Model al bazei de date - Galerie de artă

Silaghi Mara
Grupa 141

Cuprins

1.Descrierea modelului real

2.Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

3. Descrierea entităților

4. Descrierea relațiilor

5. Descrierea atributelor

6. Realizarea diagramei entitate-relație

7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație

8. Enumerarea schemelor relaționale

9.Aducerea la forma normală

10/11. SQL - CREATE, INSERT

12. Cereri SQL

13. Operații de actualizare / ștergere

14. Vizualizare complexă

15.Cereri opționale

16. Optimizare

17. Normalizare

18. Tranzacții

19. Index

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

Baza de date a unei galerii de artă ar putea cuprinde informații despre diferite entități și relații importante pentru gestionarea operei de artă, a artiștilor și a clienților. Printre entități se numără operele de artă, artiștii, colecționarii, angajații și vizitatorii. Pentru fiecare dintre aceste entități se pot colecta informații specifice, cum ar fi numele, adresa, prețul .

În ceea ce privește relațiile, acestea pot fi stabilite între entități, cum ar fi faptul că operele de artă sunt create de artiști sau că o expoziție cuprinde una sau mai multe opere. De asemenea, colecționarii pot achiziționa opere de artă de la galerie, iar angajații pot fi responsabili de administrarea și gestionarea operelor de artă și a relațiilor cu clienții. În plus, vizitatorii pot achiziționa bilete și pot lua parte la expozițiile galeriei.

Astfel, o bază de date a unei galerii de artă poate fi un instrument valoros pentru a gestiona cu succes opera de artă, artiștii, colecționarii și clienții, oferind o perspectivă mai clară și organizată asupra activităților galeriei de artă.

2.Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

- O expoziție poate avea 0 sau mai mulți vizitatori;
- Se pot împrumuta una sau mai multe opere pentru a fi expuse;
- O galerie poate organiza una sau mai multe expoziții;
- O galerie are unul sau mai mulți angajați;
- Un bilet poate fi cumpărat de către un singur vizitator;

  etc.

3. Descrierea entităților, incluzând precizarea cheii primare.

Entitate	**Cheie primară**	**Observații**
Angajat	id_angajat	Conține informații generale despre angajați
Director	id_angajat	Conține informații specifice referitoare la postul de director
Galerie	id_galerie	Conține informații legate de galerie: nume, locație etc.
Vizitator	id_vizitator	Reține date legate de vizitatorii expozițiilor: nume, vârstă etc.

etc.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Relație	**Cardinalitate**	**Observații**
cumpără	Operă - Colecționar one-to-many	Un colecționar poate cumpăra una sau mai multe opere. O operă poate fi cumpărată de un singur colecționar.
expune	Expoziție - Operă many-to-many	O expoziție poate expune una sau mai multe opere. O operă poate fi expusă în 0 sau mai multe expoziții.
lucrează	Angajat - Galerie one-to-many	Un angajat poate lucra în cadrul unei singure galerii. Într-o galerie lucrează unul sau mai mulți angajați.
cumpără	Bilet - Vizitator one-to-many	Un vizitator poate cumpăra unul sau mai multe bilete. Un bilet poate fi cumpărat de un singur vizitator.

etc.

5. Descrierea atributelor, incluzând tipul de date şi eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Entitate: Operă

Atribut	**Tip**	**Dimensiune**	**Valori posibile**	**Observații**
titlu	string	20		
id_artist	int	3		Legătura artist - operă
tip	string	9	sculptură/pictură	
an_creație	int	4		

Entitate: Bilet

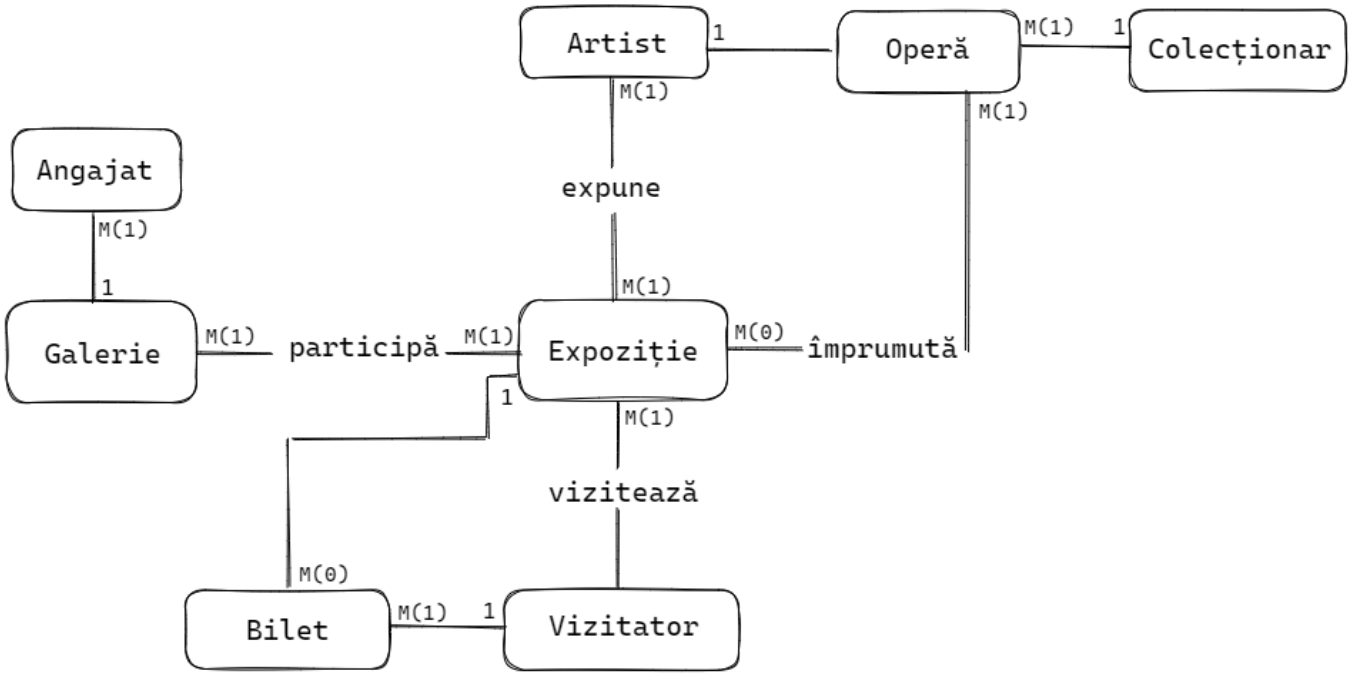
Atribut	**Tip**	**Dimensiune**	**Valori posibile**	**Observații**
preț	int	3		
id_vizitator	int	3		Legătura vizitator - bilet
tip	string	6	întreg/redus	
id_galerie	int	3		

