

**Tema 1:**

**Laborator 1, exercitiul scris cu bold de la pagina 3**

Creați o cerere prin care să se afișeze toate datele din tabelul EMPLOYEES pe o singură coloană. Separați fiecare coloană printr-o virgulă. Etichetați coloana "Informatii complete".

```
SELECT concat(first_name,' ')|| last_name || ' castiga ' ||  
          salary || ' lunar dar doreste ' || salary * 3 "Salariu Ideal"  
FROM employees;
```

**Laborator 1, ex 10**

10. Să se afișeze numele, job-ul și data la care au început lucrul salariații angajați între 20 Februarie 1987 și 1 Mai 1989. Rezultatul va fi ordonat crescător după data de început

```
SELECT last_name, job_id, hire_date  
FROM employees  
WHERE hire_date BETWEEN '20-FEB-1987' and '1-MAY-1989'  
ORDER BY hire_date;
```

**Tema 2:**

**Laborator 2, ex 23**

23. Creați o cerere prin care să se afișeze **numele angajatilor, codul job-ului, titlul job-ului, numele departamentului și salariul** angajaților. Se vor include și angajații al căror departament nu este cunoscut.

```
SELECT last_name, j.job_id, job_title, department_name, salary  
FROM employees e, departments d, jobs j  
WHERE e.department_id = d.department_id (+) and j.job_id = e.job_id;
```

**Laborator 2, ex 25**

25. Să se afișeze **numele** salariatului și **data angajării** împreună cu **numele și data angajării șefului direct** pentru salariații care au fost angajați **înaintea** șefilor lor. Se vor eticheta coloanele Angajat, Data\_ang, Manager si Data\_mgr.

```
SELECT ang.last_name Angajat, ang.hire_date "Data angajarii ang",  
       sef.last_name Manager, sef.hire_date "Data angajarii sef"  
FROM employees ang, employees sef  
WHERE ang.manager_id = sef.employee_id  
      AND ang.hire_date < sef.hire_date;
```

## Tema 3:

### Laborator 3, ex 18

18. Să se afișeze **numele angajaților**, **codul departamentului** și **codul job-ului** salariaților al căror departament se află în Toronto.

```
SELECT last_name, department_id, job_id
FROM employees
WHERE department_id = (SELECT department_id
                      FROM departments d join locations l on (d.location_id = l.location_id)
                      WHERE lower(l.city) = 'toronto');
```

### Laborator 4, ex 9

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) >= 1000
ORDER BY 2 DESC;
```

### Laborator 4, ex 10

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;
```

### Laborator 4, ex 12

12. Să se afișeze **maximul salariilor medii** pe departamente.

```
SELECT max(avg(salary))
FROM employees
GROUP BY department_id;
```

**Laborator 4, ex 13**

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

```
SELECT job_id, job_title, avg(salary)
FROM employees JOIN jobs USING(job_id)
GROUP BY job_id, job_title
HAVING avg(salary) = (SELECT min(avg(salary))
                     FROM employees
                     GROUP BY job_id);
```

**Laborator 4, ex 14**

14. Să se afișeze **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;
```