

Grupa 143

# Gestiunea unui Cinematograf

## Ioana Potlog

Ioana Potlog

## 1. Descrierea modelului real și regulile de funcționare:

Tema pe care am ales-o pentru proiect este Cinematograful „The Retro Cinema”, un cinematograful unde se pot viziona filme din perioada anilor '80.

Prin intermediul bazei de date, cinematograful stochează, organizează și gestionează eficient date legate de filme, săli de cinema, comenzi, clienți, angajați și multe altele, astfel oferind o multitudine de avantaje. Printre acestea se regăsesc următoarele: planificarea eficientă a programului de proiecții a filmelor, gestionarea vânzărilor și stocului de bilete, produse alimentare și băuturi, analiza performanței cinematografului, ușurința accesării informațiilor etc.

Reguli de funcționare:

- Un film poate fi difuzat în mai multe proiecții la cinema.
- O proiecție poate prezenta un singur film.
- O proiecție se desfășoară într-o anumită sală de cinema.
- O sală de cinema poate difuza mai multe proiecții.
- O sală de cinema este alocată unui cinema anume.
- Un cinema poate avea mai multe săli de cinema disponibile.
- O proiecție poate avea mai multe bilete vândute.
- Un bilet aparține unei singure proiecții.
- O comandă poate conține unul sau mai multe bilete și unul sau mai multe produse achiziționate de un client.
- Un bilet poate fi asociat cu o anumită comandă.
- Un produs poate fi asociat mai multor comenzi.
- Un angajat poate prelua mai multe comenzi.
- O comandă poate fi preluată de un singur angajat.
- O comandă aparține unui client.
- Un client poate să plaseze una sau multe comenzi.

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

- Unicitatea cheilor primare: Fiecare entitate din modelul de date are o cheie primară unică care identifică în mod unic înregistrările din acea entitate.
- Constrângeri de tipul NOT NULL: Acestea specifică că anumite câmpuri nu pot fi lăsate necompletate sau cu valori nule. De exemplu, attributele titlu\_film sau nume\_cinema nu pot fi lăsate necompletate.
- Constrângeri de tipul NULL: Acestea permit existența valorilor nule într-un atribut, ceea ce înseamnă că atributul poate fi lăsat necompletat sau să nu conțină o valoare validă. Spre exemplu, atributul descriere poate fi completat sau necompletat.

- Constrângeri de tipul UNIQUE: Acestea asigură că anumite câmpuri trebuie să fie unice în cadrul unei entități. De exemplu, adresa de e-mail a unui client trebuie să fie unică.
- Constrângeri de tipul DEFAULT: Acestea permit specificarea unei valori implicite pentru un câmp în cazul în care nu este furnizată o valoare explicită. De exemplu, se poate stabili că metoda de plată implicită pentru o comandă este "cash".
- Constrângeri de tipul CHECK: Acestea permit specificarea unei condiții logice care trebuie să fie satisfăcută pentru a se insera sau actualiza o înregistrare în baza de date. Spre exemplu, este impus ca metoda de plată să fie sub formă de "cash" sau "card".

### 3. Descrierea entităților, incluzând precizarea cheii primare:

- 3.1 Entitatea CINEMA – stochează informații despre cinematografe. Cheia primară este id\_cinema.
- 3.2 Entitatea SALA – stochează informații despre sălile de cinema disponibile. Cheia primară este id\_sala.
- 3.3 Entitatea FILM – stochează informații despre filmele care rulează la cinematograf. Cheia primară este id\_film.
- 3.4 Entitatea PROIECȚIE – stochează informații despre proiecțiile unui film. Cheia primară este id\_proiecție.
- 3.5 Entitatea BILET – stochează informații despre biletele vândute pentru proiecțiile filmelor. Cheia primară este id\_bilet.
- 3.6 Entitatea PRODUS – stochează informații despre produsele alimentare și băuturi. Cheia primară este id\_produs.
- 3.7 Entitatea CLIENT – stochează informații despre clienții care au cumpărat bilete și produse alimentare și băuturi. Cheia primară este id\_client.
- 3.8 Entitatea ANGAJAT – stochează informații despre angajații cinematografului. Cheia primară este id\_angajat.
- 3.9 Entitatea COMANDĂ – stochează informații despre comenzile plasate pentru produse. Cheia primară este id\_comandă.

### 4. Descrierea relațiilor, incluzând precizarea cardinalității acestora:

- 4.1. Relația FILM - PROIECȚIE este de tipul one-to-many.

Un film poate fi difuzat în mai multe proiecții (M), dar o proiecție poate prezenta un singur film (1). Această relație este stabilită prin intermediul atributului id\_film din entitatea FILM și atributul id\_film din entitatea PROIECȚIE.

- 4.2. Relația PROIECȚIE - SALA este de tipul many-to-one.

O proiecție se desfășoară într-o anumită sală de cinema (1), iar o sală de cinema poate găzdui mai multe proiecții (M). Această relație este stabilită prin intermediul atributului `id_sala` din entitatea PROIECȚIE și atributul `id_sala` din entitatea SALA.