Entitate: Colecționar

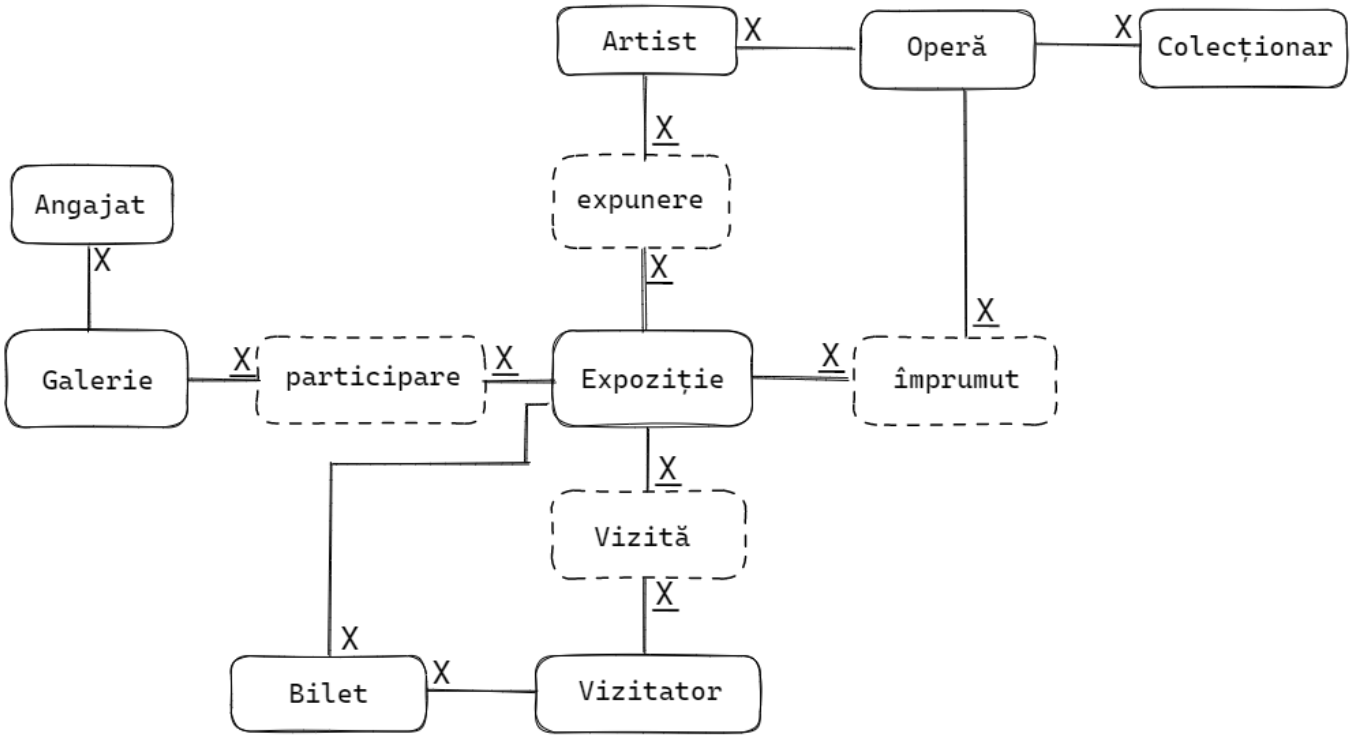
Atribut	Tip	Dimensiune	Valori posibile	Observații
nume	string	10		
prenume	string	10		
adresa_mail	string	30		
numar_telefon	string	10		

etc.

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



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



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

- **Opera**: id_opera, titlu, id_artist, tip, an_creatie
- **Artist**: id_artist, nume, data nașterii, naționalitate
- **Colectionar**: id_colectionar, nume, prenume, telefon, adresa_mail, id_opera
- **Galerie**: id_galerie, denumire, locație
- **Expoziții**: id_expozitie, nume_expozitie, data_inceput, data_final
- **Angajat**: id_angajat, nume, poziție, telefon, adresa_mail, programul de lucru, id_galerie
- **Vizitator**: id_vizitator, nume, prenume, adresa_mail
- **Bilet**: id_bilet, pret, data_vizita, id_vizitator, id_expozitie
- **Expunere**: id_expozitie, id_artist
- **Participare**: id_expozitie, id_galerie
- **Vizita**: id_expozitie, id_vizitator
- **Imprumut**: id_expozitie, id_opera, data_imprumut

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

• Forma normală 1 (FN1)

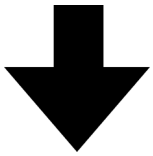
O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizabilă.

Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel încât să fie identificată unic prin intermediul unei chei primare.
Voi lua entitățile **EXPOZITIE** și **OPERĂ**.
În cadrul unei expoziții, denumirea operelor nu se repetă, dar în cadrul unei galerii, acest lucru nu este garantat.

Expoziție	**Operă**
Between Worlds and Traditions	The Two Fridas
Gardens	Bird in Space, Water Lilies
One Night Exhibition	Starry Night, The Thinker, Water Lilies
Retrospectiva	The Two Fridas
Semne și simboluri	Pieta, Starry Night, Guernica

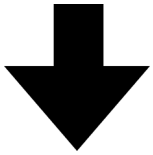
Reținerea datelor în modul de mai sus ar fi complicat și ineficient. Spre exemplu, o interogare care ar selecta acele expoziții care expun și opera "*The Two Fridas*", și opera "*Water Lillies*", ar trebui să parcurgem fiecare șir "Operă", să identificăm subșirurile "*The Two Fridas*" și "*Water Lillies*" și să selectăm numai acele înregistrări în care apar ambele subșiruri.

Expoziție	**Operă**
Between Worlds and Traditions	The Two Fridas
Gardens	Bird in Space, Water Lilies
One Night Exhibition	Starry Night, The Thinker, Water Lilies
Retrospectiva	The Two Fridas
Semne și simboluri	Pieta, Starry Night, Guernica



Expoziție	**Operă**
Between Worlds and Traditions	The Two Fridas
Gardens	Bird in Space
Gardens	Water Lilies

One Night Exhibition	Starry Night
One Night Exhibition	The Thinker
One Night Exhibition	Water Lilies
Retrospectiva	The Two Fridas
Semne și simboluri	Pieta
Semne și simboluri	Starry Night
Semne și simboluri	Guernica



ID	Expoziție	Operă
1	Between Worlds and Traditions	The Two Fridas
2	Gardens	Bird in Space
3	Gardens	Water Lilies
4	One Night Exhibition	Starry Night
5	One Night Exhibition	The Thinker
6	One Night Exhibition	Water Lilies
7	Retrospectiva	The Two Fridas
8	Semne și simboluri	Pieta
9	Semne și simboluri	Starry Night
10	Semne și simboluri	Guernica

Pentru a asigura unicitatea unei înregistrări, se va utiliza cheia primară. În exemplul de mai sus, prin introducerea unei coloane adiționale de tip întreg se asigura unicitatea fiecărei înregistrări.

- **Forma Normală 2 (FN2)**

O relație se află în a doua formă normală dacă și numai dacă această relație este deja în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Dacă unul sau mai multe elemente sunt dependente funcțional numai de o parte a cheii primare, atunci ele trebuie sa fie separate în tabele diferite. Dacă tabela are o cheie primară formată din numai un atribut, atunci ea este automat în FN2.

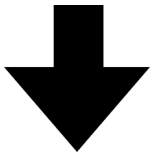
Pentru baza mea de date, voi exemplifica pentru cazul diagramei **EXPUNERE**.

ID_Expozitie	ID_Artist	Nume_Artist
1	1	Vincent Van Gogh
1	6	Michelangelo
1	5	Frida Kahlo
2	4	Claude Monet

Relația este în FN1 – avem identificator unic pentru toate intrările din table. În cazul nostru, atributul Nume_Artist nu este cheie și trebuie sa depindă direct de întreaga cheie primară id_expozitie# si id_artist#. Se observă dependența directă dintre Nume_Artist și ID_Artist, însemnând că Nume_Artist depinde direct doar de o parte a cheii primare, și anume doar de ID_Artist, astfel relația nu se află în FN2.

Deci, pentru a avea relația în FN2, numele artistului trebuie să fie doar în entitatea Artist.

ID_Expozitie	ID_Artist	Nume_Artist
1	1	Vincent Van Gogh
1	6	Michelangelo
1	5	Frida Kahlo
2	4	Claude Monet



ID_Expozitie	ID_Artist
1	1
1	6
1	5

2	4

- **Forma Normală 3 (FN3)**

O relație este în a treia formă normală dacă și numai dacă este in FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

Iau tabela **COLECȚIONAR**. Inițial, aceasta ar fi fost de forma:

ID_colecționar	Nume	Prenume	Telefon	Titlu_operă	An_creație_operă
1	Dîrjan	Alexandra	0701073147	Mona Lisa	1503
2	Nițu	Paul	0248292876	Guernica	1937
3	Marin	Andrei	0785282188	The Two Fridas	1939
4	Cristea	Ionuț	0270491235	Starry Night	1889

Se poate observa că atributul An_creatie_operă depinde de atributul Titlu_operă, care depinde la rândul său de cheia primară ID_colecționar, astfel nefiind in FN3.

Pentru a aduce in FN3, separ atributele despre operă din COLECȚIONAR, apărând astfel tabela OPERĂ.

