

# Proiect Baze de date

## An I

Stîngă Alexandru-Ionuț

Grupa 143

### Contents

I.	Descrierea modelului real, a utilității acestuia și a regulilor de funcționare .....	2
II.	Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.....	2
III.	Descrierea entităților, incluzând precizarea cheii primare. ....	3
IV.	Descrierea relațiilor, incluzând precizarea cardinalității acestora. ....	4
V.	Descrierea atributelor, incluzând tipul de date și eventualele constrângerile, valori implicate, valori posibile ale atributelor.....	5
VI.	Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.....	7
VII.	Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ. ....	8
VIII.	Enumerarea schemelor relationale corespunzătoare diagramei conceptuale proiectate la punctul 7.	8
IX.	Realizarea normalizării până la forma normală 3 (FN1-FN3).....	9
X.	Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). .....	11
XI.	Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele associative; maxim 30 de înregistrări în fiecare tabel).....	11
XII.	Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:.....	23
XIII.	Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.....	28
XIV.	Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.....	31
XV.	Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.	
	33	
XVI.	Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relationale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.....	36

## I. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare

Site-ul unui muzeu permite crearea unui cont de **Utilizator**.

Utilizatorul poate achiziționa **Bilete** pentru **Evenimentele** organizate în respectiva perioada sau pentru diferitele **Galerii** ale muzeului.

Evenimentele reprezintă accesul neconditionat la anumite galerii din muzeu conform tematicii alese de organizatori. Totodată, orice eveniment are cel puțin un **Sponsor**. Orice sponsor poate să sponsorizeze mai multe evenimente.

Galeriile contin, bineînțeles, un număr variabil de **Exponate**. Aceste exponate sunt împărțite, la randul lor, în mai multe **Categorii**. Un exponat poate să apartină doar unei singure categorii.

Normal ca la fiecare galerie lucrează mai mulți **Angajați**. Aceștia sunt de două tipuri: **Ghizi** și **Supervizori**. Un utilizator poate să cumpere biletele să decida dacă va face turul împreună cu un ghid. Supervizorii îi sunt în subordine pe ghizi, însă și supervizorii sunt supervizorii lor. Un angajat este ca supervisor orice persoană ce se află deasupra lui în ierarhia conducerii Muzeului. Bineînțeles că ghizii sunt în punctul cel mai de jos al ierarhiei, motiv pentru care nu pot fi în supervizorii altor angajați.

Ghizii, galeriile și biletele sunt legate prin intermediul **Orarului**, deoarece toate vizitele sunt realizate împreună cu un ghid.

## II. Prezentarea constrângерilor (restrictii, reguli) impuse asupra modelului.

- Fiecare **Eveniment** trebuie să aibă cel puțin un **Sponsor**.
- Un **Exponat** poate apartine doar unei singure **Categorii**.
- Un **Utilizator** poate achiziționa bilete pentru mai multe **Evenimente** și/sau **Galerii**.
- Un **Angajat** poate fi **Ghid** sau **Supervizor**. Angajații pot avea și ei la randul lor alți supervisori, dar un **Ghid** nu poate avea alți angajați în subordine.

- Vizitele sunt realizate împreună cu un **Ghid**, la o anumita **Galerie**, aceste entități fiind conectate prin intermediul **Orarului**.

### III. Descrierea entităților, incluzând precizarea cheii primare.

#### 1. Utilizator:

- Cheie primara:** utilizator\_id
- Descriere:** Contul de utilizator al clientului ce contine diferite detalii despre acesta

#### 2. Galerie:

- Cheie primara:** galerie\_id
- Descriere:** Locatia unde obiectele de arta si de colectie sunt depozitate si expuse

#### 3. Eveniment:

- Cheie primara:** event\_id
- Descriere:** Proiecte ocasionale ce pot cuprinde mai multe galerii

#### 4. Sponsor:

- Cheie primara:** nume\_sponsor
- Descriere:** Cel ce depune majoritatea fondurilor pentru desfasurarea evenimentelor

#### 5. Exponat:

- Cheie primara:** exponat\_id
- Descriere:** Piesa de arta prezentata publicului contra cost

#### 6. Categorie:

- Cheie primara:** nume\_categorie
- Descriere:** Criteriu de distingere si separare pentru variile exponate

#### 7. Bilet:

- PK:** bilet\_id
- Descriere:** Document ce atesta permisiunea unei vizite
  - a. Bilet\_galerie:**
    - Cheie primara:** bilet\_id(FK)
    - Descriere:** Bilet ce permite intrarea într-o anumita galerie
  - b. Bilet\_eveniment:**
    - Cheie primara:** bilet\_id(FK)
    - Descriere:** Bilet ce permite participarea la un anumit eveniment

## 8.Angajat:

- **Cheie primara:** angajat\_id
- **Descriere:** Individ ce lucreaza in cadrul unei galerii
  - a. **Ghid:**
    - **Cheie primara:** angajat\_id (FK)
    - **Descriere:** Tip de angajat ce prezinta vizitatorilor exponatele
  - b. **Supervizor:**
    - **Cheie primara:** angajat\_id (FK)
    - **Descriere:** Tip de angajat ce are mai multe obligatii decat un ghid, dar care are in subordine acesti ghizi

## 9. Orar:

- **Cheie primara:** orar\_id
- **Descriere:** Liantul intre o galerie, un ghid si mai multe bilete

## IV. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

- **Bilet - Utilizator:**
  - **Cardinalitate:** Many to One (Un utilizator poate avea mai multe bilete)
- **Bilet\_galerie - Galerie:**
  - **Cardinalitate:** Many to One (Mai multe bilete de galerie pot sa permita intrarea la o galerie)
- **Bilet\_eveniment -Eveniment:**
  - **Cardinalitate:** Many to One (Mai multe bilete de evenimente pot sa permita participarea la un eveniment)
- **Galerie - Exponat:**
  - **Cardinalitate:** One to Many  
(O galerie poate avea mai multe exponate)
- **Categorie - Exponat:**
  - **Cardinalitate:** One to Many  
(O categorie poate avea mai multe exponate)
- **Angajat - Ghid/Supervizor:**
  - **Cardinalitate:** One to One (Un angajat poate fi ghid sau supervisor)
- **Orar - Angajat/Galerie:**

- **Cardinalitate:** Many to one (Un orar este legat de un ghid și de o galerie)
- **Galerie - Eveniment:**
  - **Cardinalitate:** Many to Many  
(O galerie poate să fie inclusă în mai multe evenimente și un eveniment poate include mai multe galerii)
- **Eveniment - Sponsor:**
  - **Cardinalitate:** Many to Many  
(Un eveniment poate avea mai mulți sponsori și un sponsor poate sprijini mai multe evenimente)

V. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicate, valori posibile ale atributelor.

#### 1. Entitate: Utilizator

- `utilizator_id`: integer, primary key, auto-increment
- `nume`: varchar(50), not null
- `prenume`: varchar(50), not null
- `data_nastere`: date, not null
- `telefon`: varchar(15), unique, not null
- `email`: varchar(100), unique, not null

#### 2. Entitate: Galerie

- `galerie_id`: integer, primary key, auto-increment
- `nume`: varchar(100), not null
- `nr_exponate`: integer, not null
- `permanenta`: varchar(1), default 'Y'
- `data_incheiere`: date, nullable

#### 3. Entitate: Eveniment

- `event_id`: integer, primary key, auto-increment
- `nume`: varchar(100), not null
- `data_inceput`: date, not null
- `data_incheiere`: date, not null
- `descriere`: text, nullable

#### 4. Entitate: Sponsor

- `nume_sponsor`: varchar(100), primary key

- **adresa\_sediu**: varchar(200), not null
- **fax**: varchar(20), unique, nullable
- **telefon**: varchar(15), unique, not null
- **nume\_CEO**: varchar(100), not null

**5. Entitate: Exponat**

- **exponat\_id**: integer, primary key, auto-increment
- **galerie\_id**: integer, FK referențiat de **Galerie(galerie\_id)**, not null
- **nume\_categorie**: varchar(100), FK referențiat de **Categorie(nume\_categorie)**, not null
- **nume**: varchar(100), not null
- **an**: integer, not null, check (**an** >= 0)
- **creator**: varchar(100), not null

**6. Entitate: Categorie**

- **nume\_categorie**: varchar(100), primary key
- **perioada**: varchar(50), not null
- **zona**: varchar(50), not null
- **descriere**: text, nullable

**7. Entitate: Bilet**

- **bilet\_id**: integer, primary key, auto-increment
- **utilizator\_id**: integer, FK referențiat de **Utilizator(utilizator\_id)**, not null
- **tip**: varchar(50), not null
- **orar\_id**: integer, FK referențiat de **Orar(orar\_id)**, not null
- **pret**: decimal(10, 2), not null, check (**pret** >= 0)