4.3. Relația CINEMA - SALA este de tipul one-to-many.

Un cinema poate avea mai multe săli de cinema disponibile (M), dar o sală de cinema este alocată unui singur cinema (1). Această relație este stabilită prin intermediul atributului `id_cinema` din entitatea CINEMA și atributul `id_sala` din entitatea SALA.

4.4. Relația PROIECȚIE - BILET este de tipul one-to-many.

O proiecție poate avea mai multe bilete vândute (M), dar un bilet aparține unei singure proiecții (1). Această relație este stabilită prin intermediul atributului `id_proiecție` din entitatea BILET și atributul `id_proiecție` din entitatea PROIECȚIE.

4.5. Relația COMANDĂ - BILET este de tipul one-to-many.

O comandă poate conține unul sau mai multe bilete (M), iar un bilet poate fi asociat cu o anumită comandă (1). Această relație este stabilită prin intermediul atributului `id_comandă` din entitatea BILET și atributul `id_comandă` din entitatea COMANDĂ.

4.6. Relația COMANDĂ - PRODUS este de tipul many\_to\_many.

O comandă poate conține mai multe produse (M), iar un produs poate aparține mai multor comenzi (M). Pentru a conecta cele doua entități se folosește o nouă entitate intermediară numită DETALIU\_COMANDĂ.

4.7. Relația ANGAJAT - COMANDĂ este de tipul many-to-one.

Un angajat poate fi asociat cu mai multe comenzi (M), iar o comandă poate fi preluată de un singur angajat (1). Această relație este stabilită prin intermediul atributului `id_angajat` din entitatea COMANDĂ și atributul `id_angajat` din entitatea ANGAJAT.

4.8. Relația COMANDĂ - CLIENT este de tipul many-to-one.

O comandă aparține unui client (1), iar un client poate să plaseze una sau multe comenzi (M). Această relație este stabilită prin intermediul atributului `id_client` din entitatea COMANDĂ și atributul `id_client` din entitatea CLIENT.

## 5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor:

### 4.1 Entitatea CINEMA:

| ATRIBUT     | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|-------------|----------|------------|-----------------------------|---------------------------|
| id_cinema   | number   | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| nume_cinema | varchar2 | N/A        | N/A                         | NOT NULL                  |
| adresa      | varchar2 | N/A        | N/A                         | NOT NULL                  |
| oraș        | varchar2 | N/A        | N/A                         | NOT NULL                  |

### 4.2 Entitatea SALA:

| ATRIBUT    | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|------------|----------|------------|-----------------------------|---------------------------|
| id_sală    | number   | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| id_cinema  | number   | N/A        | N/A                         | ( FK ) NOT NULL           |
| număr      | varchar2 | N/A        | N/A                         | NOT NULL                  |
| capacitate | number   | N/A        | N/A                         | NOT NULL                  |

### 4.3 Entitatea FILM:

| ATRIBUT    | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|------------|----------|------------|-----------------------------|---------------------------|
| id_film    | number   | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| titlu_film | varchar2 | N/A        | N/A                         | NOT NULL                  |
| durată     | number   | N/A        | N/A                         | NOT NULL                  |
| gen        | varchar2 | N/A        | N/A                         | NOT NULL                  |
| descriere  | varchar2 | N/A        | N/A                         | NULL                      |

### 4.4 Entitatea PROIECȚIE:

| ATRIBUT      | TIP    | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|--------------|--------|------------|-----------------------------|---------------------------|
| id_proiecție | number | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| id_film      | number | N/A        | N/A                         | ( FK ) NOT NULL           |
| id_sală      | number | N/A        | N/A                         | ( FK ) NOT NULL           |
| oră          | time   | N/A        | N/A                         | NOT NULL                  |

### 4.5 Entitatea BILET:

| ATRIBUT  | TIP    | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|----------|--------|------------|-----------------------------|---------------------------|
| id_bilet | number | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |

|              |         |     |     |                 |
|--------------|---------|-----|-----|-----------------|
| id_proiectie | number  | N/A | N/A | ( FK ) NOT NULL |
| id_comandă   | number  | N/A | N/A | ( FK ) NOT NULL |
| rând         | varchar | N/A | N/A | NOT NULL        |
| loc          | number  | N/A | N/A | NOT NULL        |

#### 4.6 Entitatea PRODUS:

| ATRIBUT     | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|-------------|----------|------------|-----------------------------|---------------------------|
| id_produs   | number   | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| nume_produs | varchar2 | N/A        | N/A                         | NOT NULL                  |
| preț        | number   | N/A        | N/A                         | NOT NULL                  |
| stoc        | number   | N/A        | N/A                         | NOT NULL                  |

#### 4.7 Entitatea CLIENT:

| ATRIBUT        | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|----------------|----------|------------|-----------------------------|---------------------------|
| id_client      | number   | N/A        | N/A                         | ( PK ) NOT NULL<br>UNIQUE |
| nume_client    | varchar2 | N/A        | N/A                         | NOT NULL                  |
| prenume_client | number   | N/A        | N/A                         | NOT NULL                  |
| telefon        | varchar2 | 10         | N/A                         | NOT NULL UNIQUE           |
| email          | varchar2 | N/A        | N/A                         | NOT NULL UNIQUE           |