Astfel, înlocuiesc aceste atribute cu cheia străină ID_Operă (apărând astfel o relație one-to-many între operă și colecționar) pentru a determina mai ușor atributele operei cumpărate de un colecționar.

ID_colecționar	Nume	Prenume	Telefon	ID_operă
1	Dîrjan	Alexandra	0701073147	1
2	Nițu	Paul	0248292876	3
3	Marin	Andrei	0785282188	5
4	Cristea	Ionuț	0270491235	2

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele +

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).

```
CREATE TABLE Artist (
  id_artist INT PRIMARY KEY,
  nume VARCHAR(255) NOT NULL,
  data_nasterii DATE,
  nationalitate VARCHAR(255)
);



INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (1, 'Leonardo da Vinci', TO_DATE('1452-04-15', 'YYYY-MM-DD'), 'italian');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (2, 'Vincent van Gogh', TO_DATE('1853-03-30', 'YYYY-MM-DD'), 'olandez');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (3, 'Pablo Picasso', TO_DATE('1881-10-25', 'YYYY-MM-DD'), 'spaniol');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (4, 'Claude Monet', TO_DATE('1840-11-14', 'YYYY-MM-DD'), 'francez');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (5, 'Frida Kahlo', TO_DATE('1907-07-06', 'YYYY-MM-DD'), 'mexican');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (6, 'Michelangelo', TO_DATE('1475-03-06', 'YYYY-MM-DD'), 'italian');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (7, 'Auguste Rodin', TO_DATE('1840-11-12', 'YYYY-MM-DD'), 'francez');

INSERT INTO Artist (id_artist, nume, data_nasterii, nationalitate)
VALUES
  (8, 'Constantin Brancusi', TO_DATE('1876-02-19', 'YYYY-MM-DD'), 'roman');
```

	ID_ARTIST	NUME	DATA_NASTERII	NATIONALITATE
1	1	Leonardo da Vinci	15-APR-52	italian
2	2	Vincent van Gogh	30-MAR-53	olandez
3	3	Pablo Picasso	25-OCT-81	spaniol
4	4	Claude Monet	14-NOV-40	francez
5	5	Frida Kahlo	06-JUL-07	mexican
6	6	Michelangelo	06-MAR-75	italian
7	7	Auguste Rodin	12-NOV-40	francez
8	8	Constantin Brancusi	19-FEB-76	roman

```sql
CREATE TABLE Opera (
  id_opera INT PRIMARY KEY,
  titlu VARCHAR(255) NOT NULL,
  id_artist INT,
  tip VARCHAR(255) NOT NULL,
  an_creatie INT,
  id_colectionar INT,
  FOREIGN KEY (id_artist) REFERENCES Artist(id_artist),
  FOREIGN KEY (id_colectionar) REFERENCES Artist(id_colectionar)
);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (1, 'Mona Lisa', 1, 'Pictura', 1503, 1);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (2, 'Starry Night', 2, 'Pictura', 1889, 2);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (3, 'Guernica', 3, 'Pictura', 1937, 1);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (4, 'Water Lilies', 4, 'Pictura', 1919, 3);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (5, 'The Two Fridas', 5, 'Pictura', 1939, 4);
```

```
INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (6, 'David', 6, 'Sculptura', 1504, 2);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (7, 'The Thinker', 7, 'Sculptura', 1902, NULL);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (8, 'Pieta', 6, 'Sculptura', 1499, NULL);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (9, 'Bird in Space', 8, 'Sculptura', 1923, 3);

INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (10, 'The Kiss', 7, 'Sculptura', 1889, 5);
INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (11, 'Cina cea de Taina', 1, 'Pictura', NULL, 2);
INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (12, 'Peisaj', 5, 'Pictura', 1869, 1);
INSERT INTO Opera (id_opera, titlu, id_artist, tip, an_creatie, id_colectionar)
VALUES
  (13, 'The Thinker', 5, 'Pictura', 1869, 4);
```

	ID_OPERA	TITLU	ID_ARTIST	TIP	AN_CREATIE
1	1	Mona Lisa	1	Pictura	1503
2	2	Starry Night	2	Pictura	1889
3	3	Guernica	3	Pictura	1937
4	4	Water Lilies	4	Pictura	1919
5	5	The Two Fridas	5	Pictura	1939
6	6	David	6	Sculptura	1504
7	7	The Thinker	7	Sculptura	1902
8	8	Pieta	6	Sculptura	1499
9	9	Bird in Space	8	Sculptura	1923
10	10	The Kiss	7	Sculptura	1889
11	11	Cina cea de Taina	1	Pictura	(null)
12	12	Peisaj	1	Pictura	1898
13	13	Peisaj	1	Pictura	1898

```
CREATE TABLE Colectionar (
  id_colectionar INT PRIMARY KEY,
  nume VARCHAR(255) NOT NULL,
  prenume VARCHAR(255) NOT NULL,
  telefon VARCHAR(255),
  adresa_mail VARCHAR(255) NOT NULL,
```

```
);

INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (1, 'Dîrjan', 'Alexandra', '0701073147', 'dirjan.alexandra@gmail.com', 1);


INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (2, 'Nitu', 'Paul', '0248292876', 'nitu.paul@gmail.com', 3);


INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (3, 'Marin', 'Andrei', '0785282188', 'marin.andrei@gmail.com', 5);


INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (4, 'Cristea', 'Ionut', '0270491235', 'cristea.ionut@gmail.com', 2);


INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (5, 'Dumitru', 'Clara', '0264719906 ', 'dumitru.clara@gmail.com', 4);


INSERT INTO Colectionar (id_colectionar, nume, prenume, telefon, adresa_mail)
VALUES
  (6, 'Tumulescu', 'Andreea', '0365884320  ', 'tumulescu.andreea@gmail.com', 7);
```

	ID_COLECTIONAR	NUME	PRENUME	TELEFON	ADRESA_MAIL	ID_OPERA
1	1	Dîrjan	Alexandra	0701073147	dirjan.alexandra@gmail.com	1
2	2	Nitu	Paul	0248292876	nitu.paul@gmail.com	3
3	3	Marin	Andrei	0785282188	marin.andrei@gmail.com	5
4	4	Cristea	Ionut	0270491235	cristea.ionut@gmail.com	2
5	5	Dumitru	Clara	0264719906	dumitru.clara@gmail.com	4
6	6	Tumulescu	Andreea	0365884320	tumulescu.andreea@gmail.com	7

```
CREATE TABLE Galerie (
  id_galerie INT PRIMARY KEY,
  denumire VARCHAR(255) NOT NULL,
  locatie VARCHAR(255)
);
```

```
INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (1, 'Galeria Nationala', 'Bucuresti');

INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (2, 'Museum of Modern Art', 'New York');

INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (3, 'Louvre Museum', 'Paris');

INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (4, 'Tate Modern', 'Londra');

INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (5, 'Uffizi Gallery', 'Florenta');
INSERT INTO Galerie (id_galerie, denumire, locatie)
VALUES
  (6, 'Museum of Modern Art', 'Bruxelles');
```