**8. Entitate: Angajat**

- **angajat\_id**: integer, primary key, auto-increment
- **galerie\_id**: integer, FK referențiat de **Galerie(galerie\_id)**, not null
- **tip**: varchar(50), check (**tip** in ('ghid', 'supervizor')), not null
- **vechime**: integer, not null, check (**vechime** >= 0)
- **nume**: varchar(100), not null
- **varsta**: integer, not null, check (**varsta** >= 18)
- **telefon**: varchar(15), unique, not null
- **email**: varchar(100), unique, not null

**9. Entitate: Ghid**

- **angajat\_id**: integer, FK referențiat de **Angajat(angajat\_id)**, not null
- **limba**: varchar(50), not null
- **salariu\_ghid**: decimal(10, 2), not null, check (**salariu** >= 0)

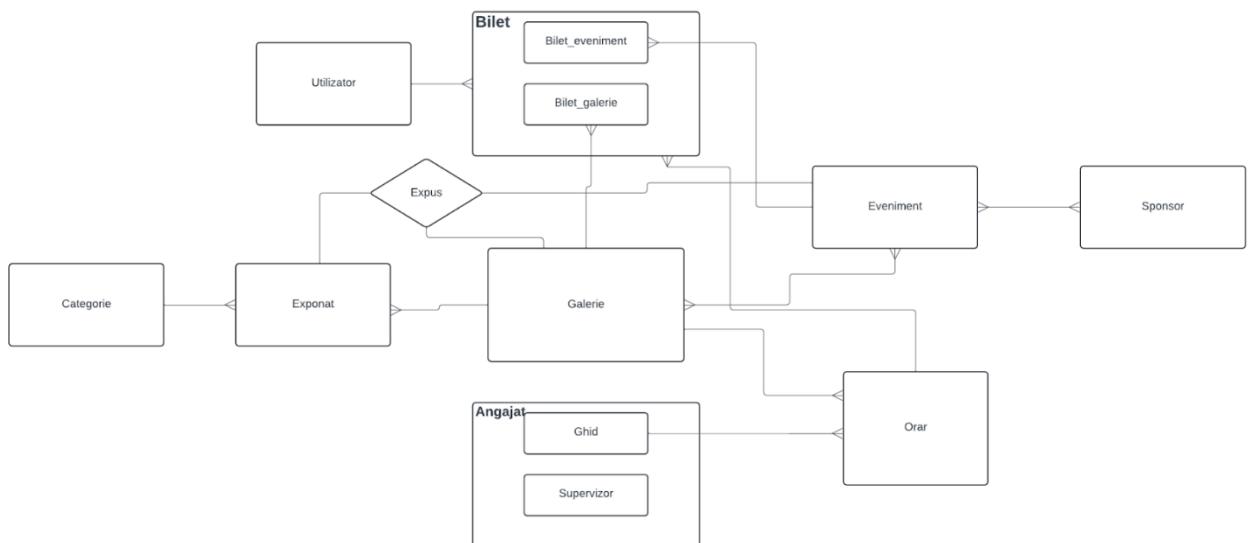
**10. Entitate: Supervizor**

- **angajat\_id**: integer, FK referențiat de **Angajat(angajat\_id)**, not null
- **pozitie**: varchar(100), not null
- **obligatii**: text, not null
- **salariu\_supervizor**: decimal(10, 2), not null, check (**salariu >= 0**)

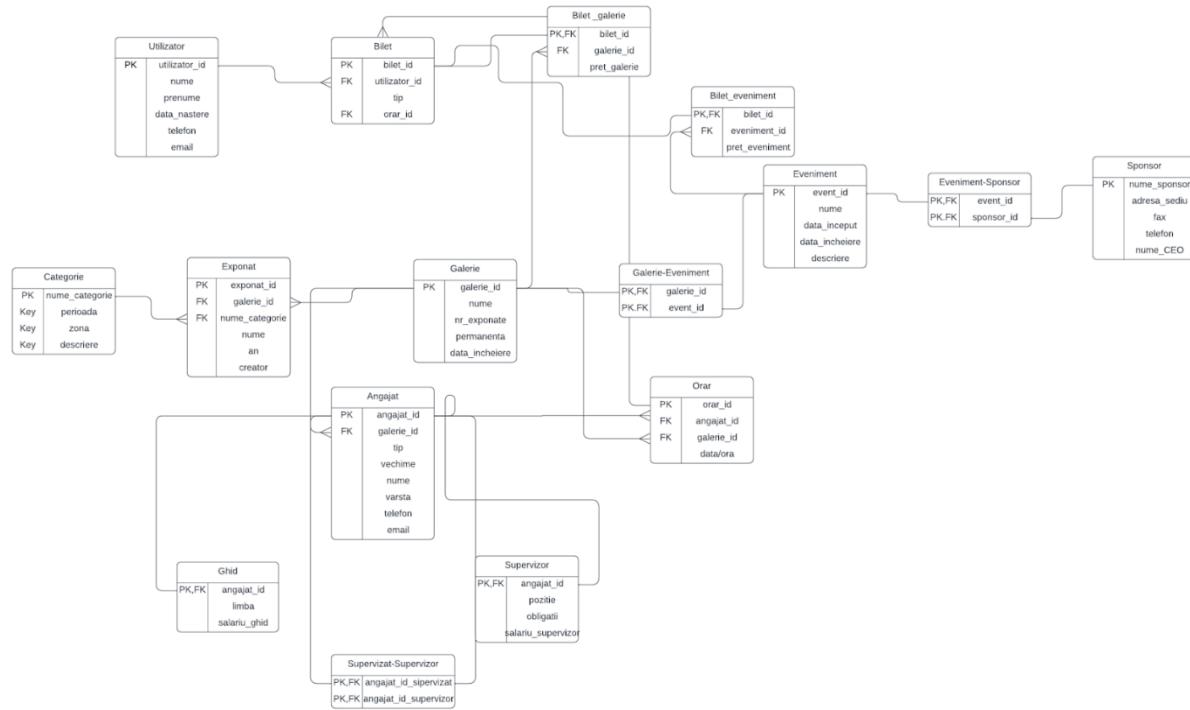
## 11. Relatie: Orar

- **orar\_id**: integer, primary key, auto-increment
- **angajat\_id**: integer, FK referențiat de **Angajat(angajat\_id)**, not null
- **galerie\_id**: integer, FK referențiat de **Galerie(galerie\_id)**, not null
- **data\_ora**: datetime, not null
- **Constrângeri**: Un orar specifică timpul și ghidul alocat pentru o vizită în galerie.

VI. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



VII. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



VIII. Enumerarea schemelor relationale corespunzătoare diagramei conceptuale proiectate la punctul 7.

1. **Utilizator** (utilizator\_id#, nume, prenume, data\_nastere, telefon, email)
  2. **Galerie** (galerie\_id#, nume, nr\_exponate, permanenta, data\_incheiere)
  3. **Eveniment** (event\_id#, name, data\_inceput, data\_incheiere, descriere)
  4. **Sponsor** (nume\_sponsor#, adresa\_sediu, fax, telefon, nume\_CEO)
  5. **Exponat** (exponat\_id#, galerie\_id#, nume\_categorie#, nume, an, creator)
  6. **Categorie** (nume\_categorie#, perioada, zona, descriere)
  7. **Bilet** (bilet\_id#, utilizator\_id#, tip, orar\_id#)

8. **Bilet\_galerie** (`bilet_id#`, `galerie_id#`, `pret_galerie`)
9. **Bilet\_Eveniment** (`bilet_id#`, `eveniment_id#`, `pret_eveniment`)
10. **Angajat** (`angajat_id#`, `galerie_id#`, `tip`, `vechime`, `nume`, `varsta`, `telefon`, `email`)
11. **Ghid** (`angajat_id#`, `limba`, `salariu_ghid`)
12. **Supervizor** (`angajat_id#`, `pozitie`, `obligatii`, `salariu_supervizor`)
13. **Orar** (`orar_id#`, `angajat_id#`, `galerie_id#`, `data_ora`)
14. **Supervizat\_Supervizor** (`angajat_id_supervizat#`, `angajat_id_sipervizor#`)
15. **Galerie\_Eveniment** (`galerie_ud#`, `event_id#`)
16. **Eveniment\_Sponsor** (`event_id#`, `nume_sponsor#`)

## IX. Realizarea normalizării până la forma normală 3 (FN1-FN3)

### 1. **Forma FN1**

Un tabel apartine FN1 daca are toate atributele atomice. Un exemplu pentru un astfel de tabel este  $R=\{event\_id, nume, descriere, data\_inceput, data\_incheiere, galerie\_id\}$  si o intrare:

<code>event_id</code>	<code>galerie_id</code>
1	1, 2

Rezolvare:

<code>event_id</code>	<code>galerie_id</code>
1	1
1	2

### 2. **Forma FN2**

Un exemplu de non-FN2 table este unul in care exista atribute non-cheie ce au dependente partiale de supercheie. Exemplu:

bilet_id	utilizator_id	galerie_id	data	nume_utilizator	pret_galerie	orar_id	tip
1	1	1	2024-05-18	Johnny Bravo	50	1	galerie
2	2	2	2024-05-18	Johnny Test	60	2	galerie

Rezolvare:

bilet_id	utilizator_id	data	nume_utilizator	orar_id	tip
1	1	2024-05-18	Johnny Bravo	1	galerie
2	2	2024-05-18	Johnny Test	2	galerie

bilet_id	galerie_id	pret_id
1	1	50
2	2	60

### 3. Forma FN3

O tabela se afla in FN3 daca:

1. Tabela este deja in FN2.
2. Atributele non-cheie nu au dependențe tranzitive, adică, fiecare atribut non-cheie depinde direct de cheia primară.