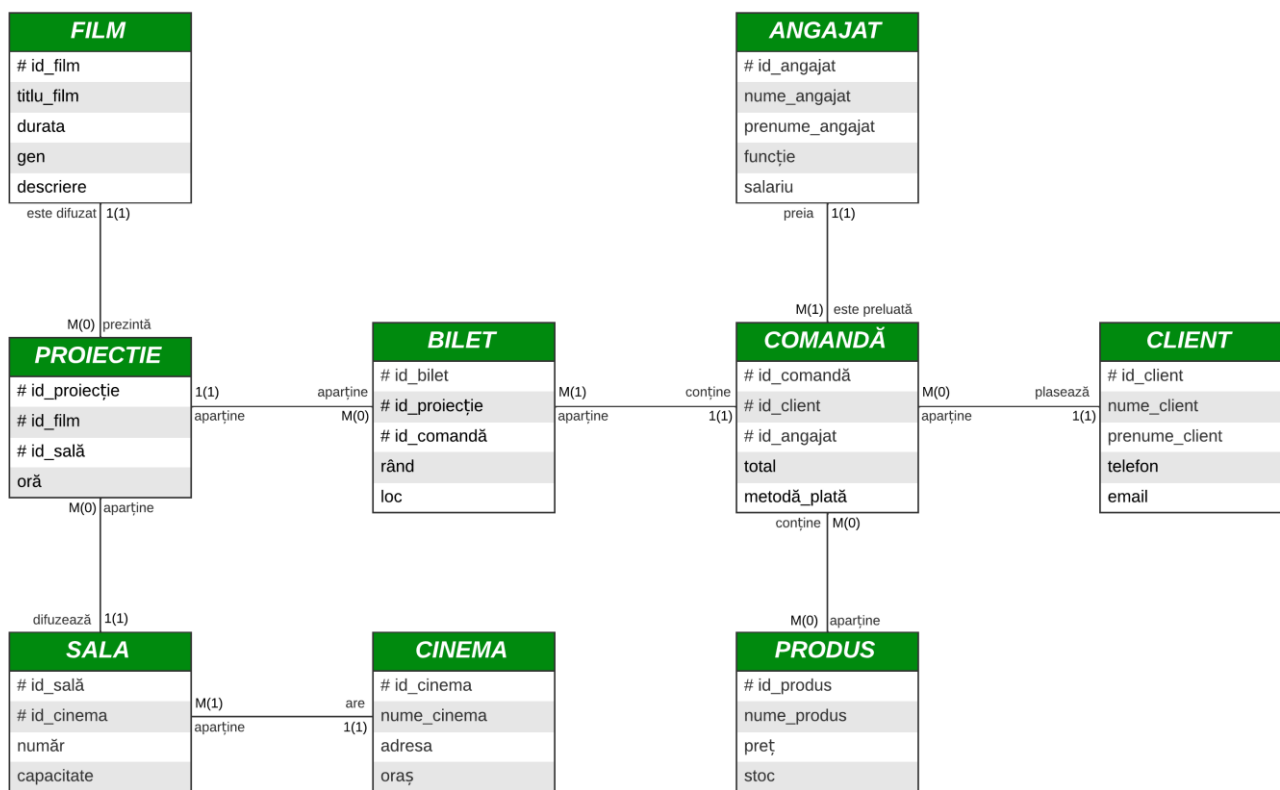
#### 4.8 Entitatea ANGAJAT:

| ATRIBUT         | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE | CONSTRÂNGERI              |
|-----------------|----------|------------|-----------------------------|---------------------------|
| id_angajat      | number   | N/A        |                             | ( PK ) NOT NULL<br>UNIQUE |
| nume_angajat    | varchar2 | N/A        | N/A                         | NOT NULL                  |
| prenume_angajat | varchar2 | N/A        | N/A                         | NOT NULL                  |
| funcție         | varchar2 | N/A        | N/A                         | NOT NULL                  |
| salariu         | number   | N/A        | N/A                         | NOT NULL                  |