	ID_GALERIE	DENUMIRE	LOCATIE
1	1	Galeria Nationala	Bucuresti
2	2	Museum of Modern Art	New York
3	3	Louvre Museum	Paris
4	4	Tate Modern	Londra
5	5	Uffizi Gallery	Florenta
6	6	Museum of Modern Art	Bruxelles

```
CREATE TABLE Expozitie (
  id_expozitie INT PRIMARY KEY,
  nume_expozitie VARCHAR(255) NOT NULL,
  data_inceput TIMESTAMP NOT NULL,
  data_final TIMESTAMP NOT NULL,
  pret_bilet INT NOT NULL
);

INSERT INTO Expozitie (id_expozitie, nume_expozitie, data_inceput, data_final,
pret_bilet)
VALUES
    (1, 'Semne si simboluri', TO_TIMESTAMP('2022-06-01 08:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
```

```
   TO_TIMESTAMP('2022-06-30 21:59:00', 'YYYY-MM-DD HH24:MI:SS'), 25);

INSERT INTO Expozitie (id_expozitie,nume_expozitie, data_inceput, data_final,
pret_bilet)
VALUES
    (2, 'One Night Exhibition', TO_TIMESTAMP('2023-07-15 07:00:00', 'YYYY-MM-DD
HH24:MI:SS'),
   TO_TIMESTAMP('2023-07-16 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 40);

INSERT INTO Expozitie (id_expozitie, nume_expozitie, data_inceput, data_final,
pret_bilet)
VALUES
    (3, 'Between Worlds and Traditions', TO_TIMESTAMP('2022-09-01 08:00:00',
'YYYY-MM-DD HH24:MI:SS'),
  TO_TIMESTAMP('2022-09-30 23:59:00', 'YYYY-MM-DD HH24:MI:SS'), 15);

INSERT INTO Expozitie (id_expozitie, nume_expozitie, data_inceput, data_final,
pret_bilet)
VALUES
  (4, 'Gardens', TO_TIMESTAMP('2023-10-15 10:00:00', 'YYYY-MM-DD HH24:MI:SS'),
  TO_TIMESTAMP('2023-11-15 17:00:00', 'YYYY-MM-DD HH24:MI:SS'), 25);

INSERT INTO Expozitie (id_expozitie, nume_expozitie, data_inceput, data_final,
pret_bilet)
VALUES
     (5, 'Retrospectiva', TO_TIMESTAMP('2023-12-01 09:30:00 ', 'YYYY-MM-DD
HH24:MI:SS'),
  TO_TIMESTAMP('2023-12-31 18:00:00', 'YYYY-MM-DD HH24:MI:SS'), 45);
```

	ID_EXPOZITIE	NUME_EXPOZITIE	DATA_INCEPUT	DATA_FINAL	PRET_BILET
1	1	Semne si simboluri	01-JUN-22 08.00.00.000000000 AM	30-JUN-22 09.59.00.000000000 PM	25
2	2	One Night Exhibition	15-JUL-23 07.00.00.000000000 AM	16-JUL-23 12.00.00.000000000 AM	40
3	3	Between Worlds and Traditions	01-SEP-22 08.00.00.000000000 AM	30-SEP-22 11.59.00.000000000 PM	15
4	4	Gardens	15-OCT-23 10.00.00.000000000 AM	15-NOV-23 05.00.00.000000000 PM	25
5	5	Retrospectiva	01-DEC-23 09.30.00.000000000 AM	31-DEC-23 06.00.00.000000000 PM	45

```
CREATE TABLE Angajat (
  id_angajat INT PRIMARY KEY,
  nume VARCHAR(255) NOT NULL,
  pozitie VARCHAR(255) NOT NULL,
  salariu INT NOT NULL,
  id_galerie INT,
  FOREIGN KEY (id_galerie) REFERENCES Galerie(id_galerie)
);


INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (1, 'Dimitrescu', 'Manager', 6000, 1);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
```

```
VALUES
  (2, 'Barbu', 'Ghid', 3500, 2);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (3, 'Gheorghe', 'Director', 8000, 3);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (4, 'Iacob', 'Curator', 4500, 4);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (5, 'Stanciu', 'Ghid', 3500, 5);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (6, 'Simion', 'Manager', 6000, 2);

INSERT INTO Angajat (id_angajat, nume, pozitie, salariu, id_galerie)
VALUES
  (7, 'Palade', 'Ghid', 3500, 3);
```

	ID_ANGAJAT	NUME	POZITIE	SALARIU	ID_GALERIE
1	1	Dimitrescu	Manager	6000	1
2	2	Barbu	Ghid	3500	2
3	3	Gheorghe	Director	8000	3
4	4	Iacob	Curator	4500	4
5	5	Stanciu	Ghid	3500	5
6	6	Simion	Manager	6000	2
7	7	Palade	Ghid	3500	3

```
CREATE TABLE Vizitator (
  id_vizitator INT PRIMARY KEY,
  nume VARCHAR(255) NOT NULL,
  prenume VARCHAR(255) NOT NULL,
  adresa_mail VARCHAR(255)
);

INSERT INTO Vizitator (id_vizitator, nume, prenume, adresa_mail)
VALUES
  (1, 'Diaconescu', 'Cosmin', 'd.cosmin@gmail.com');

INSERT INTO Vizitator (id_vizitator, nume, prenume, adresa_mail)
VALUES
  (2, 'Voinea', 'David', 'david.v@gmail.com');

INSERT INTO Vizitator (id_vizitator, nume, prenume, adresa_mail)
```

```
VALUES
  (3, 'Vlasceanu', 'Luminita', 'v.luminita@gmail.com');

INSERT INTO Vizitator (id_vizitator, nume, prenume, adresa_mail)
VALUES
  (4, 'Paul', 'Marin', 'marin.paul@gmail.com');

INSERT INTO Vizitator (id_vizitator, nume, prenume, adresa_mail)
VALUES
  (5, 'Dragan', 'Andreea', andreea.dragan@gmail.com');
```

	ID_VIZITATOR	NUME	PRENUME	ADRESA_MAIL
1	2	Voinea	David	david.v@gmail.com
2	3	Vlasceanu	Luminita	v.luminita@gmail.com
3	4	Paul	Marin	marin.paul@gmail.com
4	5	Dragan	Andreea	maria.martinez@gmail.com
5	1	Diaconescu	Cosmin	d.cosmin@gmail.com

```
CREATE TABLE Expunere (
      ID INT,
  id_expozitie INT,
  id_artist INT,
      PRIMARY KEY (ID),
  FOREIGN KEY (id_expozitie) REFERENCES Expozitie(id_expozitie),
  FOREIGN KEY (id_artist) REFERENCES Artist(id_artist),
      UNIQUE (id_expozitie, id_artist)
);

INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (1, 1, 1);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (2, 1, 6);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (3, 1, 5);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (4, 2, 4);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (5, 2, 7);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (6, 2, 2);
```

```
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (7, 2, 5);
  INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (8, 3, 3);
INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (9, 4, 3);
  INSERT INTO Expunere (ID, id_expozitie, id_artist)
VALUES
  (10, 4, 1);
```

	ID	ID_EXPOZITIE	ID_ARTIST
1	1	1	1
2	2	1	6
3	3	1	5
4	4	2	4
5	5	2	7
6	6	2	2
7	7	2	5
8	8	3	3
9	9	4	3
10	10	4	1

```
CREATE TABLE Bilet (
  id_bilet INT PRIMARY KEY,
  data_vizita DATE NOT NULL,
  id_vizitator INT,
      id_prezentare INT,
  FOREIGN KEY (id_vizitator) REFERENCES Vizitator(id_vizitator),
  FOREIGN KEY (id_prezentare) REFERENCES Prezentare(id_prezentare),
  UNIQUE (id_prezentare, id_vizitator, data_vizita)
);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (1, TO_DATE('2023-06-15', 'YYYY-MM-DD'), 1, 5);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (2, TO_DATE('2023-07-01', 'YYYY-MM-DD'), 2, 3);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (3, TO_DATE('2023-08-10', 'YYYY-MM-DD'), 3, 1);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
```

```
  (4, TO_DATE('2023-09-20', 'YYYY-MM-DD'), 4, 3);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (5, TO_DATE('2023-10-05', 'YYYY-MM-DD'), 5, 4);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (6, TO_DATE('2022-01-15', 'YYYY-MM-DD'), 2, 5);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (7, TO_DATE('2022-02-23', 'YYYY-MM-DD'), 2, 2);

INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (8, TO_DATE('2022-03-15', 'YYYY-MM-DD'), 3, 1);

  INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (9, TO_DATE('2022-03-15', 'YYYY-MM-DD'), 3, 5);

  INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (10, TO_DATE('2022-03-15', 'YYYY-MM-DD'), 4, 1);
INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (11, TO_DATE('2023-06-15', 'YYYY-MM-DD'), 1, 2);

  INSERT INTO Bilet (id_bilet, data_vizita, id_vizitator, id_prezentare)
VALUES
  (12, TO_DATE('2023-06-15', 'YYYY-MM-DD'), 5, 3);
```