Exemplu: Daca in tabela bilet am adauga campul data, iar supercheia fiind (bilet\_id) ar aparea dependenta tranzitiva, deoarece bilet\_id->orar\_id, iar orar\_id->data. Din acest motiv cea mai buna solutie este

separarea tabelelor în aceeași formă precum în tabela conceptuală, date existând doar în tabelul Orar.

- X. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

```
CREATE SEQUENCE utilizator_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE galerie_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE event_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE exponat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE bilet_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE angajat_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE orar_seq START WITH 1 INCREMENT BY 1;
```

Sequence UTILIZATOR\_SEQ created.

Sequence GALERIE\_SEQ created.

Sequence EVENT\_SEQ created.

Sequence EXPONAT\_SEQ created.

Sequence BILET\_SEQ created.

Sequence ANGAJAT\_SEQ created.

Sequence ORAR\_SEQ created.

- XI. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

## 1. Crearea tabelelor

```
CREATE TABLE Utilizator (
    utilizator_id INT DEFAULT utilizator_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(50),
    prenume VARCHAR(50),
    data_nastere DATE,
    telefon VARCHAR(15),
    email VARCHAR(100) );
```

Script Output X | Task completed in 0.07 seconds

Table UTILIZATOR created.

```
CREATE TABLE Galerie (
    galerie_id INT DEFAULT galerie_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    nr_expoante INT,
    permanenta VARCHAR(1),
    data_incheiere DATE );
```

Script Output X | Task completed in 0.025 seconds

Table GALERIE created.

```
CREATE TABLE Eveniment (
    event_id INT DEFAULT event_seq.NEXTVAL PRIMARY KEY,
    nume VARCHAR(100),
    data_inceput DATE,
    data_incheiere DATE,
    descriere VARCHAR(300) );
```

Script Output X | Task completed in 0.026 seconds

Table EVENIMENT created.

```
CREATE TABLE Sponsor (
    nume_sponsor VARCHAR(100) PRIMARY KEY,
    adresa_sediu VARCHAR(255),
    fax VARCHAR(20),
    telefon VARCHAR(15),
    nume_CEO VARCHAR(100) );|
```

```
Script Output X | Task completed in 0.026 seconds
Table SPONSOR created.
CREATE TABLE Categorie (
    nume_categorie VARCHAR(100) PRIMARY KEY,
    perioada VARCHAR(50),
    zona VARCHAR(50),
    descriere VARCHAR(300) );|
```

```
Script Output X | Task completed in 0.025 seconds
Table CATEGORIE created.
CREATE TABLE Exponat(
    exponat_id INT DEFAULT exponat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    nume_categorie VARCHAR(100) REFERENCES Categorie(nume_categorie),
    nume VARCHAR(100),
    an INT,
    creator VARCHAR(100) );|
```

```
Script Output X | Task completed in 0.026 seconds
Table EXPONAT created.
```

```
CREATE TABLE Angajat (
    angajat_id INT DEFAULT angajat_seq.NEXTVAL PRIMARY KEY,
    galerie_id INT REFERENCES Galerie(galerie_id),
    tip VARCHAR(50),
    vechime INT,
    nume VARCHAR(100),
    varsta INT,
    telefon VARCHAR(15),
    email VARCHAR(100));
```

Script Output X | Task completed in 0.028 seconds

Table ANGAJAT created.

```
CREATE TABLE Ghid (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    limba VARCHAR(50),
    salariu_ghid NUMBER(10, 2));
```

Script Output X | Task completed in 0.022 seconds

Table GHID created.

```
CREATE TABLE Supervisor (
    angajat_id INT PRIMARY KEY,
    FOREIGN KEY(angajat_id) REFERENCES Angajat(angajat_id),
    pozitie VARCHAR(100),
    obligatii VARCHAR(4000),
    salariu_supervizor NUMBER(10, 2));
```

Script Output X | Task completed in 0.025 seconds

Table SUPERVIZOR created.

```
CREATE TABLE Orar (
    orar_id INT DEFAULT orar_seq.NEXTVAL PRIMARY KEY,
    angajat_id INT REFERENCES Angajat(angajat_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    data_orar DATE );
```

Script Output x | Task completed in 0.024 seconds

```
Table ORAR created.
CREATE TABLE Bilet (
    bilet_id INT DEFAULT bilet_seq.NEXTVAL PRIMARY KEY,
    utilizator_id INT REFERENCES Utilizator(utilizator_id),
    tip VARCHAR(50),
    orar_id INT REFERENCES Orar(orar_id));
```

Script Output x | Task completed in 0.021 seconds

```
Table BILET created.
CREATE TABLE Bilet_Galerie (
    bilet_id INT PRIMARY KEY ,
    FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
    galerie_id INT REFERENCES Galerie(galerie_id),
    pret_galerie NUMBER(10, 2) );
```

Script Output x | Task completed in 0.025 seconds

```
Table BILET_GALERIE created.
```

```
CREATE TABLE Bilet_Eveniment (
    bilet_id INT PRIMARY KEY ,
    FOREIGN KEY(bilet_id) REFERENCES Bilet(bilet_id),
    event_id INT REFERENCES Eveniment(event_id),
    pret_eveniment NUMBER(10, 2 ) ;
```

Script Output X | Task completed in 0.027 seconds

Table BILET\_EVENIMENT created.

```
CREATE TABLE Supervizat_Supervizor (
    angajat_id_supervizat INT,
    angajat_id_supervisor INT,
    PRIMARY KEY(angajat_id_supervizat,angajat_id_supervisor),
    FOREIGN KEY(angajat_id_supervizat) REFERENCES Angajat (angajat_id),
    FOREIGN KEY(angajat_id_supervisor) REFERENCES Angajat(angajat_id));
```

Script Output X

| Task completed in 0.019 seconds

Table SUPERVIZAT\_SUPERVIZOR created.

```
CREATE TABLE Galerie_Eveniment (
    galerie_id INT,
    event_id INT,
    PRIMARY KEY(galerie_id,event_id),
    FOREIGN KEY(galerie_id) REFERENCES Galerie (galerie_id),
    FOREIGN KEY(event_id) REFERENCES Eveniment(event_id));
```

Script Output X

| Task completed in 0.026 seconds

Table GALERIE EVENIMENT created.

```

CREATE TABLE Eveniment_Sponsor (
    event_id INT,
    nume_sponsor VARCHAR(100),
    PRIMARY KEY(event_id, nume_sponsor),
    FOREIGN KEY(event_id) REFERENCES Eveniment (event_id),
    FOREIGN KEY(nume_sponsor) REFERENCES Sponsor(nume_sponsor));

```

Script Output x  
Task completed in 0.022 seconds  
Table EVENIMENT\_SPONSOR created.

## 2. Inserarea datelor

### -- Inserarea în tabelul Utilizator

```

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Popescu', 'Ion', TO_DATE('1985-05-15', 'YYYY-MM-DD'), '0741234567', 'popion@yahoo.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Ionescu', 'Maria', TO_DATE('1990-07-20', 'YYYY-MM-DD'), '0742345678', 'maria9ionescu@gmail.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Georgescu', 'Alexandru', TO_DATE('1978-03-10', 'YYYY-MM-DD'), '0743456789', 'alexandru_georgescu@yahoo.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Vasilescu', 'Elena', TO_DATE('1982-11-25', 'YYYY-MM-DD'), '0744567890', 'elena.vasilescu@gmail.com');

INSERT INTO Utilizator (nume, prenume, data_nastere, telefon, email) VALUES
('Marin', 'Daniel', TO_DATE('2000-01-30', 'YYYY-MM-DD'), '0745678901', 'daniel100marin@yahoo.com');

```

UTILIZATOR_ID	NUME	PRENUME	DATA_NASTERE	TELEFON	EMAIL
1	Popescu	Ion	15-MAY-85	0741234567	popion@yahoo.com
2	Ionescu	Maria	20-JUL-90	0742345678	maria9ionescu@gmail.com
3	Georgescu	Alexandru	10-MAR-78	0743456789	alexandru_georgescu@yahoo.com
4	Vasilescu	Elena	25-NOV-82	0744567890	elena.vasilescu@gmail.com
5	Marin	Daniel	30-JAN-00	0745678901	daniel100marin@yahoo.com