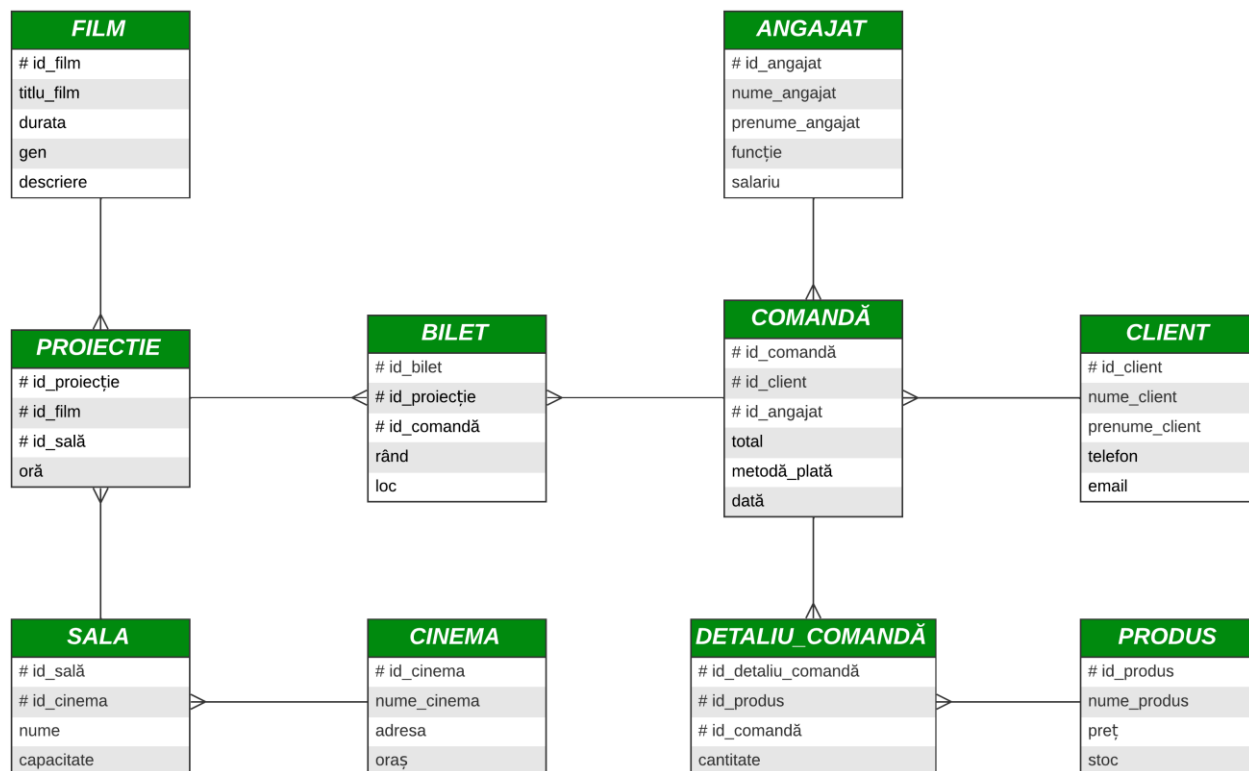
#### 4.9 Entitatea COMANDĂ:

| ATRIBUT      | TIP      | DIMENSIUNE | VALORI POSIBILE / IMPLICITE        | CONSTRÂNGERI              |
|--------------|----------|------------|------------------------------------|---------------------------|
| id_comandă   | number   | N/A        | N/A                                | ( PK ) NOT NULL<br>UNIQUE |
| id_client    | number   | N/A        | N/A                                | ( FK ) NOT NULL           |
| id_angajat   | number   | N/A        | N/A                                | ( FK ) NOT NULL           |
| total        | number   | N/A        | DEFAULT: 0                         | NOT NULL                  |
| metoda_plată | varchar2 | N/A        | CHECK: cash, card<br>DEFAULT: cash | NOT NULL                  |
| dată         | date     | N/A        | N/A                                | NOT NULL                  |

## 6. Diagrama entitate – relație:



## 7. Diagrama conceptuală corespunzătoare diagramei entitate – relație:



## 8. Schemele relaționale corespunzătoare diagramei conceptuale:

- CINEMA (id\_cinema#, nume\_cinema, adresă, oraș)
- SALA (id\_sală#, id\_cinema#(FK), număr, capacitate)
- FILM (id\_film#, titlu\_film, durată, gen, descriere)
- PROIECTIE (id\_proiecție#, id\_film#(FK), id\_sală(FK), oră)
- BILET (id\_bilet#, id\_proiecție#(FK), id\_comandă#(FK), rând, loc)
- COMANDĂ (id\_comandă#, id\_client#(FK), id\_angajat#(FK), total, metodă\_plată)
- PRODUS (id\_produs#, nume\_produs, preț, stoc)
- DETALIU\_COMANDĂ (id\_detaliu\_comandă#, id\_produs#(FK), id\_comandă#(FK), cantitate)
- CLIENT (id\_client#, nume\_client, prenume\_client, telefon, email)
- ANGAJAT (id\_angajat#, nume\_angajat, prenume\_angajat, funcție, salariu)

## 9. Normalizarea:

### ➤ Forma normală 1 (FN1)

O relație se află în prima formă normală dacă și numai dacă fiecare înregistrare este definită astfel încât să fie identificată în mod unic prin intermediul unei chei primare. FN1 exclude existența grupurilor repetitive cerând ca fiecare atribut să cuprindă numai o valoare singulară.

Selectez entitățile CINEMA și SALA.