	ID_BILET	DATA_VIZITA	ID_VIZITATOR	ID_PREZENTARE
1	8	15-MAR-22	3	1
2	3	10-AUG-23	3	1
3	10	15-MAR-22	4	1
4	11	15-JUN-23	1	2
5	7	23-FEB-22	2	2
6	2	01-JUL-23	2	3
7	4	20-SEP-23	4	3
8	12	15-JUN-23	5	3
9	5	05-OCT-23	5	4
10	1	15-JUN-23	1	5
11	6	15-JAN-22	2	5
12	9	15-MAR-22	3	5

```sql
CREATE TABLE Prezentare (
  id_prezentare INT,
  id_expozitie INT,
  id_galerie INT,
      PRIMARY KEY (ID),
  FOREIGN KEY (id_expozitie) REFERENCES Expozitie(id_expozitie),
  FOREIGN KEY (id_galerie) REFERENCES Galerie(id_galerie),
      UNIQUE (id_expozitie, id_galerie)
);

INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (0, 1, 2);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (1, 1, 3);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (2, 1, 4);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (3, 2, 4);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (4, 3, 3);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (5, 4, 5);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (6, 5, 3);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (7, 5, 2);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (8, 5, 1);
  INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (9, 5, 5);
  INSERT INTO Prezentare (IDid_prezentare id_expozitie, id_galerie)
VALUES
  (10, 5, 4);
INSERT INTO Prezentare (id_prezentare, id_expozitie, id_galerie)
VALUES
  (11, 4, 1);
```

	ID_PREZEN...	ID_EXPOZITIE	ID_GALERIE
1	0	1	2
2	1	1	3
3	2	1	4
4	3	2	4
5	4	3	3
6	5	4	5
7	6	5	3
8	7	5	2
9	8	5	1
10	9	5	5
11	10	5	4

```
CREATE TABLE Imprumut (
  ID INT,
  id_expozitie INT,
  id_opera INT,
  PRIMARY KEY (ID),
  FOREIGN KEY (id_expozitie) REFERENCES Expozitie(id_expozitie),
  FOREIGN KEY (id_opera) REFERENCES Opera(id_opera),
  UNIQUE (id_expozitie, id_opera)
);

INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (1, 1, 3);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (2, 1, 2);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (3, 1, 8);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (4, 2, 2);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (5, 2, 4);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (6, 2, 7);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (7, 3, 5);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (8, 4, 4);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
```

```
VALUES
  (9, 4, 9);
  INSERT INTO Imprumut (ID, id_expozitie, id_opera)
VALUES
  (10, 5, 5);
```

	ID	ID_EXPOZITIE	ID_OPERA
1	1	1	3
2	2	1	2
3	3	1	8
4	4	2	2
5	5	2	4
6	6	2	7
7	7	3	5
8	8	4	4
9	9	4	9
10	10	5	5

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

• subcereri sincronizate în care intervin cel puțin 3 tabele

• subcereri nesincronizate în clauza FROM

• grupări de date cu subcereri nesincronizate în care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)

• ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

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

- Dintre cele mai vizitate două expoziții, să se aleagă cea care a început cel mai devreme. Nu se va ține cont de ora la care încep expozițiile.

```
SELECT *
FROM (SELECT *                          /*subcerere nesincronizata in clauza FROM */
      FROM (
            SELECT E.NUME_EXPOZITIE, COUNT(V.NUME), E.DATA_INCEPUT
            FROM EXPOZITII E
            JOIN VIZITA VIZ ON VIZ.ID_EXPOZITIE = E.ID_EXPOZITIE
            JOIN VIZITATOR V ON V.ID_VIZITATOR = VIZ.ID_VIZITATOR
            GROUP BY E.NUME_EXPOZITIE, E.DATA_INCEPUT
            ORDER BY COUNT(V.NUME) DESC, ROUND(E.DATA_INCEPUT) ASC   /*functie pe
data calendaristica*/
            )
            WHERE ROWNUM = 1
      );
```

	NUME_EXPOZITIE	COUNT(V.NUME)	DATA_INCEPUT
1	Semne si simboluri	4	01-06-2022 08:00:00

- Afișați numele și naționalitatea artiștilor care au opere expuse la expozițiile afișate de Museum of Modern Art.

```
SELECT CONCAT(CONCAT(A.NUME , ', '), A.NATIONALITATE) ARTIST   /*functie pe siruri de
caractere*/
FROM ARTIST A
WHERE A.ID_ARTIST IN             /*subcereri sincronizate în care intervin tabelele
expunere, participare, galerie*/
         (
          SELECT E.ID_ARTIST
          FROM EXPUNERE E
          WHERE E.ID_EXPOZITIE IN (
                         SELECT P.ID_EXPOZITIE
                         FROM PARTICIPARE P
                         WHERE P.ID_GALERIE = (
                                   SELECT ID_GALERIE
                                   FROM GALERIE
                                   WHERE UPPER(DENUMIRE) = 'MUSEUM OF MODERN ART'
/*functie pe siruri de caractere*/
                                              )
                                   )
            );
```

	ARTIST
1	Leonardo da Vinci, italian
2	Frida Kahlo, mexican
3	Michelangelo, italian

- Obțineți numărul de luni dintre cea mai veche operă din colecție și data curentă, pentru a determina câți ani și luni au trecut de la crearea acelei opere până în prezent.

```
WITH CEAMAIVECHEOPERA AS (
  SELECT
    TO_DATE(MIN(AN_CREATIE), 'YYYY') DATA_CREATIE
  FROM OPERA
)
SELECT
  TRUNC(MONTHS_BETWEEN(SYSDATE, DATA_CREATIE) / 12) ANI,     /*functii pe date
calendaristice*/
  ROUND(MOD(MONTHS_BETWEEN(SYSDATE, DATA_CREATIE), 12)) LUNI
FROM
  CEAMAIVECHEOPERA;
```

	ANI	LUNI
1	524	1

- Să se afișeze titlul și numele artistului operelor cu număr maxim de împrumuturi.

```
SELECT O.TITLU, A.NUME
FROM OPERA O
JOIN ARTIST A ON A.ID_ARTIST = O.ID_ARTIST
LEFT JOIN IMPRUMUT I ON I.ID_OPERA = O.ID_OPERA
GROUP BY O.TITLU, A.NUME     /*functie grup*/
/*filtrare la nivel de grupuri*/
HAVING NVL(COUNT(I.ID_EXPOZITIE), 0) = (SELECT MAXIM
                                        FROM
                                          (SELECT O.TITLU, NVL(COUNT(I.ID_EXPOZITIE), 0)
MAXIM
                                           FROM OPERA O
                                           LEFT JOIN IMPRUMUT I ON I.ID_OPERA = O.ID_OPERA
                                           LEFT JOIN EXPOZITII E ON E.ID_EXPOZITIE =
I.ID_EXPOZITIE
                                           GROUP BY O.TITLU    /*functie grup*/
                                           ORDER BY NVL(COUNT(E.ID_EXPOZITIE), 0) DESC
                                           )
                                        WHERE ROWNUM = 1
                                        );
```

	TITLU	NUME
1	Starry Night	Vincent van Gogh
2	Water Lilies	Claude Monet
3	The Two Fridas	Frida Kahlo
4	The Thinker	Auguste Rodin

- Să se afișeze, pentru fiecare pictură, perioada în care se plasează: perioada anterioară secolului XX, secolul XX, secolul XXI. În dreptul picturilor cărora nu le este cunoscută data creării se va scrie 'Necunoscut'.

```
WITH      /*clauza with*/
    PICTURI AS (
        SELECT ID_OPERA, TITLU, NVL(AN_CREATIE, 0) AS AN_CREATIE_NOTNULL    /*NVL*/
        FROM OPERA
        WHERE UPPER(TIP) = 'PICTURA'   /*functie siruri*/
    ),
    PICTURI_UPDATE AS (
        SELECT ID_OPERA, TITLU, DECODE(AN_CREATIE_NOTNULL, 0, 0, AN_CREATIE_NOTNULL) AS
AN_CREATIE_UPDATE  /*DECODE*/
        FROM PICTURI
    )
SELECT TITLU, AN_CREATIE_UPDATE,
    CASE
        WHEN AN_CREATIE_UPDATE < 1900 AND AN_CREATIE_UPDATE > 0 THEN 'Perioada anterioară
secolului XX'
        WHEN AN_CREATIE_UPDATE >= 1900 AND AN_CREATIE_UPDATE < 2000 THEN 'Secolul XX'
        WHEN AN_CREATIE_UPDATE >= 2000 THEN 'Secolul XXI'
        ELSE 'Necunoscut'
    END PERIOADA
FROM PICTURI_UPDATE
ORDER BY AN_CREATIE_UPDATE;
```

