**PRACTIC:**

**1 . Actualizati salariul angajatului cu identificatorul 200 si egalati-l cu cel al unuia dintre ceilalti salariati, al carui identificator il furnizati dinamic de la tastatura.**

UPDATE employees

SET salary = (SELECT salary

FROM employees

WHERE employee\_id=&a)

WHERE employee\_id =200;

**2. Transferati angajatul cu identificatorul 108 intr-un departamnet al carui identificator il introduceti de la tastatura.**

UPDATE employees

SET department\_id= &id\_dep\_nou

WHERE employee\_id=108;

**3. Utilizand expresiile conditionale scrieti o interigare care sa afiseze numele departamentului in care lucreaza angajatii pe baza valorilor din coloana department-id astfel:**

Graphical user interface, application, table

Description automatically generated

SELECT d.department\_name ,e.department\_id ,

CASE e.department\_id

WHEN 10 THEN 'Administrativ'

WHEN 40 THEN 'Resurse umane'

WHEN 60 THEN 'Informatica'

ELSE 'Alte departamente'

END

FROM Departments d, Employees e

WHERE e.department\_id=d.department\_id;

SAU

SELECT d.department\_name,e.department\_id ,

DECODE ( e.department\_id ,

10 , 'Administrativ',

40, 'Resurse umane',

60 , 'Informatica',

'Alte departamente')

FROM Departments d, Employees e

WHERE e.department\_id=d.department\_id;

**4. Scrieti o interogare care sa intoarca:**

**- media salariilor din companie;**

**- media salariior pentru fiecare tip de job;**

**- in cadrul fiecarui tip de job, media salariilor pentru fiecare department.**

SELECT department\_id, job\_id, AVG(salary)

FROM employees

GROUP BY ROLLUP (department\_id, job\_id);

**5. Scrieti o interogare care sa intoarca istoricul angajatilor din companie : nume, prenume, numele departamentelor in care au lucrat si numele joburilor ocupate.**

SELECT h.employee\_id, h.job\_id,e.last\_name, e.first\_name, d.department\_name, j.job\_title

FROM Employees e , Departments d, Jobs j, Job\_history h

WHERE h.department\_id=d.department\_id AND h.job\_id=j.job\_id AND e.employee\_id=h.employee\_id;

**6. Scrieti o interogare care sa intoarca numele si prenumele angajatilor, numele departamentelor in care lucreaza si orasul in care se gasesc.**

SELECT e.last\_name, e.first\_name, d.department\_name, l.city

FROM Employees e , Departments d, locations l

WHERE e.department\_id=d.department\_id AND d.location\_id=l.location\_id;

**7. Afisati numele angajatului, salariul si numele managerului pentru toti angajatii care au salariul si comisionul precum ‘Kochhar’.**

SELECT e1.first\_name, e1.last\_name, e2.salary , e2.first\_name, e2.last\_name

FROM Employees e1, Employees e2

WHERE(e1.salary,NVL(e1.commission\_pct,0)) IN

(SELECT salary ,NVL(commission\_pct,0)

FROM Employees

WHERE last\_name='Kochhar')

AND e1.manager\_id = e2.employee\_id ;

**8. Utilizând expresiile conditionale scrieti o interogare care sa clasifice angajatii pe baza valorilor din coloana JOB\_ID astfel:**

|  |  |
| --- | --- |
| Job\_ID | Clasa |
| AD\_PRES | Management |
| AD\_VP | Management |
| IT\_PPROG | Executie |
| Nici unul din cele de mai sus | Nespecificat |
|  |  |
|  |  |

SELECT first\_name ,last\_name,

CASE job\_id

WHEN 'AD\_PRES' THEN 'Management'

WHEN 'AD\_VP' THEN 'Management'

WHEN 'IT\_PROG' THEN 'Executie'

ELSE 'Nespecificat'

END

FROM Employees ;

SAU

SELECT first\_name, last\_name,

DECODE ( job\_id ,

'AD\_PRES' , 'Management',

'AD\_VP' , 'Management',

'IT\_PROG' , 'Executie',

'Nespecificat')

FROM employees;

**9. Gasiti numele angajatului si numele functiei pe care o ocupa, pentru acei angajati care nu au manageri.**

SELECT last\_name||' '||first\_name,j.job\_title

FROM Employees e, Jobs j

WHERE e.manager\_id IS NULL and e.job\_id=j.job\_id;