### 1) Mai multe valori semnificative în același atribut:

Interogările pentru a selecta înregistrări pe baza componentei atributelor conținând mai multe valori semnificative sunt foarte dificile.

| CINEMA | SALĂ    |
|--------|---------|
| Studio | 1, 2, 3 |
| Patria | 1, 2, 3 |



| CINEMA | SALĂ |
|--------|------|
| Studio | 1    |
| Studio | 2    |
| Studio | 3    |
| Patria | 1    |
| Patria | 2    |



|        |   |
|--------|---|
| Patria | 3 |
|--------|---|



| id | CINEMA | SALĂ |
|----|--------|------|
| 1  | Studio | 1    |
| 2  | Studio | 2    |
| 3  | Studio | 3    |
| 4  | Patria | 1    |
| 5  | Patria | 2    |
| 6  | Patria | 3    |

2) Mai multe coloane reprezintă același tip de date:

| CINEMA | SALĂ | SALĂ(1) | SALĂ(2) |
|--------|------|---------|---------|
| Studio | 1    | 2       | 3       |
| Patria | 1    | 2       | 3       |



| CINEMA | SALĂ |
|--------|------|
| Studio | 1    |
| Studio | 2    |
| Studio | 3    |
| Patria | 1    |
| Patria | 2    |
| Patria | 3    |



| id | CINEMA | SALĂ |
|----|--------|------|
| 1  | Studio | 1    |
| 2  | Studio | 2    |
| 3  | Studio | 3    |

|   |        |   |
|---|--------|---|
| 4 | Patria | 1 |
| 5 | Patria | 2 |
| 6 | Patria | 3 |

Așadar, în exemplele de mai sus, am introdus o coloană adițională de tip numeric pentru a 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 toate atributele care nu sunt cheie sunt dependente direct de cheia primară în întregime, nu doar de o parte a cheii primare.

Selectez tabela **DETALII\_COMANDĂ** (înainte de FN2):

| id_detaliu_comandă | id_produs | id_comandă | nume_produs | cantitate |
|--------------------|-----------|------------|-------------|-----------|
| 1                  | 1         | 1          | popcorn     | 15        |
| 2                  | 1         | 2          | popcorn     | 15        |
| 3                  | 2         | 3          | nachos      | 20        |

Relația este în FN1, deoarece avem un identificator unic pentru toate intrările din tabel. Observăm că atributul `nume_produs` nu este cheie primară și nici nu depinde direct de aceasta (`id_detaliu_comandă`), deoarece este dependent de cheia `id_produs`. Astfel, relația nu se află în FN2 și, aplicându-se regula Casey Delobel, este necesară eliminarea atributului `nume_produs` din tabel.

| id_detaliu_comandă | id_produs | id_comandă | cantitate |
|--------------------|-----------|------------|-----------|
| 1                  | 1         | 1          | 15        |
| 2                  | 1         | 2          | 15        |
| 3                  | 2         | 3          | 20        |

### ➤ Forma normală 3 (FN 3)

O relație se află în a treia formă normală dacă și numai dacă această relație este deja în FN2 și toate atributele non-chei depind direct de cheia primară.

Selectez entitatea **PROIECȚIE** (înainte de FN3):

| id_proiecție | titlu | durată | gen | oră |
|--------------|-------|--------|-----|-----|
|--------------|-------|--------|-----|-----|

|   |                    |     |         |       |
|---|--------------------|-----|---------|-------|
| 1 | Dirty Dancing      | 100 | Romance | 18:30 |
| 2 | GhostBusters       | 107 | Comedy  | 20:00 |
| 3 | Dead Poets Society | 128 | Drama   | 14:00 |

Obsevăm că atributele titlu, durată și gen depind de cheia primară id\_proiecție.

Pentru a aduce relația în FN3, trebuie să separăm atributele despre film din PROIECȚIE, apărând astfel entitatea FILM. Așadar, în entitatea PROIECȚIE înlocuiesc aceste atribute cu cheia străină **id\_film** pentru a determina mai ușor la ce oră are loc proiecția unui film.

Entitatea **PROIECȚIE** modificată:

| id_proiecție | id_film | oră   |
|--------------|---------|-------|
| 1            | 1       | 18:30 |
| 2            | 2       | 20:00 |
| 3            | 3       | 14:00 |

Noua entitate **FILM**:

| id_film | titlu              | durată | gen     |
|---------|--------------------|--------|---------|
| 1       | Dirty Dancing      | 100    | Romance |
| 2       | GhostBusters       | 107    | Comedy  |
| 3       | Dead Poets Society | 128    | Drama   |

## 10. Crearea unei secvențe:

```
CREATE SEQUENCE cinema_seq
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

## 11. Crearea tabelelor în SQL și inserarea de date în acestea:

### TABELA CINEMA:

| ID_CINEMA | NUME_CINEMA | ADRESA                 | ORAS        |
|-----------|-------------|------------------------|-------------|
| 1         | Retroplex   | Str. Victoriei nr. 15  | Bucuresti   |
| 2         | Starlight   | Bd. Unirii nr. 20      | Bucuresti   |
| 3         | Pixelplex   | Str. Palat nr. 5       | Iasi        |
| 4         | Cinematica  | Str. Republicii nr. 10 | Brasov      |
| 5         | Flashback   | Piata Unirii nr. 2     | Cluj-Napoca |

CREATE TABLE CINEMA