	TITLU	AN_CREATIE_UPDATE	PERIOADA
1	Cina cea de Taina	0	Necunoscut
2	Mona Lisa	1503	Perioada anterioară secolului XX
3	Starry Night	1889	Perioada anterioară secolului XX
4	Water Lilies	1919	Secolul XX
5	Guernica	1937	Secolul XX
6	The Two Fridas	1939	Secolul XX

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

- Toate expozițiile care durează mai mult de 5 zile vor fi amânate cu o zi.

```
UPDATE EXPOZITII
SET DATA_INCEPUT = DATA_INCEPUT + INTERVAL '1' DAY
WHERE ID_EXPOZITIE IN (
    SELECT ID_EXPOZITIE
    FROM EXPOZITII
    WHERE DATA_FINAL - DATA_INCEPUT > INTERVAL '5' DAY
);
```

	ID_EXPOZITIE	NUME_EXPOZITIE	DATA_INCEPUT	DATA_FINAL
1	1	Semne si simboluri	02-06-2022 08:00:00,000000000	30-06-2022 21:59:00
2	2	One Night Exhibition	15-07-2023 07:00:00,000000000	16-07-2023 00:00:00
3	3	Between Worlds and Traditions	02-09-2022 08:00:00,000000000	30-09-2022 23:59:00
4	4	Gardens	16-10-2023 10:00:00,000000000	15-11-2023 17:00:00
5	5	Retrospectiva	02-12-2023 09:30:00,000000000	31-12-2023 18:00:00

- Vizitatorii cu nume care încep cu litera ‘D’ au renunțat la biletele lor, deci se șterg din baza de date.

```
DELETE FROM BILET
WHERE ID_VIZITATOR IN (
    SELECT ID_VIZITATOR
    FROM VIZITATOR
    WHERE VIZITATOR.NUME LIKE 'D%'
    );
```

	ID_BILET	PRET	DATA_VIZITA	ID_VIZITATOR	ID_EXPOZITIE
1	2	15	01-07-2023	2	4
2	3	20	10-08-2023	3	3
3	4	30	20-09-2023	4	2
4	6	13	15-01-2022	2	5
5	7	26	23-02-2022	2	3
6	8	10	15-03-2022	3	1

- Se mărește cu 10% prețul tuturor biletelor ce corespund expoziției "Semne și simboluri".

```
UPDATE BILET
SET PRET = PRET * 1.1
WHERE ID_EXPOZITIE = (
    SELECT ID_EXPOZITIE
    FROM EXPOZITII
    WHERE UPPER(NUME_EXPOZITIE) = 'SEMNE SI SIMBOLURI'
);
```

	ID_BILET	PRET	DATA_VIZITA	ID_VIZITATOR	ID_EXPOZITIE
1	2	15	01-07-2023	2	4
2	3	20	10-08-2023	3	3
3	4	30	20-09-2023	4	2
4	6	13	15-01-2022	2	5
5	7	26	23-02-2022	2	3
6	8	13,31	15-03-2022	3	1

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

Vizualizare: operele cu artiștii și cumpărătorii lor, pentru operele care au cumpărători

```
CREATE VIEW VIZ_OPERE AS
SELECT O.TITLU, A.NUME NUME_ARTIST, C.NUME NUME_COLECTIONAR
FROM OPERA O
JOIN ARTIST A ON A.ID_ARTIST = O.ID_ARTIST
JOIN COLECTIONAR C ON C.ID_OPERA = O.ID_OPERA;
```

LMD permis:

```
DELETE FROM VIZ_OPERE
WHERE NUME_COLECTIONAR = 'Dumitru';
```

LMD nepermis - există coloane NOT NULL fără valoare implicită în tabelul de bază şi care nu au fost selectate în vedere.

```
INSERT INTO VIZ_OPERE ( TITLU, NUME_ARTIST, NUME_COLECTIONAR)
VALUES ('The Water Lily Pond', 'Claude Monet', 'Istrate');
```

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

- Sa se afișeze toate operele, împreună cu expozițiile la care sunt expuse și artiștii care le-au creat. În cazul în care o operă nu este expusă, sau există o expunere fără opere, se va mentiona acest lucru.

```
SELECT NVL(O.TITLU, 'nu are opere expuse') TITLU,
       NVL(E.NUME_EXPOZITIE, 'nu este expusa') EXPOZITIE,
       NVL(A.NUME, 'nu are artisti expusi') ARTIST
FROM OPERA O
FULL OUTER JOIN ARTIST A ON A.ID_ARTIST = O.ID_ARTIST    /*outer join pe 4 tabele*/
FULL OUTER JOIN IMPRUMUT I ON I.ID_OPERA = O.ID_OPERA
FULL OUTER JOIN EXPOZITII E ON E.ID_EXPOZITIE=I.ID_EXPOZITIE
ORDER BY O.TITLU;
```

	TITLU	EXPOZITIE	ARTIST
1	Bird in Space	Gardens	Constantin Brancusi
2	Cina cea de Taina	nu este expusa	Leonardo da Vinci
3	David	nu este expusa	Michelangelo
4	Guernica	Semne si simboluri	Pablo Picasso
5	Mona Lisa	nu este expusa	Leonardo da Vinci
6	Pieta	Semne si simboluri	Michelangelo
7	Starry Night	Semne si simboluri	Vincent van Gogh
8	Starry Night	Gardens	Vincent van Gogh
9	Starry Night	Between Worlds and Traditions	Vincent van Gogh
10	Starry Night	One Night Exhibition	Vincent van Gogh
11	Starry Night	Retrospectiva	Vincent van Gogh
12	The Kiss	nu este expusa	Auguste Rodin
13	The Thinker	One Night Exhibition	Auguste Rodin
14	The Thinker	One Night Exhibition	Auguste Rodin
15	The Two Fridas	Between Worlds and Traditions	Frida Kahlo
16	The Two Fridas	Retrospectiva	Frida Kahlo
17	Water Lilies	One Night Exhibition	Claude Monet
18	Water Lilies	Gardens	Claude Monet

- Să se afișeze opera care a fost expusă în toate expozițiile.

```
SELECT *
FROM Opera o
WHERE NOT EXISTS (
    (SELECT e.id_expozitie FROM Expozitii e)
    EXCEPT
    (SELECT ep.id_expozitie FROM Imprumut ep WHERE ep.id_opera = o.id_opera)
);
```

	ID_OPERA	TITLU	ID_ARTIST	TIP	AN_CREATIE
1	2	Starry Night	2	Pictura	1889

- Să se afișeze cele mai recente 5 opere.

```
SELECT O.TITLU, A.NUME, O.AN_CREATIE
FROM (SELECT *
      FROM OPERA
      WHERE AN_CREATIE IS NOT NULL
      ORDER BY AN_CREATIE DESC) O
JOIN ARTIST A ON O.ID_ARTIST = A.ID_ARTIST
WHERE ROWNUM <= 5;
```

	TITLU	NUME	AN_CREATIE
1	The Two Fridas	Frida Kahlo	1939
2	Guernica	Pablo Picasso	1937
3	Bird in Space	Constantin Brancusi	1923
4	Water Lilies	Claude Monet	1919
5	The Thinker	Auguste Rodin	1902

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

Să se afișeze toți artiștii francezi care au opere ce fac parte din expoziția One Night Exhibition, împreună cu operele acestora, care au fost realizate după anul 1900.