**10. Afisati numele, prenumele si nr de telefon pentru toti angajatii care au acelasi salariu si commission ‘Tobias’.**

SELECT FIRST\_NAME,LAST\_NAME,PHONE\_NUMBER

FROM employees

WHERE (salary, NVL (commission\_pct, 0)) IN

(SELECT salary, NVL (commission\_pct, 0)

FROM employees

WHERE last\_name = 'Tobias');

**2. Actualizati jobul angajatului cu identificatorul 200 si egalati-l cu cel al unuia dintre ceilalti salariati, al carui identificator il furnizati dinamic de la tastatura.**

UPDATE employees

SET job\_id = (SELECT salary FROM employees WHERE employee\_id = &id)

WHERE employee\_id = 200;

**11. Scrieti o interogare care sa intoarca:**

**- salariul minim din companie;**

**- salariul minim pentru fiecare tip de job;**

**- in cadrul fiecarui tip de job, salariul minim pentru fiecare department.**

SELECT department\_id, job\_id, MIN (salary)

FROM employees

GROUP BY ROLLUP (department\_id, job\_id);

**12. Intocmiti o lista cu angajatii al caror nume(last\_name) incepe cu literele: K, A si M(exact in aceiasi ordine).**

SELECT first\_name, last\_name

FROM employees

WHERE last\_name LIKE 'K%'

UNION ALL

SELECT first\_name, last\_name

FROM employees

WHERE last\_name LIKE'A%'

UNION ALL

SELECT first\_name, last\_name

FROM employees

WHERE last\_name LIKE'M%';

**13. Gasiti o lista a departamentelor pentru care nu exista angajati.(lab 6-6)**

SELECT department\_id

FROM departments

MINUS

SELECT department\_id

FROM employees

WHERE department\_id IS NOT NULL;

**14. Care este identificatorul locatiilor in care avem cel putin un angajator.**

select location\_id

from Departments

minus

select employee\_id

from employees

where 1> (select count(employee\_id)

from employees);

1. **Scrieți o interogare care să întoarcă ierarhia din departamentul în care se gășește angajatul Lex De Haans. Afișați numele, salariul, identificatorul departamentului și identificatorul managerului.**

select first\_name,last\_name,salary,department\_id,manager\_id,level

from employees

start with department\_id = (select department\_id from employees

where first\_name='Lex')

connect by prior manager\_id = employee\_id;

select first\_name,last\_name,salary,department\_id,manager\_id,level

from employees

start with last\_name='De Haan'

connect by prior manager\_id = employee\_id;

1. **Care este numele departamentului în care se plătește(cumulat) cea mai mare sumă pentru salarii.**

select department\_name from (select department\_name, SUM(salary) as sumsal from employees e, departments d

Where e.department\_id = d.department\_id

group by department\_name order by sumsal desc) where rownum = 1;

select \* from

(select \*

from( select dep ,max(salary) as sal from(

select d.department\_name as dep, sum(e.salary) as salary from employees e,departments d

where e.department\_id=d.department\_id

group by d.department\_name

)

group by dep

)

where sal !=0

order by sal desc

)

where rownum<=1;

1. **Se dorește o listă cu numele și salariile angajațiilor care câștigă mai mult decât o sumă specificată de la tastatură.**

select first\_name,last\_name,salary from employees

where salary>&sal;

1. **Găsiți numărul de țări din fiecare regiune**

select count(l.country\_name),r.region\_name from locations l,regions r

where l.region\_id=r.region\_id

group by r.region\_name;

1. **Găsiți acei angajați care lucrează în departamente din locația identificatorului 1700.**

select e.first\_name, e.last\_name from employees e,departments d

where e.department\_id=d.department\_id and d.location\_id=1700;

1. **Care este deăpartamentul cu cea mai mica medie salariala.**

select department\_name from (select department\_name, AVG(salary) as sumsal from employees e, departments d

Where e.department\_id = d.department\_id

group by department\_name order by sumsal ) where rownum = 1;

with

dep\_name\_avg as

(select d.department\_name dep, avg(e.salary) salary from employees e,departments d

where e.department\_id=d.department\_id

group by d.department\_name)

select dep from dep\_name\_avg

where salary <(select min(salary) from dep\_name\_avg);

SELECT department\_name

FROM departments d, employees e

WHERE d.department\_id = e.department\_id

HAVING AVG (salary) = (

SELECT MIN (salary)