```
( id_cinema NUMBER(5) CONSTRAINT PKEY_CINEMA PRIMARY KEY,  
  nume_cinema VARCHAR(20) CONSTRAINT nume_cinema NOT NULL UNIQUE,  
  adresa VARCHAR(50) CONSTRAINT adresa_cinema NOT NULL,  
  oras VARCHAR(20) CONSTRAINT oras_cinema NOT NULL  
);
```

INSERT INTO CINEMA

VALUES (cinema\_seq.nextval, 'Retroplex', 'Str. Victoriei nr. 15', 'Bucuresti');

INSERT INTO CINEMA

VALUES (cinema\_seq.nextval, 'Starlight', 'Bd. Unirii nr. 20', 'Bucuresti');

INSERT INTO CINEMA

VALUES (cinema\_seq.nextval, 'Pixelplex', 'Str. Palat nr. 5', 'Iasi');

INSERT INTO CINEMA

VALUES (cinema\_seq.nextval, 'Cinematica', 'Str. Republicii nr. 10', 'Brasov');

INSERT INTO CINEMA

VALUES (cinema\_seq.nextval, 'Flashback', 'Piata Unirii nr. 2', 'Cluj-Napoca');

## TABELA SALA:

| ID_SALA | ID_CINEMA | NUME     | CAPACITATE |
|---------|-----------|----------|------------|
| 6       | 1         | Ultra    | 40         |
| 7       | 1         | Galaxy   | 30         |
| 8       | 2         | Epika    | 25         |
| 9       | 2         | Infinity | 50         |
| 10      | 3         | Astral   | 25         |
| 11      | 4         | Orion    | 15         |
| 12      | 5         | Mega     | 30         |

CREATE TABLE SALA

```
( id_sala NUMBER(5) CONSTRAINT PKEY_SALA PRIMARY KEY,  
  id_cinema NUMBER(5), CONSTRAINT fk_sala FOREIGN KEY(id_cinema) REFERENCES  
  CINEMA(id_cinema),  
  nume VARCHAR(50) CONSTRAINT numar_sala NOT NULL,
```

capacitate NUMBER(10) CONSTRAINT capacitate\_sala NOT NULL

);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Retroplex'), 'Ultra', 40);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Retroplex'), 'Galaxy', 30);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Starlight'), 'Epika', 25);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Starlight'), 'Infinity', 50);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Pixelplex'), 'Astral', 25);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Cinematica'), 'Orion', 15);

INSERT INTO SALA

VALUES (cinema\_seq.nextval, (SELECT id\_cinema FROM CINEMA WHERE nume\_cinema = 'Flashback'), 'Mega', 30);

## TABELA FILM:

| ID_FILM | TITLU_FILM         | DURATA | GEN     | DESCRIERE   |
|---------|--------------------|--------|---------|---|
| 13      | Dirty Dancing      | 100    | Romance | The movie showcases a forbidden romance, set against the backdrop of a dance competition.   |
| 14      | Back to the Future | 116    | Sci-Fi  | (null)  |
| 15      | GhostBusters       | 107    | Comedy  | The movie features a group of friends who start a ghost-catching business in New York City. |
| 16      | The Breakfast Club | 92     | Drama   | (null)  |
| 17      | Dead Poets Society | 128    | Drama   | The movie explores themes of conformity, self-discovery, and the power of literature.       |
| 18      | The Terminator     | 108    | Action  | (null)  |
| 19      | The Shining        | 142    | Horror  | (null)  |

CREATE TABLE FILM

( id\_film NUMBER(5) CONSTRAINT PKEY\_FILM PRIMARY KEY,

titlu\_film VARCHAR(50) CONSTRAINT titlu\_film NOT NULL,

```

durata NUMBER(10) CONSTRAINT durata_film NOT NULL,
gen VARCHAR(20) CONSTRAINT gen_film NOT NULL,
descriere VARCHAR(100) CONSTRAINT descriere_film NULL
);

```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'Dirty Dancing', 100, 'Romance', 'The movie showcases a forbidden romance, set against the backdrop of a dance competition.');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'Back to the Future', 116, 'Sci-Fi', '');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'GhostBusters', 107, 'Comedy', 'The movie features a group of friends who start a ghost-catching business in New York City.');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'The Breakfast Club', 92, 'Drama', '');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'Dead Poets Society', 128, 'Drama', 'The movie explores themes of conformity, self-discovery, and the power of literature.');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'The Terminator', 108, 'Action', '');
```

```
INSERT INTO FILM
```

```
VALUES (cinema_seq.nextval, 'The Shining', 142, 'Horror', '');
```