### -- Inserarea în tabelul Sponsor

```

INSERT INTO Sponsor (nume_sponsor, adresa_sediul, fax, telefon, nume_CEO) VALUES
('Dacia', 'Calea Floreasca 166, Bucuresti', '0211234567', '0721123456', 'Christophe Dridi');

INSERT INTO Sponsor (nume_sponsor, adresa_sediul, fax, telefon, nume_CEO) VALUES
('BCR', 'Bd. Regina Elisabeta 5, Bucuresti', '0212345678', '0722234567', 'Sergiu Manea');

INSERT INTO Sponsor (nume_sponsor, adresa_sediul, fax, telefon, nume_CEO) VALUES
('Petrom', 'Str. Coraililor 22, Bucuresti', '0213456789', '0723345678', 'Christina Verchere');

INSERT INTO Sponsor (nume_sponsor, adresa_sediul, fax, telefon, nume_CEO) VALUES
('Bitdefender', 'Str. Barbu Vacarescu 54, Bucuresti', '0214567890', '0724456789', 'Florin Talpes');

INSERT INTO Sponsor (nume_sponsor, adresa_sediul, fax, telefon, nume_CEO) VALUES
('eMAG', 'Str. Bdul. Pierre de Coubertin 3-5, Bucuresti', '0215678901', '0725567890', 'Iulian Stanciu');

```

NUME_SPONSOR	ADRESA_SEDIU	FAX	TELEFON	NUME_CEO
1 Dacia	Calea Floreasca 166, Bucuresti	0211234567	0721123456	Christophe Dridi
2 BCR	Bd. Regina Elisabeta 5, Bucuresti	0212345678	0722234567	Sergiu Manea
3 Petrom	Str. Coralilor 22, Bucuresti	0213456789	0723345678	Christina Verchere
4 Bitdefender	Str. Barbu Vacarescu 54, Bucuresti	0214567890	0724456789	Florin Talpes
5 eMAG	Str. Bdul. Pierre de Coubertin 3-5, Bucuresti	0215678901	0725567890	Iulian Stanciu

## --Inserarea in Exponat

```

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
(1, 'Pictura sec XIX', 'Ciobanasi cu turma de oi', 1895, 'Nicolae Grigorescu');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
(2, 'Sculptura sec XIX', 'Domnitorul Alexandru Ioan Cuza', 1860, 'Ion Georgescu');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
(3, 'Arta Populara sec XX', 'Covor Oltenesc', 1920, 'Artisan Popular');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
(4, 'Fotografie sec XX', 'Revolutia din 1989', 1989, 'Fotoreporter Necunoscut');

INSERT INTO Exponat (galerie_id, nume_categorie, nume, an, creator) VALUES
(5, 'Arta Contemporana', 'Ingerul', 2005, 'Ion Barladeanu');

```

EXPOONAT_ID	GALERIE_ID	NUME_CATEGORIE	NUME	AN	CREATOR
1	6	1 Pictura sec XIX	Ciobanasi cu turma de oi	1895	Nicolae Grigorescu
2	7	2 Sculptura sec XIX	Domnitorul Alexandru Ioan Cuza	1860	Ion Georgescu
3	8	3 Arta Populara sec XX	Covor Oltenesc	1920	Artisan Popular
4	9	4 Fotografie sec XX	Revolutia din 1989	1989	Fotoreporter Necunoscut
5	10	5 Arta Contemporana	Ingerul	2005	Ion Barladeanu

## - Inserare Galerie

```

INSERT INTO Galerie (nume, nr_exponate, permanenta, data_incheiere) VALUES
('Galeria de Arta Moderna', 50, 'Y', NULL);

INSERT INTO Galerie (nume, nr_exponate, permanenta, data_incheiere) VALUES
('Galeria de Istorie Antica', 30, 'N', TO_DATE('2023-12-31', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, nr_exponate, permanenta, data_incheiere) VALUES
('Galeria de Stiinta', 40, 'Y', NULL);

INSERT INTO Galerie (nume, nr_exponate, permanenta, data_incheiere) VALUES
('Galeria de Arta Contemporana', 20, 'N', TO_DATE('2024-06-30', 'YYYY-MM-DD'));

INSERT INTO Galerie (nume, nr_exponate, permanenta, data_incheiere) VALUES
('Galeria de Fotografie', 25, 'Y', NULL);

```

GALERIE_ID	NUME	NR_EXPOONATE	PERMANENTA	DATA_INCHEIERE
1	1 Galeria de Arta Moderna	50	Y	(null)
2	2 Galeria de Istorie Antica	30	N	31-DEC-23
3	3 Galeria de Stiinta	40	Y	(null)
4	4 Galeria de Arta Contemporana	20	N	30-JUN-24
5	5 Galeria de Fotografie	25	Y	(null)

## - Inserare Eveniment

```

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Noaptea Muzeelor', TO_DATE('2024-05-18', 'YYYY-MM-DD'), TO_DATE('2024-05-18', 'YYYY-MM-DD'), 'Eveniment special cu acces gratuit.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Expozitia de Vara', TO_DATE('2024-06-01', 'YYYY-MM-DD'), TO_DATE('2024-08-31', 'YYYY-MM-DD'), 'Expozitie temporara de vara.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Festivalul de Arta', TO_DATE('2024-09-15', 'YYYY-MM-DD'), TO_DATE('2024-09-30', 'YYYY-MM-DD'), 'Festival de arta contemporana.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Saptamana Iстoriei', TO_DATE('2024-10-01', 'YYYY-MM-DD'), TO_DATE('2024-10-07', 'YYYY-MM-DD'), 'Saptamana dedicata istoriei.');

INSERT INTO Eveniment (nume, data_inceput, data_incheiere, descriere) VALUES
('Zilele Culorii', TO_DATE('2024-11-10', 'YYYY-MM-DD'), TO_DATE('2024-11-20', 'YYYY-MM-DD'), 'Eveniment dedicat tuturor tipurilor de pata de culoare.');

```

EVENT_ID	NUME	DATA_INCEPUT	DATA_INCHEIERE	DESCRIERE
1	Noaptea Muzeelor	18-MAY-24	18-MAY-24	Eveniment special cu acces gratuit.
2	Expozitia de Vara	01-JUN-24	31-AUG-24	Expozitie temporara de vara.
3	Festivalul de Arta	15-SEP-24	30-SEP-24	Festival de arta contemporana.
4	Saptamana Iстoriei	01-OCT-24	07-OCT-24	Saptamana dedicata istoriei.
5	Zilele Culorii	10-NOV-24	20-NOV-24	Eveniment dedicat tuturor tipurilor de pata de culoare.

## – Inserare Categorie

```

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Pictura sec XIX', '1800-1900', 'Europa de Est', 'Picturi din secolul XIX din Europa de Est');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Sculptura sec XIX', '1800-1900', 'Europa de Est', 'Sculpturi din secolul XIX din Europa de Est');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Arta Populara sec XX', '1900-2000', 'Romania', 'Obiecte de arta populara romaneasca din secolul XX');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Fotografie sec XX', '1900-2000', 'Romania', 'Fotografii istorice din Romania secolului XX');

INSERT INTO Categorie (nume_categoria, perioada, zona, descriere) VALUES
('Arta Contemporana', '2000-prezent', 'Romania', 'Lucrari de arta contemporana din Romania');

```

NUME_CATEGORIE	PERIOADA	ZONA	DESCRISERIE
1 Pictura sec XIX	1800-1900	Europa de Est	Picturi din secolul XIX din Europa de Est
2 Sculptura sec XIX	1800-1900	Europa de Est	Sculpturi din secolul XIX din Europa de Est
3 Arta Populara sec XX	1900-2000	Romania	Obiecte de arta populara romaneasca din secolul XX
4 Fotografie sec XX	1900-2000	Romania	Fotografii istorice din Romania secolului XX
5 Arta Contemporana	2000-prezent	Romania	Lucrari de arta contemporana din Romania

## – Inserare Angajat

```

INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 3, 'Ivan Ion', 32, '0741122334', 'ion_ivan13@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 5, 'Vladimirescu Cornel', 28, '0755566778', 'corneliusvlad@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 2, 'Grigorescu Ionel', 40, '0766677889', 'ionel-grigorita@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'ghid', 4, 'Negoita Florin', 35, '0777788990', 'florinita@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'ghid', 1, 'Dragulici Marin', 26, '0788899001', 'dragurini@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 6, 'Popa Vlad', 45, '0741000000', 'vladivici.popa@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 8, 'Veira Radu', 50, '0755000000', 'veiradu@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 3, 'Istratie Ionut', 35, '0766000000', 'ionut.istratie@yahoo.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (3, 'supervizor', 7, 'Stanciu Mario', 48, '0777000000', 'ggmariom@stanciu@gmail.com');
INSERT INTO Angajat (galerie_id, tip, vechime, nume, varsta, telefon, email) VALUES (1, 'supervizor', 2, 'Andreea Dumitrescu', 30, '0788000000', 'andreea.dumitrescu@yahoo.com');