FROM employees e2

WHERE e2.department\_id = e.department\_id)

1. **Scrieti o interogare care sa realizeze organigrama firmei incepand cu managerul general**

select \*

from employees

start with manager\_id is null

connect by prior employee\_id = manager\_id

1. **Scrieti o interogare care sa intoarca:**

**- numarul angajatilor din companie;**

**- numarul angajatilor pentru fiecare tip de job;**

**- in cadrul fiecarui tip de job, numarul angajatilor pentru fiecare departament**

SELECT department\_id, job\_id, COUNT (job\_id)

FROM employees

GROUP BY ROLLUP (department\_id, job\_id);

1. **Numarul cator angajati din companie incepe cu una din litere: A, K sau P?**

select COUNT(last\_name) from employees where last\_name like 'K%'

UNION ALL select COUNT(last\_name) from employees where last\_name LIKE 'A%'

UNION ALL select COUNT(last\_name) from employees where last\_name LIKE 'M%'

1. **Care este identificatorul locatiei in care lucreaza managerul general?**

select l.location\_id , e.last\_name from employees e, departments d, locations l where e.department\_id=d.department\_id and d.location\_id=l.location\_id and e.manager\_id is null

1. **Gasiti angajatii care lucreaza in aceeasi locatie cu managerul general.**

SELECT e.first\_name FROM Employees e, Departments d WHERE

e.department\_id=d.department\_id AND d.location\_id IN(SELECT d1.location\_id FROM Departments d1, Employees e1 WHERE e1.department\_id=d1.department\_id AND e1.manager\_id IS NULL);

1. **Afisati numele, identificatorul departamentului si salariul pentru angajatii al caror identificatory de department si salariu corespund celor ale angajatilor care au commission.**

select first\_name,last\_name,department\_id,salary,commission\_pct from employees

where (department\_id,salary) in (select department\_id,salary from employees where commission\_pct is **not** null);

1. **Afisati numele, identificatorul departamentului si salariul pentru angajatii al caror identificatory de department si salariu corespund celor ale angajatilor care nu au commission.**

select first\_name,last\_name,department\_id,salary,commission\_pct from employees

where (department\_id,salary) in (select department\_id,salary from employees where commission\_pct is null);

1. **Scieti o interoogare care sa intoarca:**

* **Totalul salariilor platite pentru fiecare departament**
* **Totalul salariilor platite pentru fiecare job din cadrul fiecarui departament**
* **Totalul salariilor platite pentru subordonatii fiecarui manager**

select manager\_id, department\_id, job\_id, sum(salary) from employees

group by rollup (manager\_id, department\_id, job\_id);

1. **Gasiti angajatii care lucreaza in acelasi oras cu managerul general.**

select e.first\_name, e.last\_name from employees e, locations l, departments d

where e.department\_id=d.department\_id and d.location\_id=l.location\_id and l.city = (select l.city from employees e, locations l, departments d where e.manager\_id is null and e.department\_id=d.department\_id and d.location\_id=l.location\_id);

1. **Folosind operatorii pe multimi gasiti denumirea departamenterol in care nu exista angajati.**

select department\_id from departments

minus

select department\_id from employees;

1. **Scrieti o interogare care sa intoarca numele, prenumele si salariul acelor angajati care au acelasi manger ca si Popp.**

select last\_name,first\_name,salary from employees

where manager\_id= (select manager\_id from employees

where last\_name='Popp');

1. **Scrieti o interogare care sa intoarca numele si prenumele angajatilor, precum si numele job-ului pe care il ocupa.**

select e.last\_name,e.first\_name,j.job\_title from employees e,jobs j

where e.job\_id=j.job\_id ;

1. **Scrieti o interogare care sa intoarca numele angajatilor si departamentelor , pentru angajatii care au salariul mai mare decat 5000 si nu au puncte de comision.**

select e.last\_name,e.first\_name,d.department\_name from employees e,departments d

where d.department\_id=e.department\_id and e.salary>5000 and commission\_pct is null ;

1. **Scrieti o interogare care sa realizeze organigrama firmei incepand cu managerul general, si excluxand toti angajatii a caror nume incepe cu S.**

select last\_name,first\_name,salary,level from employees

start with manager\_id is null

connect by manager\_id= prior employee\_id and last\_name not like 'S%' ;