

# Tutoriat 4

## Baze de Date

Laura-Maria Tender, Ana-Maria Comorașu

30 Martie 2021

### Curs

Realizați schema relațională corespunzătoare diagramei conceptuale proiectate pentru o platformă de music streaming, având următoarea descriere:

- Utilizatorii pot asculta mai multe melodii.
- Artiștii pot lansa mai multe melodii.
- Utilizatorul poate avea un singur tip de abonament(in prezent).
- Un playlist poate fi creat de un singur utilizator si poate contine mai multe melodii.

Pentru realizarea diagramei puteți folosi [Diagrams.net](https://diagrams.net)

**UTILIZATOR** (utilizator\_id#, username, nume\_utilizator, prenume\_utilizator, data\_inregistrarii, abonament\_id)

**ABONAMENT** (abonament\_id#, tip\_abonament, pret, durata\_abonament)

**PLAYLIST** (playlist\_id#, denumire\_playlist, mood, utilizator\_id)

**ASCULTA** (utilizator\_id#, melodie\_id#, nr\_streams, ultima\_data)

**CONTINE** (melodie\_id#, playlist\_id#)

**MELODIE** (melodie\_id#, denumire\_melodie, data\_lansarii, durata\_melodie, gen\_muzical, artist\_id)

**ARTIST** (artist\_id#, nume\_scena, nume\_artist, prenume\_artist, contact, nationalitate)

### Laborator

#### Subcereri

##### Necorelate (nesincronizate)

Afișează codul, numele, prenumele si salariul angajaților care câștigă mai mult decât media.

```
1 SELECT employee_id, first_name, last_name, salary
2 FROM EMPLOYEES
3 WHERE SALARY > (SELECT AVG(SALARY)
4                 FROM EMPLOYEES);
```

##### Corelate (sincronizate)

Operatori single-row >, =, >=, <, <=, <>

Afișează codul, numele, prenumele și salariul angajaților care câștigă mai mult decât media salariilor din departamentul său.

```
1 SELECT employee_id, first_name, last_name, salary
2 FROM EMPLOYEES e1
3 WHERE SALARY > (SELECT AVG(SALARY)
4                 FROM EMPLOYEES e2
5                 WHERE e2.DEPARTMENT_ID = e1.DEPARTMENT_ID);
```

Operatori multiple-row IN, ANY și ALL.

Afișează codul, numele, prenumele angajaților și numele departamentului pentru angajații din departamentele ce sunt conduse de Russell sau Hunold.

```
1 SELECT e.employee_id, e.first_name, e.last_name, d.department_name
2 FROM EMPLOYEES e JOIN departments d USING (department_id)
3 WHERE DEPARTMENT_ID IN (SELECT D.DEPARTMENT_ID
4                          FROM DEPARTMENTS d JOIN EMPLOYEES e
5                          ON (D.MANAGER_ID = E.EMPLOYEE_ID));
```

## Comenzi LMD

### Comanda DELETE

1. Să se ștergă toate liniile din tabela employees (emp\_pnu). Aplicați comanda rollback.

```
1 DELETE FROM emp_aco;
2
3 SELECT *
4 FROM emp_aco;
5
6 ROLLBACK;
```

2. Să se ștergă toate liniile din tabela employees (emp\_pnu), unde salariul este mai mic decăt 10000.

```
1 DELETE FROM emp_aco
2 WHERE
3     salary < 10000;
```

### Comanda INSERT

1. Să se insereze din nou liniile șterse anterior.

```
1 INSERT INTO emp_aco
2     SELECT *
3     FROM employees
4     WHERE salary < 10000;
5
6 SELECT *
7 FROM emp_aco;
```

2. Astăzi în compania noastră a mai fost angajat un intern în departamentul 60, jobul *IT\_PROG*. Adăugați-l pe Bill Gates în tabela emp\_pnu. id: 207, email = prima litera din prenume + nume, telefon: 650.501.3875, salariul 4000, manager 103.

```
1 INSERT INTO emp_aco (
2     employee_id, first_name, last_name,
3     email, phone_number, hire_date,
4     job_id, salary, manager_id,
5     department_id
6 ) VALUES (
7     207, 'Bill', 'Gates',
8     'bgates', '650.501.3875', sysdate,
9     'IT_PROG', 4000, 103,
10    60
11 );
```

### Comanda UPDATE

1. Din cauza pandemiei, compania la care lucrează cei din tabela employees s-a hotărât să micșoreze salariile. Actualizați toate salariile astfel încât sa fie 75% din salariul initial.

```
1 UPDATE emp_aco
2 SET salary = 75 / 100 * salary;
```

2. Ups! A fost o greșeală la inserarea noului angajat Bill Gates, de fapt îl cheamă Bill Clinton. Vă rugăm să faceți modificările corespunzătoare.

```

1 UPDATE emp_aco
2 SET
3     last_name = 'Clinton',
4     email = 'bclinton'
5 WHERE
6     employee_id = 207;

```

## Erori LMD

### Not null

```

1 -- not null
2 INSERT INTO emp_aco (
3     employee_id, first_name, email,
4     phone_number, hire_date, job_id,
5     salary, manager_id, department_id
6 ) VALUES (
7     207, 'Bill', 'bgates',
8     '650.501.3875', sysdate, 'IT_PROG',
9     4000, 103, 60
10 );

```

### Constrângere de unicitate

```

1 INSERT INTO emp_aco(
2     employee_id, first_name, last_name,
3     email, phone_number, hire_date,
4     job_id, salary, manager_id, department_id
5 ) VALUES (
6     206, 'Bill', 'Gates',
7     'bgates', '650.501.3875', sysdate,
8     'IT_PROG', 4000, 103, 60
9 );

```

### Constrângeri de tip check

```

1 INSERT INTO employees(
2     employee_id, first_name, last_name,
3     email, phone_number, hire_date,
4     job_id, salary, manager_id, department_id
5 ) VALUES (
6     207, 'Bill', 'Gates',
7     'bgates', '650.501.3875', sysdate,
8     'IT_PROG', 4000, 103, 280
9 );
10
11 select * from departments;
12 select * from user_constraints;

```

### Incompatibilitate în privința tipurilor de date

```

1 INSERT INTO employees(
2     employee_id, first_name, last_name,
3     email, phone_number, hire_date,
4     job_id, salary, manager_id, department_id
5 ) VALUES (
6     207, 'Bill', 'Gates',
7     'bgates', '650.501.3875', 135,
8     'IT_PROG', 4000, 103, 280
9 );
10

```

## Inserarea unei valori cu o dimensiune mai mare decât a coloanei corespunzătoare

```
1  INSERT INTO employees(  
2      employee_id, first_name, last_name,  
3      email, phone_number, hire_date,  
4      job_id, salary, manager_id, department_id  
5  ) VALUES (  
6      8000000, 'Bill', 'Gates',  
7      'bgates', '650.501.3875', sysdate,  
8      'IT_PROG', 4000, 103,  
9      280  
10 );  
11
```