## TABELA PROIECTIE:

| ID_PROIECTIE | ID_FILM | ID_SALA | ORA   |
|--------------|---------|---------|-------|
| 55           | 13      | 6       | 10:00 |
| 56           | 17      | 7       | 15:00 |
| 57           | 16      | 9       | 20:30 |
| 58           | 18      | 12      | 17:30 |
| 59           | 17      | 10      | 18:00 |
| 60           | 15      | 11      | 18:00 |
| 61           | 14      | 8       | 12:30 |
| 62           | 19      | 9       | 21:00 |
| 63           | 15      | 6       | 12:00 |
| 64           | 13      | 7       | 11:30 |

CREATE TABLE PROIECTIE

```
( id_proiectie NUMBER(5) CONSTRAINT PKEY_PROIECTIE PRIMARY KEY,  
  id_film NUMBER(5), CONSTRAINT fk_proiectie_film FOREIGN KEY(id_film) REFERENCES  
  FILM(id_film),  
  id_sala NUMBER(5), CONSTRAINT fk_proiectie_sala FOREIGN KEY(id_sala) REFERENCES  
  SALA(id_sala),  
  ora VARCHAR(10) CONSTRAINT ora_proiectie NOT NULL  
);
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'Dirty  
Dancing'), (SELECT id_sala FROM SALA WHERE nume = 'Ultra'), '10:00');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'Dead Poets  
Society'), (SELECT id_sala FROM SALA WHERE nume = 'Galaxy'), '15:00');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'The Breakfast  
Club'), (SELECT id_sala FROM SALA WHERE nume = 'Infinity'), '20:30');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'The  
Terminator'), (SELECT id_sala FROM SALA WHERE nume = 'Mega'), '17:30');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'Dead Poets  
Society'), (SELECT id_sala FROM SALA WHERE nume = 'Astral'), '18:00');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'GhostBusters'),  
(SELECT id_sala FROM SALA WHERE nume = 'Orion'), '18:00');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'Back to the  
Future'), (SELECT id_sala FROM SALA WHERE nume = 'Epika'), '12:30');
```

INSERT INTO PROIECTIE

```
VALUES (cinema_seq.nextval, (SELECT id_film FROM FILM WHERE titlu_film = 'The Shining'),  
(SELECT id_sala FROM SALA WHERE nume = 'Infinity'), '21:00');
```

INSERT INTO PROIECTIE

VALUES (cinema\_seq.nextval, (SELECT id\_film FROM FILM WHERE titlu\_film = 'GhostBusters'),  
(SELECT id\_sala FROM SALA WHERE nume = 'Ultra'), '12:00');

INSERT INTO PROIECTIE

VALUES (cinema\_seq.nextval, (SELECT id\_film FROM FILM WHERE titlu\_film = 'Dirty  
Dancing'), (SELECT id\_sala FROM SALA WHERE nume = 'Galaxy'), '11:30');

## TABELA BILET:

| ID_BILET | ID_PROIECTIE | ID_COMANDA | RAND | LOC |
|----------|--------------|------------|------|-----|
| 80       | 55           | 45 E       |      | 5   |
| 81       | 55           | 45 E       |      | 6   |
| 82       | 57           | 46 C       |      | 10  |
| 83       | 57           | 46 C       |      | 11  |
| 84       | 64           | 47 D       |      | 4   |
| 85       | 64           | 47 E       |      | 1   |
| 86       | 56           | 48 C       |      | 5   |
| 87       | 58           | 49 A       |      | 2   |
| 88       | 60           | 50 C       |      | 5   |
| 89       | 61           | 51 A       |      | 3   |
| 90       | 62           | 52 B       |      | 4   |
| 91       | 63           | 53 A       |      | 5   |
| 92       | 58           | 54 D       |      | 3   |

CREATE TABLE BILET

( id\_bilet NUMBER(5) CONSTRAINT PKEY\_BILET PRIMARY KEY,  
id\_proiectie NUMBER(5), CONSTRAINT fk\_bilet\_proiectie FOREIGN KEY(id\_proiectie)  
REFERENCES PROIECTIE(id\_proiectie),  
id\_comanda NUMBER(5), CONSTRAINT fk\_bilet\_comanda FOREIGN KEY(id\_comanda)  
REFERENCES COMANDA(id\_comanda),  
rand VARCHAR(2) CONSTRAINT rand\_bilet NOT NULL,  
loc NUMBER(10) CONSTRAINT loc\_bilet NOT NULL  
);

INSERT INTO BILET

VALUES (cinema\_seq.nextval, 55, 45, 'E', 5);

INSERT INTO BILET

VALUES (cinema\_seq.nextval, 55, 45, 'E', 6);

INSERT INTO BILET



```

VALUES (cinema_seq.nextval, 57, 46, 'C', 10);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 57, 46, 'C', 11);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 64, 47, 'D', 4);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 64, 47, 'E', 1);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 56, 48, 'C', 5);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 58, 49, 'A', 2);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 60, 50, 'C', 5);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 61, 51, 'A', 3);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 62, 52, 'B', 4);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 63, 53, 'A', 5);
INSERT INTO BILET
VALUES (cinema_seq.nextval, 58, 54, 'D', 3);

