salary) sal
          from employees
          group by job_id) a on (a.job_id = e.job_id)
group by e.job_id, job_title;
```

```

select e.last_name, e.employee_id, mm
from employees e join (select department_id, min(salary) mm
                        from employees
                        group by department_id
) d on (e.department_id = d.department_id);

```

14. Să se afișeze salariul mediu din firmă doar dacă acesta este mai mare decât 2500.

```

SELECT round(AVG(salary))
FROM employees
HAVING AVG(salary) > 2500;

```

15. 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;

```

16. 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;

-- vrem sa afisam toate depart -> atat pe cele care au ang cat si pe cele care nu au

-- departamentele care nu au angajati au zero angajati

-- $0 < 4$ -> deci vom afisa si departamentele care au zero angajati

```
SELECT d.department_id, d.department_name, COUNT(*)
```

```
FROM employees e right JOIN departments d ON (d.department_id =  
e.department_id )
```

```
GROUP BY d.department_id, d.department_name
```

```
HAVING COUNT(*) < 4;
```

```
SELECT d.department_id, d.department_name, COUNT(employee_id)
```

```
FROM employees e right JOIN departments d ON (d.department_id =  
e.department_id )
```

```
GROUP BY d.department_id, d.department_name
```

```
HAVING COUNT(employee_id) < 4;
```