LABORATOR 5

-- LABORATOR 3 + 4

10. Folosind subcereri, să se afişeze numele şi data angajării pentru salariaţii care au

fost angajaţi după Gates.

SELECT last\_name, hire\_date

FROM employees

WHERE hire\_date > ( SELECT hire\_date

FROM employees

WHERE INITCAP(last\_name)= 'Gates'

-- subcerere care returneaza data angajarii lui Gates

);

11. Folosind subcereri, scrieţi o cerere pentru a afişa numele şi salariul pentru toţi

colegii (din acelaşi departament) lui Gates. Se va exclude Gates.

Se poate înlocui operatorul IN cu = ???

-- subcerere care selecteaza departamentul lui Gates

select department\_id

from employees

where initcap(last\_name) = 'Gates';

-- cererea principala

select last\_name, salary

from employees

where department\_id = ( select department\_id

from employees

where initcap(last\_name) = 'Gates'

)

and initcap(last\_name) != 'Gates';

--Se va inlocui Gates cu King

select last\_name, salary, department\_id

from employees

where department\_id IN ( select department\_id

from employees

where initcap(last\_name) = 'King'

)

and initcap(last\_name) != 'King';

Obs: operator poate fi:

• single-row operator (>, =, >=, <, <>, <=), care poate fi utilizat dacă

subcererea returnează o singură linie;

• multiple-row operator (IN, ANY, ALL), care poate fi folosit dacă subcererea

returnează mai mult de o linie.

12. Folosind subcereri, să se afişeze numele şi salariul angajaţilor conduşi direct de

preşedintele companiei (acesta este considerat angajatul care nu are manager).

--subcerere nesincronizata care determina managerul companiei

select employee\_id

from employees

where manager\_id is null;

--cererea principala

select last\_name, salary

from employees

where manager\_id = (select employee\_id

from employees

where manager\_id is null

);

13. Scrieți o cerere pentru a afişa numele, codul departamentului și salariul

angajaților al căror cod de departament și salariu coincid cu codul departamentului și

salariul unui angajat care câștigă comision.

SELECT last\_name, department\_id, salary

FROM employees

WHERE (department\_id, salary) IN (SELECT department\_id, salary

FROM employees

WHERE commission\_pct is not null

--codul departamentului și

--salariul unui angajat care câștigă comision

);

14. Să se afișeze codul, numele și salariul tuturor angajaților al căror salariu este mai

mare decât salariul mediu;

SELECT employee\_id, last\_name, salary

FROM employees

WHERE salary > (SELECT AVG(salary)

FROM employees);

15. Scrieti o cerere pentru a afișa angajații care câștigă (salariul plus comision) mai mult

decât oricare funcționar (job-ul conţine şirul “CLERK”). Sortați rezultatele dupa salariu,

în ordine descrescătoare.

-- subcerere pentru a selecta venitul pe care il au functionarii

select salary + salary \* nvl(commission\_pct, 0)

from employees

where upper(job\_id) like '%CLERK%';

--cererea principala

select \*

from employees

where salary + salary \* nvl(commission\_pct, 0) > ALL (select salary + salary \* nvl(commission\_pct, 0)

from employees

where upper(job\_id) like '%CLERK%'

);

> ANY -> inseamna ca este suficient sa fie mai mare decat salariul unui singur functionar (mai mare ca minimul)

> ALL -> inseamna ca trebuie sa fie mai mare decat salariile tuturor functionarilor (mai mare ca maximul)

--varianta 2

select \*

from employees

where salary + salary \* nvl(commission\_pct, 0) > (select max(salary + salary \* nvl(commission\_pct, 0))

from employees

where upper(job\_id) like '%CLERK%'

)

order by salary desc;

16. Scrieţi o cerere pentru a afişa numele angajatilor, numele departamentului şi

salariul angajaţilor care nu câştigă comision, dar al căror şef direct câştigă comision.

SELECT last\_name, department\_name, salary

FROM employees e JOIN departments d USING (department\_id)

WHERE commission\_pct is null and

e.manager\_id IN (SELECT employee\_id

FROM employees

WHERE commission\_pct is not null);

select \* from employees;

17. Să se afișeze numele angajatilor, departamentul, salariul şi job-ul tuturor

angajaților al căror salariu și comision coincid cu salariul și comisionul unui angajat din

Oxford.

SELECT last\_name, department\_id, salary, job\_id, employee\_id

FROM employees

WHERE (nvl(commission\_pct, -1), salary) IN

(SELECT nvl(commission\_pct, -1), salary

FROM employees e JOIN departments d ON (e.department\_id = d.department\_id)

JOIN locations l ON (l.location\_id = d.location\_id)

WHERE initcap(l.city)='Oxford'

);

(-1, 24000) IN ...