SQL:

```
SELECT DISTINCT A.NUME, O.TITLU, O.AN_CREATIE
FROM OPERA O
JOIN ARTIST A ON A.ID_ARTIST=O.ID_ARTIST
JOIN IMPRUMUT I ON O.ID_OPERA=I.ID_OPERA
JOIN EXPOZITII E ON E.ID_EXPOZITIE=I.ID_EXPOZITIE
WHERE O.AN_CREATIE IN
        ( SELECT O2.AN_CREATIE
          FROM OPERA O2
          WHERE O2.AN_CREATIE = O.AN_CREATIE AND O2.AN_CREATIE > 1900
        )
        AND A.NATIONALITATE IN
        (
          SELECT A2.NATIONALITATE
          FROM ARTIST A2
          WHERE A2.NATIONALITATE = A.NATIONALITATE AND UPPER(A2.NATIONALITATE)='FRANCEZ'
        )
        AND E.NUME_EXPOZITIE IN
        (
          SELECT E2.NUME_EXPOZITIE
```

```
          FROM EXPOZITII E2
      WHERE E2.NUME_EXPOZITIE = E.NUME_EXPOZITIE AND UPPER(E2.NUME_EXPOZITIE)='ONE NIGHT
EXHIBITION'
        );
```

Expresie algebrică:

R1 = JOIN (opera, artist)

R2 = JOIN (imprumut, R1)

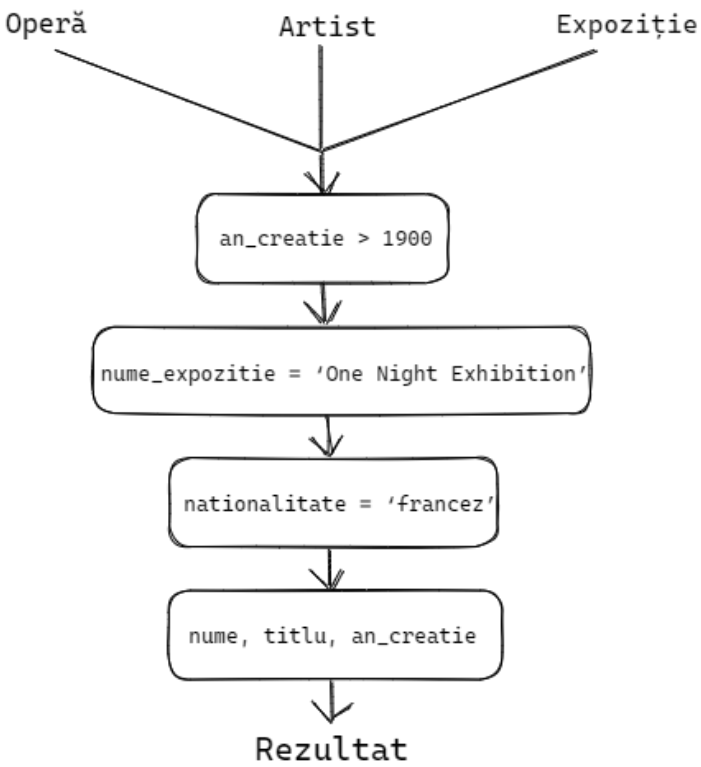
R3 = JOIN (expoziții, R2)

R4 = SELECT (R3, an_creatie > 1900)

R5 = SELECT (R4, nume_expozitie = ‘One Night Exhibition’)

R6 = SELECT (R5, nationalitate = ‘francez’)

REZULTAT = R7 = PROJECT (R6, nume, titlu, an_creatie)



Etapa intermediară:

SQL:

```
SELECT DISTINCT A.NUME, O.TITLU, O.AN_CREATIE
FROM OPERA O
JOIN ARTIST A ON A.ID_ARTIST=O.ID_ARTIST
JOIN IMPRUMUT I ON O.ID_OPERA=I.ID_OPERA
JOIN EXPOZITII E ON E.ID_EXPOZITIE=I.ID_EXPOZITIE
WHERE O.AN_CREATIE > 1900
      AND UPPER(A.NATIONALITATE)='FRANCEZ'
      AND UPPER(E.NUME_EXPOZITIE)='ONE NIGHT EXHIBITION';
```

Expresie algebrică:

R1 = JOIN (opera, artist)

R2 = JOIN (imprumut, R1)

R3 = JOIN (expoziții, R2)

R4 = SELECT (R3, an_creatie>1900 and nume_expozitie=‘One Night Exhibition’ and nationalitate=‘francez’)

REZULTAT = R5 = PROJECT (R4, nume, titlu, an_creatie)

Etapa finală:

SQL:

```
SELECT DISTINCT A.NUME, O.TITLU, O.AN_CREATIE
FROM (
     SELECT *
     FROM OPERA
     WHERE AN_CREATIE > 1900) O
JOIN (
     SELECT *
     FROM ARTIST
     WHERE UPPER(NATIONALITATE)='FRANCEZ') A ON A.ID_ARTIST=O.ID_ARTIST
JOIN IMPRUMUT I ON O.ID_OPERA=I.ID_OPERA
JOIN (
     SELECT *
     FROM EXPOZITII
  WHERE UPPER(NUME_EXPOZITIE)='ONE NIGHT EXHIBITION') E ON E.ID_EXPOZITIE=I.ID_EXPOZITIE;
```

Expresie algebrică:

R1 = SELECT (opera, an_creatie>1900)

R2 = JOIN (R1, artist, nationalitate=‘francez’)

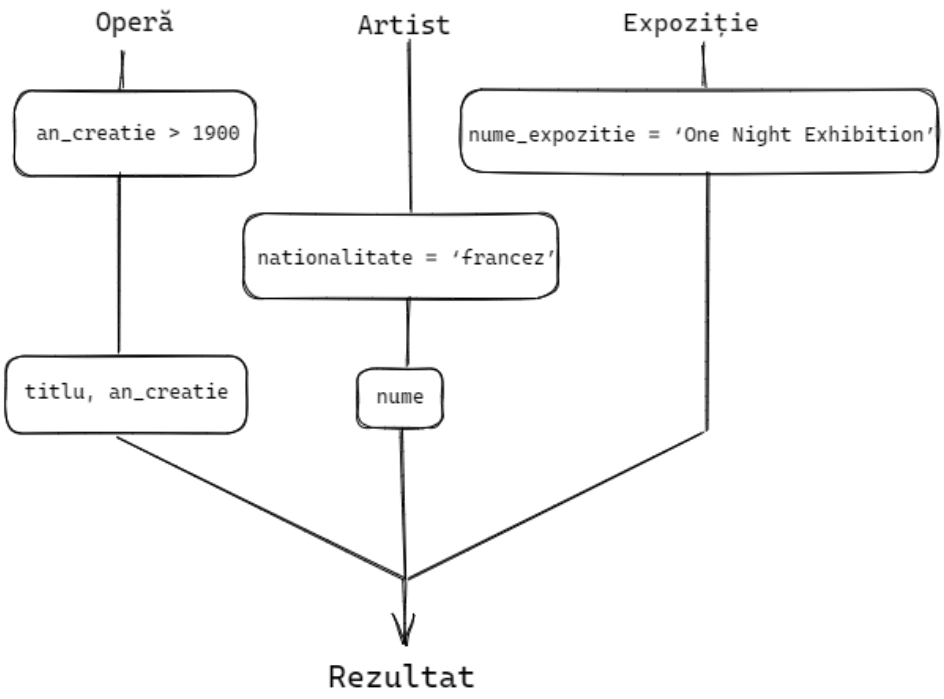
R3 = JOIN (R2, imprumut)

R4 = JOIN (R3, expoziții, nume_expozitie=‘One Night Exhibition’)

R5 = PROJECT (R1, titlu, an_creatie)

R6 = PROJECT (R2, nume)

REZULTAT = R7 = JOIN (R5, R6)



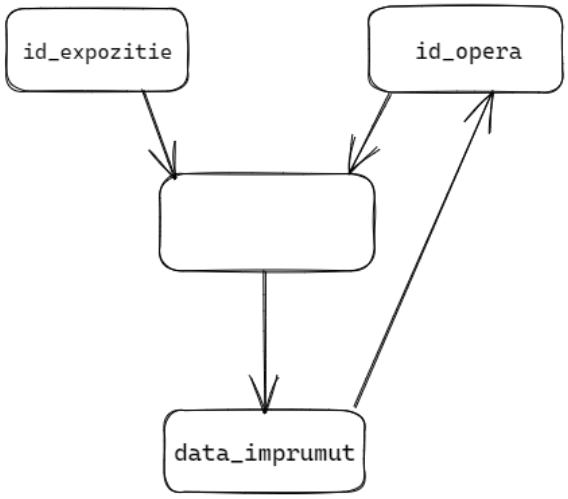
17. a. Realizarea normalizării BCNF[, FN4, FN5]

• Forma normală Boyce-Codd (BCNF)

Forma normală Boyce-Codd se bazează pe dependenţele funcţionale care iau în considerare toate cheile candidat dintr-o relaţie.