```

ANGAJAT_ID	GALERIE_ID	TIPI	VECHIME	NUME	VARSTA	TELEFON	EMAIL
1	5	1 ghid	3	Ivan Ion	32	0741122334	ion_ivan13@yahoo.com
2	6	3 ghid	5	Vladimirescu Cornel	28	0755566778	corneliusvlad@gmail.com
3	7	1 ghid	2	Grigorescu Ionel	40	0766677889	ionel-grigorita@yahoo.com
4	8	1 ghid	4	Negoita Florin	35	0777788990	florinita@gmail.com
5	9	3 ghid	1	Dragulici Marin	26	0788899001	dragurini@yahoo.com
6	10	1 supervizor	6	Popa Vlad	45	0741000000	vladivici.popa@yahoo.com
7	11	1 supervizor	8	Veira Radu	50	0755000000	veiradu@gmail.com
8	12	3 supervizor	3	Istratie Ionut	35	0766000000	ionut.istratie@yahoo.com
9	13	3 supervizor	7	Stanciu Mario	48	0777000000	ggmariom@stanciu@gmail.com
10	14	1 supervizor	2	Andreea Dumitrescu	30	0788000000	andreea.dumitrescu@yahoo.com

## -- Inserarea în tabelul Ghid

```
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (5, 'Engleză', 3500.50);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (6, 'Germană', 3700.75);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (7, 'Franceză', 3400.00);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (8, 'Spaniolă', 3600.20);
INSERT INTO Ghid (angajat_id, limba, salariu_ghid) VALUES (9, 'Italiană', 3550.80);
```

ANGAJAT_ID	LIMBA	SALARIU_GHID
1	5 Engleză	3500.5
2	6 Germană	3700.75
3	7 Franceză	3400
4	8 Spaniolă	3600.2
5	9 Italiană	3550.8

## -- Inserarea în tabelul Supervisor

```
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (10, 'Supervisor de galerie', 'Coordonarea personalului, planificarea și organizarea evenimentelor și expozițiilor, gestionarea operațiunilor zilnice ale galeriei');
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (11, 'Curator', 'Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor', 4700.00);
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (12, 'Manager de muzeu', 'Supervizarea întregii activități a muzeului, elaborarea strategiilor de dezvoltare, gestionarea resurselor umane și financiare', 5000.00);
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (13, 'Conservator', 'Restaurarea și conservarea operelor de artă, elaborarea planurilor de conservare și restaurare, documentarea și cercetarea obiectelor de artă');
INSERT INTO Supervisor (angajat_id, pozitie, obligatii, salariu_supervisor)
VALUES (14, 'Manager Human Resources', 'Organizarea personalului și rezolvarea oricărora conflicte interne ce pot apărea de-a lungul timpului.', 4600.00);

ANGAJAT_ID POZITIE OBLIGATII
1 10 Supervisor de galerie Coordonarea personalului, planificarea și organizarea evenimentelor și expozițiilor, gestionarea operațiunilor zilnice ale galeriei
2 11 Curator Planificarea și organizarea expozițiilor, cercetarea și selectarea operelor de artă, gestionarea colecțiilor
3 12 Manager de muzeu Supervizarea întregii activități a muzeului, elaborarea strategiilor de dezvoltare, gestionarea resurselor umane și financiare
4 13 Conservator Restaurarea și conservarea operelor de artă, elaborarea planurilor de conservare și restaurare, documentarea și cercetarea obiectelor de artă
5 14 Manager Human Resources Organizarea personalului și rezolvarea oricărora conflicte interne ce pot apărea de-a lungul timpului.
```

## -- Inserarea în tabelul Orar

```
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (5, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (6, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (7, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (8, 1, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
INSERT INTO Orar (angajat_id, galerie_id, data_orar) VALUES (9, 3, TO_DATE('2024-05-18', 'YYYY-MM-DD'));
```

ORAR_ID	ANGAJAT...	GALERIE_ID	DATA_OR...
1	1	5	118-MAY-24
2	2	6	318-MAY-24
3	3	7	118-MAY-24
4	4	8	118-MAY-24
5	5	9	318-MAY-24

## - Inserare in Bilet

```
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 1);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (1, 'galerie', 2);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 3);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'galerie', 4);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (4, 'galerie', 5);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 1);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (2, 'eveniment', 2);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (3, 'eveniment', 3);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 4);
INSERT INTO Bilet (utilizator_id, tip, orar_id) VALUES (5, 'eveniment', 5);
```

BILET_ID	UTILIZATOR_ID	TIP	ORAR_ID
1	1	1 galerie	1
2	2	1 galerie	2
3	3	4 galerie	3
4	4	3 galerie	4
5	5	4 galerie	5
6	6	2 eveniment	1
7	7	2 eveniment	2
8	8	3 eveniment	3
9	9	5 eveniment	4
10	10	5 eveniment	5

#### -- Inserarea în tabelul Bilet\_Galerie

```
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (1, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (2, 3, 60.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (3, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (4, 1, 50.00);
INSERT INTO Bilet_Galerie (bilet_id, galerie_id, pret_galerie) VALUES (5, 3, 60.00);
```

BILET_ID	GALERIE_ID	PRET_GALERIE
1	1	1
2	2	3
3	3	1
4	4	1
5	5	3

#### -- Inserarea în tabelul Bilet\_Eventiment

```
INSERT INTO Bilet_Eventiment (bilet_id, event_id, pret_eventiment) VALUES (6, 1, 0.00);
INSERT INTO Bilet_Eventiment (bilet_id, event_id, pret_eventiment) VALUES (7, 1, 0.00);
INSERT INTO Bilet_Eventiment (bilet_id, event_id, pret_eventiment) VALUES (8, 5, 75.00);
INSERT INTO Bilet_Eventiment (bilet_id, event_id, pret_eventiment) VALUES (9, 3, 80.00);
INSERT INTO Bilet_Eventiment (bilet_id, event_id, pret_eventiment) VALUES (10, 4, 70.00);
```

BILET_ID	EVENT_ID	PRET_EVENTIMENT
1	6	1
2	7	1
3	8	5
4	9	3
5	10	4

#### -- Inserarea în tabelul Supervizat\_Supervizor

```

INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (5, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (7, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 10);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 11);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (8, 14);
INSERT INTO Supervizat_Supervizor (angajat_id_supervizat, angajat_id_supervizor) VALUES (6, 12);

```

	ANGAJAT_ID_SUPERVIZAT	ANGAJAT_ID_SUPERVIZOR
1	5	10
2	5	11
3	5	14
4	6	12
5	7	10
6	7	11
7	7	14
8	8	10
9	8	11
10	8	14

#### -- Inserarea în tabelul Galerie\_Eveniment

```

INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 5);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (1, 3);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (3, 4);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (4, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 1);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (2, 2);
INSERT INTO Galerie_Eveniment (galerie_id, event_id) VALUES (5, 4);

```

	GALERIE_ID	EVENT_ID
1	1	1
2	1	3
3	1	5
4	2	1
5	2	2
6	3	1
7	3	4
8	4	1
9	5	1
10	5	4

#### -- Inserarea în tabelul Eveniment\_Sponsor

```

INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'Petrom');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'Bitdefender');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (1, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (2, 'BCR');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (3, 'eMAG');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (4, 'Dacia');
INSERT INTO Eveniment_Sponsor (event_id, nume_sponsor) VALUES (5, 'Bitdefender');

```

EVENT_ID	NUME_SPONSOR
1	1 BCR
2	1 eMAG
3	2 BCR
4	2 Petrom
5	3 Bitdefender
6	3 eMAG
7	4 Dacia
8	4 eMAG
9	5 Bitdefender
10	5 Dacia

XII. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:

- ***Urmatoare cerere contine urmatorul subpunct:***

a) subcereri sincronizate în care intervin cel puțin 3 tabele;  
-grupari de date;

**Cerinta:**

Afisati numele, pozitia, id-ul si salariul mediu al galeriei din care fac parte supervisorii ce apartin unei galerii ce are atribuite mai putine orare decat supervisorul are oameni in subordine. Rezultatele vor fi grupate dupa numele supervisorului si id-ul galeriei.