-- LABORATOR 4

create table grupare (id number(5) not null,

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;

--exemplu folosind having

select max(salariu)

from grupare

having max(salariu) < 10000;

--group by si having

select manager\_id, min(salariu)

from grupare

group by manager\_id

having min(salariu) <= 1000;

select \* from grupare;

1. where

2. group by

3. having -- ordinea de executie

drop table grupare;

1. a) Functiile grup includ valorile NULL in calcule?

select count(department\_id)

from employees ;--105

select count(department\_id)

from departments;--pk--27

R: da

2. Să se afişeze cel mai mare salariu, cel mai mic salariu, suma şi media salariilor

TUTUROR angajaţilor. Etichetaţi coloanele Maxim, Minim, Suma, respectiv Media. Sa se

rotunjeasca media salariilor.

SELECT MAX(salary) Maxim, MIN(salary) Minim, SUM(salary) Suma, round(AVG(salary)) Medie

FROM employees;

3. Să se modifice problema 2 pentru a se afişa minimul, maximul, suma şi media

salariilor pentru FIECARE job.

SELECT job\_id, MAX(salary) Maxim, MIN(salary) Minim, SUM(salary) Suma, round(AVG(salary)) Medie

FROM employees

GROUP BY job\_id;

4. Să se afişeze numărul de angajaţi pentru FIECARE departament.

SELECT COUNT(employee\_id), department\_id

FROM employees

GROUP BY department\_id;

--daca exista departamente fara angajati dorim sa afisam id-ul depart si zero angajati

--afisam toate departamentele chiar daca au sau nu angajati

SELECT COUNT(employee\_id), d.department\_id

FROM employees e right join departments d on (e.department\_id = d.department\_id)

GROUP BY d.department\_id; --(1)

-- count(\*) nu ignora valorile null

SELECT COUNT(\*), d.department\_id

FROM employees e right join departments d on (e.department\_id = d.department\_id)

GROUP BY d.department\_id; --(2)

count(expresie) varianta (1) -> returneaza numarul de linii pentru care expresia data NU ARE valoarea null

count(\*) varianta (2) -> NU IGNORA valorile null (adica ia in considerare toate liniile)

5. Să se determine numărul de angajaţi care sunt şefi. Etichetati coloana “Nr.

manageri”.

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

select count(distinct manager\_id) "Nr manageri"

from employees;

6. Să se afişeze diferenţa dintre cel mai mare si cel mai mic salariu. Etichetati

coloana “Diferenta”.

select max(salary) - min(salary) Diferenta

from employees;

7. Scrieţi o cerere pentru a se afişa numele departamentului, locaţia, numărul de

angajaţi şi salariul mediu pentru angajaţii din acel departament. Coloanele vor fi

etichetate corespunzător.

select department\_name, location\_id, count(employee\_id), round(avg(salary))

from employees e join departments d on (e.department\_id = d.department\_id)

group by department\_name, location\_id;

TEMA: Laborator 3 - ex: 18 + Laborator 4 - ex: 9 (0.5pct) - primele 3 surse corecte

Laborator 4 - ex: 8, 10, 12, 13, 14 - (bonus la final)

Deadline -> 28.03 (Sambata - inclusiv)

18. Să se afişeze numele angajaţilor, codul departamentului şi codul job-ului salariaţilor al căror departament se află în Toronto.

select last\_name, department\_id, job\_id

from employees

where department\_id in ( select department\_id

from departments join locations using (location\_id)

where initcap(city)='Toronto');

9. Pentru fiecare şef, să se afişeze codul său şi 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$.

Sortaţi rezultatul în ordine descrescătoare a salariilor.

select e.manager\_id , min(ee.salary)

from employees e join employees ee on (e.employee\_id=ee.employee\_id )

where e.manager\_id is not null and ee.employee\_id!=e.manager\_id and ee.salary>1000

group by e.manager\_id

ORDER BY min(ee.salary) DESC;

10. Pentru departamentele in care salariul maxim depăşeşte 3000$,

să se obţină codul, numele acestor departamente şi salariul maxim pe departament.

select d.department\_id, d.department\_name, max(salary)

from employees e join departments d on (e.department\_id=d.department\_id)

group by d.department\_id, d.department\_name, salary

having max(salary)>3000;

12.Să se afişeze maximul salariilor medii pe departament

select max(avg(salary))

from employees

group by department\_id;

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

select avg(salary)

from employees

having avg(salary)>2500;

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

select j.job\_id, j.job\_title, avg(e.salary)

from employees e join jobs j on (e.job\_id=j.job\_id)

group by j.job\_id, j.job\_title

having avg(salary) =(select min(avg(salary))

from employees

group by job\_id);