Pentru relațiile cu o singura cheie candidat, formele FN3 si BCNF sunt echivalente (Galerie, Artist etc.)

Să luăm relația ÎMPRUMUT (id_expozitie#,id_opera#, data_imprumut).



Regula Casey Delobel pentru ÎMPRUMUT (id_expozitie#,id_opera#, data_imprumut) având faptul că data_imprumut->id_opera:

- ÎMPRUMUT_1 (id_expozitie#, data_imprumut)
- ÎMPRUMUT_2 (data_imprumut, id_opera)

•Forma nomală 4 (FN4)

FN4 elimină redundanțele datorate relațiilor m:n, adică datorate dependenței multiple. O relație este în a patra formă nomală dacă și numai dacă este în BCNF și nu conține relații m:n independente.

Iau relația GALERIE(id_galerie#, denumire, locatie) și presupun că o galerie poate avea mai multe denumiri și mai multe locații.

- id_galerie# => denumire;
- id_galerie# => locatie;

Relația GALERIE este in BCNF. Pentru a aduce relația în FN4 o vom descompune prin proiecție în două relații:

GALERIE1(id_galerie#, denumire)

GALERIE2(id_galerie#, locatie)

=> GALERIE = JOIN(GALERIE1, GALERIE2)

• Forma normală 5 (FN5)

O relaţie R este în FN5 (numită şi forma normală proiecţie-uniune) dacă şi numai dacă orice dependenţă de uniune a lui R este o consecinţă a unei chei candidat a lui R.

Orice relaţie care este în FN5 este şi în FN4, deoarece fiecare dependenţă multivaloare poate fi privită ca un caz particular de dependenţă de uniune. Orice relaţie poate fi descompusă fără pierderi la uniune într-o mulţime de relaţii care sunt în FN5.

Pentru a preciza dacă o relaţie este în FN5, este suficient să cunoaştem cheile candidate şi toate dependenţele de uniune din R.

Aducerea în FN5 presupune eliminarea join dependențelor.

Să luăm relația GALERIE și să presupunem că am avea o join dependență în aceasta – iau mulțimea (fax, adresa, cod_postal) presupunem că există multiple dependențe între fiecare dintre perechile din mulțimi (fax,adresa), (adresa,cod_postal) și (fax, cod_postal) – se pierde faptul că în acest caz cod_postal se duce in fax - este un exemplu pur ipotetic.

b. Aplicarea denormalizării, justificând necesitatea acesteia.

Denormalizarea este un proces în care se combină două sau mai multe tabele normalizate într-o singură tabelă, pentru a îmbunătăți performanța și eficiența operațiilor de interogare asupra bazei de date. Acest proces implică reintroducerea redundanței controlate în baza de date.

În ciuda evidentelor dezavantaje, cum ar fi creșterea spațiului de stocare, denormalizarea are și avantaje:

•**Performanță sporită**: Prin combinarea tabelelor normalizate într-o singură tabelă, se poate reduce numărul de operații de îmbinare (join-uri) necesare pentru a obține datele. Acest lucru poate duce la o îmbunătățire semnificativă a performanței în cazul interogărilor complexe și a volumelor mari de date.

•**Simplificarea interogărilor**: Denormalizarea poate face interogările mai simple și mai ușor de scris, deoarece datele necesare sunt disponibile într-un singur loc. Nu este nevoie de join-uri complexe pentru a obține date din mai multe tabele.

•**Reducerea complexității aplicației**: Prin denormalizare, se poate reduce complexitatea aplicației, deoarece nu mai este nevoie de logica complexă de îmbinare a datelor din tabele separate.

•**Optimizarea operațiilor de scriere**: În anumite situații, denormalizarea poate îmbunătăți performanța operațiunilor de scriere, cum ar fi inserări, actualizări și ștergeri, deoarece nu este necesară actualizarea mai multor tabele.

Aici, abordarea denormalizării subliniază conceptul că, plasând toate datele într-un singur loc, ar putea elimina necesitatea căutării acelor fișiere multiple pentru a colecta aceste date. În cadrul bazei mele de date, luând spre exemplu relația GALERIE și descompunerea ei de la FN4, este inutil și mult mai costisitor din punct de vedere al timpului de executare să parcurgem datele și din GALERIE1 și GALERIE2.

18. Tranzacții: ilustrarea consistency levels in Oracle cu tranzacții care operează asupra modelului ales.

Nivelele de izolare în SQL sunt mecanisme care controlează modul în care tranzacțiile accesează și modifică datele simultan într-o bază de date multi-utilizator. Aceste nivele asigură consistența și integritatea datelor în timpul tranzacțiilor și previn anomalii precum dirty reads, non-repeatable reads și phantoms. Aici este o scurtă prezentare a nivelelor de izolare din Oracle (acesta nu are *REPEATABLE READ* și *READ UNCOMMITED*) și a anomaliilor asociate:

- *READ COMMITTED*: Acest nivel permite o vizibilitate a modificărilor numai după ce acestea au fost confirmate (comise). Datele necomise de alte tranzacții nu sunt vizibile. Cu toate acestea, o tranzacție poate experimenta non-repeatable reads, în care aceeași interogare returnează rezultate diferite în timpul execuției, din cauza modificărilor comise de alte tranzacții între citiri.

- *SERIALIZABLE*: La acest nivel, tranzacțiile sunt executate într-o ordine ‘serială’, ceea ce înseamnă că tranzacțiile nu pot interacționa în mod concurent. Aceasta asigură cea mai mare izolare și evită non-repeatable reads și phantom. Anomalia phantom apare atunci când o tranzacție observă înregistrări noi care apar între citiri, ca urmare a inserării sau ștergerii efectuate de alte tranzacții.

Exemple de tranzacții:

```
create table EXP as select * from EXPOZITII;
create table OP as select * from OPERA;

SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
UPDATE OP
SET ID_OPERA = 25 WHERE ID_OPERA = 2;
COMMIT;

SELECT *    /*id = 25*/
FROM OP;
```

```
ROLLBACK;

SELECT *    /*id = 2*/
FROM OP;
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

DELETE FROM EXP
WHERE NUME_EXPOZITIE = 'One Night Exhibition';
SAVEPOINT salvare1;

SELECT *
FROM EXP;   /*nu mai apare 'One Night Exhibition'*/

DELETE FROM EXP
WHERE NUME_EXPOZITIE = 'Semne si simboluri';
SAVEPOINT salvare2;

SELECT *
FROM EXP;   /*nu mai apar 'One Night Exhibition' si 'Semne si simboluri'*/

ROLLBACK TO SAVEPOINT salvare1;

SELECT *
FROM EXP;   /*reapare doar 'Semne si simboluri'*/
```

19. Optimizarea a două cereri utilizând indexare.

- Numarul de opere al artistului cu id-ul 7

```
CREATE INDEX IND_ARTIST
ON OPERA (ID_ARTIST);

SELECT A.NUME, COUNT(*) NR_OPERE
FROM OPERA O
JOIN ARTIST A ON O.ID_ARTIST = A.ID_ARTIST
WHERE O.ID_ARTIST=7
GROUP BY A.NUME;
```

- Biletele cumparate de vizitatorul 2

```
CREATE INDEX IND_VIZITATOR
ON VIZITATOR (ID_VIZITATOR);

SELECT *
FROM BILET B
JOIN VIZITATOR V ON V.ID_VIZITATOR = B.ID_VIZITATOR
WHERE B.ID_VIZITATOR = 2;
```