**Query:**

```

SELECT a.numere, s.pozitie, AVG(s.salariu_supervizor) AS salariu_mediul
FROM Supervisor s
JOIN Angajat a ON s.angajat_id = a.angajat_id
WHERE a.galerie_id IN (
    SELECT g.galerie_id
    FROM Galerie g
    WHERE (
        SELECT COUNT(o.orar_id)
        FROM Orar o
        WHERE o.galerie_id = g.galerie_id
    ) < (
        SELECT COUNT(ss.angajat_id_supervizat)
        FROM Supervizat_Supervisor ss
        WHERE ss.angajat_id_supervizat = s.angajat_id
    )
)
GROUP BY a.numere, a.galerie_id, s.pozitie;

```

Autotrace x | Explain Plan x | Query Result x

SQL | All Rows Fetched: 1 in 0.019 seconds

NUME	POZITIE	SALARIU_MEDIU
Popa Vlad	Supervisor de galerie	4725

- Urmatoare cerere contine urmatorul subpunct:**

- d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- e) utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**Cerinta:**

Afişați informații despre evenimente și sponsori, inclusiv numele evenimentului, numele sponsorului, telefonul sponsorului, adresa sediului sponsorului (fără spații la început și sfârșit), anul de început al evenimentului și numărul de zile până la sfârșitul anului 2024 de la începutul datei evenimentului. În plus, trebuie să fie afișat și un status al anului care indică dacă anul evenimentului este curent (2024) sau anterior. În cazul în care telefonul sponsorului nu este disponibil, să se afișeze

"N/A". Rezultatele trebuie ordonate descrescător după nr de zile pana la sfarsitul anului și numele evenimentului.

### Query:

```

1 WITH EventSponsorInfo AS (
2     SELECT
3         e.event_id, e.nume AS nume_eveniment, s.numa_sponsor, NVL(s.telefon, 'N/A') AS telefon,
4         TRIM(both ' ' FROM s.adresa_sediu) AS adresa_sediu, EXTRACT(YEAR FROM e.data_inceput) AS an_inceput_eveniment, e.data_inceput
5         FROM Eveniment e JOIN Eveniment_Sponsor es ON e.event_id = es.event_id JOIN Sponsor s ON es.numa_sponsor = s.numa_sponsor
6     )
7     SELECT event_id, nume_eveniment, nume_sponsor, telefon, adresa_sediu, an_inceput_eveniment,
8         DECODE(an_inceput_eveniment, 2024, 'An curent', 'An anterior') AS status_an,
9         TO_CHAR(TO_DATE('2024-12-31', 'YYYY-MM-DD') - data_inceput, '999') || ' zile' AS zile_pana_la_sfarsitul_anului
10    FROM EventSponsorInfo
11   ORDER BY zile_pana_la_sfarsitul_anului DESC, nume_eveniment;

```

Autotrace X | Explain Plan X | Query Result X  
SQL | All Rows Fetched: 10 in 0.009 seconds

EVENT_ID	NUME_EVENIMENT	NUME_SPONSOR	TELEFON	ADRESA_SEDIU	AN_INCEPUT_EVENTIMENT	STATUS_AN	ZILE_PANA_LA_SFARSITUL_ANULUI
1	1 Noaptea Muzeelor	eMAG	0725567890	Str. Bdul. Pierre de Coubertin 3-5, Bucuresti	2024	An curent	227 zile
2	1 Noaptea Muzeelor	BCR	0722234567	Bd. Regina Elisabeta 5, Bucuresti	2024	An curent	227 zile
3	2 Expozitia de Vara	Petrom	0723345678	Str. Corailor 22, Bucuresti	2024	An curent	213 zile
4	2 Expozitia de Vara	BCR	0722234567	Bd. Regina Elisabeta 5, Bucuresti	2024	An curent	213 zile
5	3 Festivalul de Arta	Bitdefender	0724456789	Str. Barbu Vacarescu 54, Bucuresti	2024	An curent	107 zile
6	3 Festivalul de Arta	eMAG	0725567890	Str. Bdul. Pierre de Coubertin 3-5, Bucuresti	2024	An curent	107 zile
7	4 Saptamana Istoriei	Dacia	0721123456	Calea Floreasca 166, Bucuresti	2024	An curent	51 zile
8	4 Saptamana Istoriei	Dacia	0721123456	Calea Floreasca 166, Bucuresti	2024	An curent	51 zile
9	5 Zilele Culorii	Bitdefender	0724456789	Str. Barbu Vacarescu 54, Bucuresti	2024	An curent	51 zile
10	5 Zilele Culorii	Dacia	0721123456	Calea Floreasca 166, Bucuresti	2024	An curent	51 zile

- Urmatoare cerere contine urmatorul subpunct:**

- subcereri nesincronizate în clauza FROM
- grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (în cadrul aceleiași cereri)

### Cerinta:

Afisati informatiile despre galeriile care au mai puține orare decat numarul total de angajati din acele galerii. Includeti numele galeriei, numărul de orare și numărul total de angajati. Utilizati subcereri nesincronizate în clauza FROM pentru a obține numărul de angajați din fiecare galerie și filtrati la nivel de grupuri cu subcereri nesincronizate în clauza HAVING.

### Query:

```

1| SELECT
2|     g.nume AS nume_galerie,
3|     COUNT(o.orar_id) AS numar_orare,
4|     (SELECT COUNT(*) FROM Angajat a WHERE a.galerie_id = g.galerie_id) AS numar_angajati
5| FROM
6|     Galerie g
7| LEFT JOIN Orar o ON g.galerie_id = o.galerie_id
8| GROUP BY
9|     g.nume,
10|     g.galerie_id
11| HAVING
12|     COUNT(o.orar_id) < (
13|         SELECT COUNT(a.galerie_id)
14|         FROM Angajat a
15|         WHERE a.galerie_id = g.galerie_id
16|     );

```

Autotrace x Explain Plan x Query Result x

SQL | All Rows Fetched: 2 in 0.015 seconds

NUME_GALERIE	NUMAR_ORARE	NUMAR_ANGAJATI
1 Galeria de Arta Modernă	3	7
2 Galeria de Știință	2	4

- Urmatoare cerere contine urmatorul subpunct:**

- subcereri nesincronizate în clauza FROM
- grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (în cadrul aceleiași cereri)

### **Cerinta:**

Afişați media salariilor ghizilor care au mai mult de 4 ani vechime și lucrează în galerii care au cel puțin 10 exponate permanente. Includeți numele galeriei și media salariilor ghizilor. Utilizați subcereri nesincronizate în clauza FROM pentru a obține numărul de exponate permanente din fiecare galerie și filtrați la nivel de grupuri cu subcereri nesincronizate în clauza HAVING pentru a selecta ghizii cu mai mult de 4 ani vechime și galeriile cu cel puțin 10 exponate permanente.

### **Query:**

```

1 | SELECT
2 |     g.nume AS nume_galerie,
3 |     AVG(gh.salariu_ghid) AS media_salariilor_ghizilor
4 | FROM
5 |     Galerie g
6 | INNER JOIN Angajat a ON g.galerie_id = a.galerie_id
7 | INNER JOIN Ghid gh ON a.angajat_id = gh.angajat_id
8 | WHERE
9 |     g.nr_exponate >= 10
10 |    AND a.vechime > 4
11 | GROUP BY
12 |     g.nume
13 | HAVING
14 |     COUNT(g.galerie_id) >= 1;

```

Autotrace x Explain Plan x Query Result x

SQL | All Rows Fetched: 2 in 0.012 seconds

NUME_GALERIE	MEDIA_SALARIILOR_GHIZILOR
1 Galeria de Stiinta	3700.75
2 Galeria de Arta Moderna	3500

- ***Urmatoare cerere contine urmatorul subpunct:***

f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

**Cerinta:**

Afişați numărul total de bilete achiziționate de către utilizatori pentru evenimentele care au loc înainte de data curentă, împărtășite pe tipuri de bilete.

**Query:**

```

1 WITH BiletePerTip AS (
2     SELECT
3         b.tip,
4         COUNT(*) AS total_bilete
5     FROM
6         Bilet b
7     LEFT JOIN Bilet_Eveniment be ON b.bilet_id = be.bilet_id
8     LEFT JOIN Eveniment e ON be.event_id = e.event_id AND e.data_inceput < SYSDATE
9     WHERE
10        e.event_id IS NULL
11    GROUP BY
12        b.tip
13 )
14 SELECT
15     tip,
16     NVL(total_bilete, 0) AS total_bilete
17 FROM
18     BiletePerTip;

```

Autotrace x | Explain Plan x | Query Result x

SQL | All Rows Fetched: 2 in 0.012 seconds

NUME_GALERIE	MEDIA_SALARIIILOR_GHIZILOR
1 Galeria de Stiinta	3700.75
2 Galeria de Arta Moderna	3500

XIII. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

1. Actualizarea salariilor ghizilor ce lucrează în galeriile cu un numar de exponate mai mare decat 50, crescându-le salariul cu 10%.

```

1 UPDATE Ghid
2 SET salariu_ghid = salariu_ghid * 1.1
3 WHERE angajat_id IN (
4     SELECT angajat_id
5     FROM Angajat
6     WHERE galerie_id IN (
7         SELECT galerie_id
8         FROM Galerie
9         WHERE nr_exponate > 50
10    )
11 );

```

Autotrace x | Explain Plan x | Query Result x | S

SQL | Task completed in 0.098 seconds

0 rows updated.

2. Actualizarea numelui categoriei pentru toate exponatele care nu sunt inca asociate cu nicio categorie.

```
1 UPDATE Exponat
2 SET nume_categorie = 'Necunoscută'
3 WHERE NOT EXISTS (
4     SELECT *
5     FROM Categorie
6     WHERE Exponat.nume_categorie = Categorie.nume_categorie
7 );
8
9
10
```

Autotrace x Explain Plan x Query Result x Script Output x  
Task completed in 0.051 seconds

0 rows updated.

3. Actualizarea salariilor supervisorilor ce au subordonăți cu o vechime mai mică de 3 ani, crescându-le salariul cu 5%.

```
1 UPDATE Supervisor
2 SET salariu_supervisor = salariu_supervisor * 1.05
3 WHERE angajat_id IN (
4     SELECT angajat_id_supervisor
5     FROM Supervizat_Supervisor
6     INNER JOIN Angajat ON Supervizat_Supervisor.angajat_id_supervizat = Angajat.angajat_id
7     WHERE Angajat.vechime < 3
8 );
9
10
11
12
13
14
15
16
17
18
19
20
```

Autotrace x Explain Plan x Query Result x Script Output x  
Task completed in 0.039 seconds

3 rows updated.

4. Stergerea tuturor biletelor ce sunt achiziționate de utilizatori ce nu au nicio adresă de e-mail înregistrată.

```

1  DELETE FROM Bilet
2  WHERE utilizator_id IN (
3      SELECT utilizator_id
4      FROM Utilizator
5      WHERE email IS NULL
6  );
7
8
9
10
11
12
13
14
15
16
17
18
19
20

```

Autotrace x Explain Plan x | Query Result x  
 Task completed in 0.

0 rows deleted.

## 5. Stergerea tuturor biletelor pentru evenimentele ce nu au niciun sponsor.

```

1  DELETE FROM Bilet_Eventiment
2  WHERE event_id IN (
3      SELECT event_id
4      FROM Eveniment
5      WHERE NOT EXISTS (
6          SELECT *
7          FROM Eveniment_Sponsor
8          WHERE Eveniment.event_id = Eveniment_Sponsor.event_id
9      )
10 );
11 DELETE FROM Bilet
12 WHERE bilet_id IN (
13     SELECT bilet_id FROM Bilet_Eventiment
14     WHERE event_id IN (
15         SELECT event_id
16         FROM Eveniment
17         WHERE NOT EXISTS (
18             SELECT *
19             FROM Eveniment_Sponsor
20             WHERE Eveniment.event_id = Eveniment_Sponsor.event_id)));

```

Autotrace x Explain Plan x | Query Result x | Script Output x  
 Task completed in 0.038 seconds

0 rows deleted.

## 6. Stergerea tuturor categoriilor ce nu sunt asociate cu niciun exponent.

```

1: DELETE FROM Categorie
2: WHERE NOT EXISTS (
3:   SELECT *
4:   FROM Exponat
5:   WHERE Categorie.nume_categorie = Exponat.nume_categorie
6: );
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:

```

Autotrace x | Explain Plan x | Query Result x | Script Output x  
 Task completed in 0.035 seconds

0 rows deleted.

XIV. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

### **Crearea View-ului:**

```

CREATE VIEW DetaliiGalerii AS
WITH
    SupervizorCount AS (
        SELECT
            ss.angajat_id_supervizor,
            COUNT(ss.angajat_id_supervizat) AS numar_supervizati
        FROM Supervizat_Supervizor ss
        GROUP BY ss.angajat_id_supervizor
    ),
    OrarCount AS (
        SELECT
            o.galerie_id,
            COUNT(o.orar_id) AS numar_orare
        FROM Orar o
        GROUP BY o.galerie_id
    ),
    TotalBilete AS (
        SELECT
            b.utilizator_id,
            SUM(bg.pret_galerie) AS total_pret_galerie,

```

```

        SUM(bg.pret_galerie) AS total_pret_galerie,
        SUM(be.pret_veninment) AS total_pret_veninment
    FROM Bilet b
    LEFT JOIN Bilet_Galerie bg ON b.bilet_id = bg.bilet_id
    LEFT JOIN Bilet_Eveniment be ON b.bilet_id = be.bilet_id
    GROUP BY b.utilizator_id
)
SELECT
    a.angajat_id,
    a.numere AS numere_angajat,
    g.numere AS numere_galerie,
    CASE
        WHEN s.pozitie IS NULL AND gh.angajat_id IS NOT NULL THEN 'Ghid'
        ELSE s.pozitie
    END AS pozitie,
    NVL(s.salariu_supervizor, gh.salariu_ghid) AS salariu,
    TRIM(a.email) AS email_curat,
    SUBSTR(a.numere, 1, 3) AS numere_prescurtat,
    a.varsta,
    CASE
        WHEN a.varsta >= 50 THEN 'Senior'
        ELSE 'Junior'
    END AS categorie_varsta,
    NVL(sc.numar_supervizati, 0) AS numar_supervizati,
    NVL(oc.numar_orare, 0) AS numar_orare,
    NVL(tb.total_pret_galerie, 0) + NVL(tb.total_pret_veninment, 0) AS total_incasat
FROM
    Angajat a
LEFT JOIN Supervisor s ON a.angajat_id = s.angajat_id
LEFT JOIN Ghid gh ON a.angajat_id = gh.angajat_id
LEFT JOIN Galerie g ON a.galerie_id = g.galerie_id
LEFT JOIN SupervisorCount sc ON a.angajat_id = sc.angajat_id_supervizor
LEFT JOIN OrarCount oc ON a.galerie_id = oc.galerie_id
LEFT JOIN TotalBilete tb ON a.angajat_id = tb.utilizator_id;

```

Autotrace x | Explain Plan x | Query Result x | Script Output x

Task completed in 0.056 seconds

View DETALIIIGALERII created.

## Operatie LMD permisa:

```

1 INSERT INTO Angajat (angajat_id, galerie_id, tip, vechime, numere, varsta, telefon, email)
2 VALUES (15, 5, 'Ghid', 3, 'Anghel Luminita', 45, '0723623456', 'luminitaanghel98@yahoo.com');
3 INSERT INTO Ghid (angajat_id, limba, salariu_ghid)
4     VALUES (15, 'Japoneza', 3500);
5
6
7

```

Autotrace x | Explain Plan x | Query Result x | Script Output x

Task completed in 0.033 seconds

1 row inserted.

1 row inserted.

## Operatie LMD nepermisa:

```
1 UPDATE DetaliiGalerii
2 SET total_incasat = total_incasat + 100;
3
4
5
6
```

Autotrace x | Explain Plan x | Query Result x | Script Output  
Task completed in 0.079 seconds

Error at Command Line : 2 Column : 5  
Error report -  
SQL Error: ORA-01733: virtual column not allowed here  
01733. 00000 - "virtual column not allowed here"  
\*Cause:  
\*Action:

Cum total\_incasat este o coloana virtuala, adica calculata in interiorul view-ului, modificarea directa a acesteia este nepermisa, deoarece nu apartine niciunui tabel din view.

XV. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

### ***Outer-join pe minim 4 tabele:***

#### ***Cerinta:***

Obțineți o listă cu toate galeriile, inclusiv galeriile permanente și temporare, și pentru fiecare galerie includeți toate exponatele, angajații (supervizori și ghizi, dacă nu există va apărea mențiunea “Nu există”) și evenimentele asociate, chiar dacă unele dintre aceste elemente nu există pentru anumite galerii. De asemenea, indicați tipul galeriei (permanentă sau data de încheiere pentru cele temporare).

#### ***Query:***

1	SELECT
2	g.galerie_id,g.numa AS nume_galerie,g.nr_exponate,
3	CASE
4	WHEN g.data_incheiere IS NULL THEN 'Permanenta'
5	ELSE TO_CHAR(g.data_incheiere, 'DD-MM-YYYY')
6	END AS status_galerie,
7	e.exponat_id,e.numa AS nume_exponat,NVL(a.numa, 'Nu exista') AS nume_angajat,
8	CASE
9	WHEN s.angajat_id IS NOT NULL THEN 'Supervizor' WHEN gh.angajat_id IS NOT NULL THEN 'Ghid' ELSE 'Necunoscut'
10	END AS pozitie_angajat,
11	ev.event_id, ev.numa AS nume_eveniment
12	FROM
13	Galerie g
14	LEFT JOIN Exponat e ON g.galerie_id = e.galerie_id
15	LEFT JOIN Angajat a ON g.galerie_id = a.galerie_id
16	LEFT JOIN Supervizor s ON a.angajat_id = s.angajat_id
17	LEFT JOIN Ghid gh ON a.angajat_id = gh.angajat_id
18	LEFT JOIN Galerie_Eveniment ge ON g.galerie_id = ge.galerie_id
19	LEFT JOIN Eveniment ev ON ge.event_id = ev.event_id
20	ORDER BY g.galerie_id, e.exponat_id, a.angajat_id, ev.event_id;

1	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Ivan Ion	Ghid	1 Noapte
2	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Ivan Ion	Ghid	3 Festiv
3	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Ivan Ion	Ghid	5 Zilele
4	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Grigorescu Ionel	Ghid	1 Noapte
5	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Grigorescu Ionel	Ghid	3 Festiv
6	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Grigorescu Ionel	Ghid	5 Zilele
7	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Negoita Florin	Ghid	1 Noapte
8	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Negoita Florin	Ghid	3 Festiv
9	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Negoita Florin	Ghid	5 Zilele
10	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Popa Vlad	Supervizor	1 Noapte
11	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Popa Vlad	Supervizor	3 Festiv
12	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Popa Vlad	Supervizor	5 Zilele
13	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Veira Radu	Supervizor	1 Noapte
14	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Veira Radu	Supervizor	3 Festiv
15	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Veira Radu	Supervizor	5 Zilele
16	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Andreea Dumitrescu	Supervizor	1 Noapte
17	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Andreea Dumitrescu	Supervizor	3 Festiv
18	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Andreea Dumitrescu	Supervizor	5 Zilele
19	Moderna	50 Permanenta	6 Ciobană cu turma de oi	Popescu Maria	Ghid	1 Noapte

## Division:

### Cerinta:

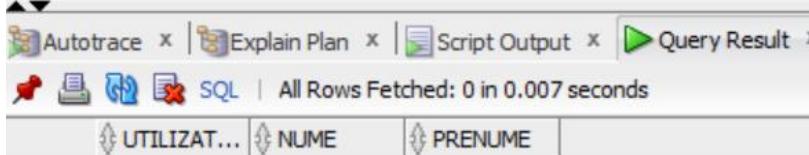
Selectează toți utilizatorii care au cumpărat bilete pentru toate evenimentele.

### Query:

```

1 | SELECT u.utilizator_id, u.nume, u.prenume
2 | FROM Utilizator u
3 | WHERE NOT EXISTS (
4 |   SELECT *
5 |   FROM Eveniment e
6 |   WHERE NOT EXISTS (
7 |     SELECT *
8 |     FROM Bilet b
9 |     WHERE b.utilizator_id = u.utilizator_id
10 |    AND EXISTS (
11 |      SELECT *
12 |      FROM Bilet_Eveniment be
13 |      WHERE be.bilet_id = b.bilet_id
14 |      AND be.event_id = e.event_id
15 |    )
16 |  )
17 | );
18 |
19 |
20 |

```



### **Analiza top-n:**

#### **Cerinta:**

Afișează top 2 galerii cu cel mai mare salariu mediu al angajaților, ținând cont de salariile ghizilor și supervisorilor.

#### **Query:**

```

1 WITH Salarii AS (
2     SELECT a.galerie_id, g.salariu_ghid AS salariu
3     FROM Ghid g
4     JOIN Angajat a ON g.angajat_id = a.angajat_id
5     UNION ALL
6     SELECT a.galerie_id, s.salariu_supervizor AS salariu
7     FROM Supervizor s
8     JOIN Angajat a ON s.angajat_id = a.angajat_id
9 )
10 SELECT g.galerie_id, g.nume AS nume_galerie, AVG(s.salariu) AS salariu_mediul
11 FROM Galerie g
12 JOIN Salarii s ON g.galerie_id = s.galerie_id
13 GROUP BY g.galerie_id, g.nume
14 ORDER BY salariu_mediul DESC
15 FETCH FIRST 2 ROWS ONLY;
16
17
18
19
20

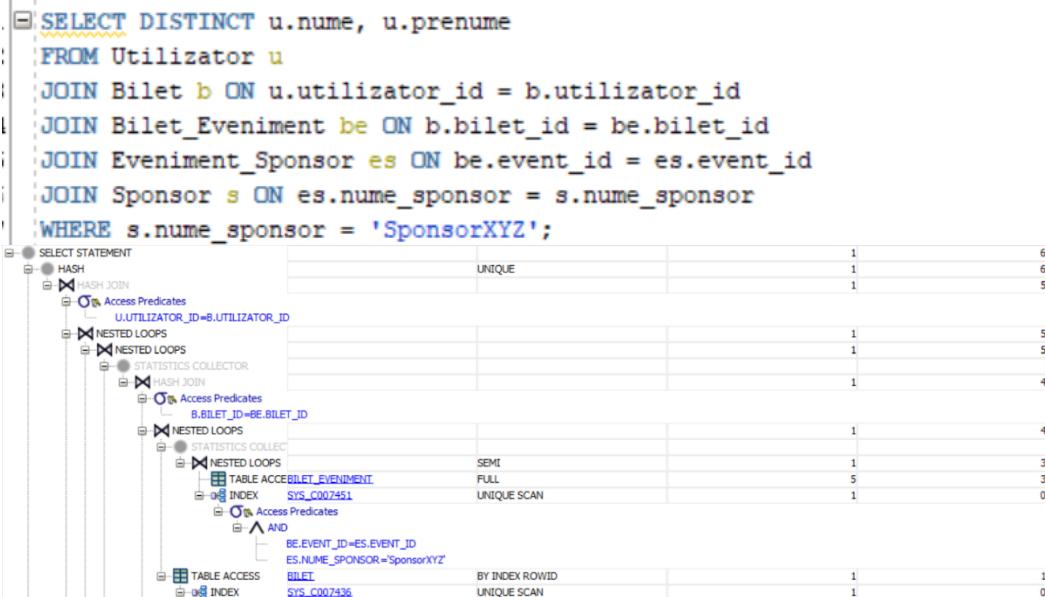
```

The screenshot shows the SQL tab with the query executed. The results are displayed in a table:

GALERIE_ID	NUME_GALERIE	SALARIU_MEDIUL
1	Galeria de Arta Modernă	4323.61
2	Galeria de Știință	4262.8875

XVI. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.

### Cost mic:



	Access Predicates				
	B.BILET_ID=B.BILET_ID				
TABLE ACCESS	BILET	FULL		1	1
INDEX	SYS_C007419	UNIQUE SCAN		1	0
	Access Predicates				
	U.USUARIOS_ID=B.USUARIOS_ID				
TABLE ACCESS	USUARIOS	BY INDEX ROWID		1	1
TABLE ACCESS	USUARIOS	FULL		1	1

Initial se face un Nested Loop intre Bilet\_Eveniment si Eveniment\_Sponsor, urmand ca rezultatul sa fie folosit in alt Nested Loop impreuna cu tabelul Bilet. Apoi, ce rezulta din acea operatie va fi folosit intr-un Hash Join cu biletele ce sunt pentru evenimente, urmand ca rezultatele sa fie utilizate in doua Nested Loop: unul cu Bilet, iar apoi cu utilizatorii ce au cumparat bilete.

### Cost mare:

```

SELECT DISTINCT u.nume, u.prenume
FROM Utilizator u
JOIN Bilet b ON u.utilizator_id = b.utilizator_id
JOIN (
    SELECT be.bilet_id, b.utilizator_id
    FROM Bilet_Eveniment be
    JOIN Bilet b ON be.bilet_id = b.bilet_id
    JOIN Eveniment_Sponsor es ON be.event_id = es.event_id
    WHERE es.numere_sponsor = 'SponsorXYZ'
) sub ON u.utilizator_id = sub.utilizator_id;

```

	SELECT STATEMENT				
	HASH				
	HASH JOIN				
	Access Predicates				
	U.USUARIOS_ID=B.USUARIOS_ID				
	NESTED LOOPS				
	NESTED LOOPS				
	STATISTICS COLLECTOR				
	NESTED LOOPS				
	MERGE JOIN				
	TABLE ACCESS	BILET	BY INDEX ROWID	10	2
	INDEX	SYS_C007436	FULL SCAN	10	1
	SORT		JOIN	5	4
	Access Predicates				
	BE.BILET_ID=B.BILET_ID				
	Filter Predicates				
	BE.BILET_ID=B.BILET_ID				
	TABLE ACCESS	BILET_EVENTIMENT	FULL	5	3
	INDEX	SYS_C007451	UNIQUE SCAN	1	0
	Access Predicates				
	AND				
	BE.EVENT_ID=ES.EVENT_ID				
	ES.NUMERE_SPONSOR='SponsorXYZ'				
	INDEX	SYS_C007419	UNIQUE SCAN	1	0
	Access Predicates				
	U.USUARIOS_ID=B.USUARIOS_ID				
TABLE ACCESS	USUARIOS	BY INDEX ROWID		1	1
TABLE ACCESS	USUARIOS	FULL		1	1
TABLE ACCESS	BILET	FULL		10	3

Aici se foloseste un Merge Join intre Bilet si biletele ce sunt pentru evenimente, rezultatul fiind apoi introdus intr-un Nested Loop. Aceste informatii sunt colectate de Statistics Collector impreuna cu id-urile utilizatorilor ce au achizitionat bilete, iar apoi introduse in doua Nested Loops succesive ambele dati cu tabelul utilizator. Ce rezulta este introdus

apoi intr-un Hash join cu Bilet, rezultatul Hash joined cu id-ul utilizatorilor ce au bilet.