```

## TABELA PRODUS:

| ID_PRODUS | NUME_PRODUS | PRET | STOC |
|-----------|-------------|------|------|
| 35        | popcorn     | 20   | 100  |
| 36        | nachos      | 25   | 100  |
| 37        | suc         | 10   | 150  |
| 38        | apa         | 5    | 150  |
| 39        | alune       | 15   | 100  |

```
CREATE TABLE PRODUS
```

```
( id_produs NUMBER(5) CONSTRAINT PKEY_PRODUS PRIMARY KEY,
```

```

    nume_produs VARCHAR(30) CONSTRAINT nume_produs NOT NULL,
    pret NUMBER(10) CONSTRAINT pret_produs NOT NULL,
    stoc NUMBER(10) CONSTRAINT stoc_produs NOT NULL
);

INSERT INTO PRODUS
VALUES (cinema_seq.nextval, 'popcorn', 20, 100);

INSERT INTO PRODUS
VALUES (cinema_seq.nextval, 'nachos', 25, 100);

INSERT INTO PRODUS
VALUES (cinema_seq.nextval, 'suc', 10, 150);

INSERT INTO PRODUS
VALUES (cinema_seq.nextval, 'apa', 5, 150);

INSERT INTO PRODUS
VALUES (cinema_seq.nextval, 'alune', 15, 100);

```

### TABELA CLIENT:

| ID_CLIENT | NUME_CLIENT | PRENUME_CLIENT | TELEFON    | EMAIL                    |
|-----------|-------------|----------------|------------|--------------------------|
| 25        | Stan        | Bianca         | 0719472600 | bianastan123@gmail.com   |
| 26        | Dumitrescu  | Eric           | 0709012574 | dumitrescueric@yahoo.com |
| 27        | Dobre       | Stefan         | 0799006584 | stefan27@gmail.com       |
| 28        | Paun        | Raluca         | 0771063900 | ralucapaun@gmail.com     |
| 29        | Munteanu    | Laura          | 0761352678 | lalamunteanu@gmail.com   |
| 30        | Popescu     | Andrei         | 0721123456 | andrei.popescu@gmail.com |
| 31        | Ionescu     | Maria          | 0731987654 | maria.ionescu@gmail.com  |
| 32        | Stoica      | Larisa         | 0761890123 | sstoica@gmail.com        |
| 33        | Ciurnea     | Ioan           | 0761352875 | ioan1234@yahoo.com       |
| 34        | Pestritu    | Monica         | 0761000678 | monicapestritu@gmail.com |

```

CREATE TABLE CLIENT
( id_client NUMBER(5) CONSTRAINT PKEY_CLIENT PRIMARY KEY,
  nume_client VARCHAR(20) CONSTRAINT nume_client NOT NULL,
  prenume_client VARCHAR(20) CONSTRAINT prenume_client NOT NULL,
  telefon VARCHAR(10) CONSTRAINT telefon_client UNIQUE NOT NULL,
  email VARCHAR(30) CONSTRAINT email_client UNIQUE NOT NULL
);

INSERT INTO CLIENT

```

```

VALUES (cinema_seq.nextval, 'Stan', 'Bianca', '0719472600', 'bianastan123@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Dumitrescu', 'Eric', '0709012574',
'dumitrescueric@yahoo.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Dobre', 'Stefan', '0799006584', 'stefan27@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Paun', 'Raluca', '0771063900', 'ralucapaun@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Munteanu', 'Laura', '0761352678', 'lalamunteanu@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Popescu', 'Andrei', '0721123456', 'andrei.popescu@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Ionescu', 'Maria', '0731987654', 'maria.ionescu@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Stoica', 'Larisa', '0761890123', 'sstoica@gmail.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Ciurnea', 'Ioan', '0761352875', 'ioan1234@yahoo.com');
INSERT INTO CLIENT
VALUES (cinema_seq.nextval, 'Pestritu', 'Monica', '0761000678', 'monicapestritu@gmail.com');

```

## TABELA ANGAJAT:

| ID_ANGAJAT | NUME_ANGAJAT | PRENUME_ANGAJAT | FUNCTIE   | SALARIU |
|------------|--------------|-----------------|-----------|---------|
| 40         | Ionescu      | Mihai           | casier    | 3000    |
| 41         | Vasile       | Iustina         | casier    | 3000    |
| 42         | Popescu      | Dragos          | casier    | 3000    |
| 43         | Dascalu      | Andrei          | tehnician | 4000    |
| 44         | Olteanu      | Maria           | curatenie | 1500    |

CREATE TABLE ANGAJAT

```

( id_angajat NUMBER(5) CONSTRAINT PKEY_ANGAJAT PRIMARY KEY,
  nume_angajat VARCHAR(20) CONSTRAINT nume_angajat NOT NULL,
  prenume_angajat VARCHAR(20) CONSTRAINT prenume_angajat NOT NULL,
  functie VARCHAR(20) CONSTRAINT functie_angajat NOT NULL,
  salariu NUMBER(10) CONSTRAINT salariu_angajat NOT NULL
);

```

```

INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Ionescu', 'Mihai', 'casier', 3000);
INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Vasile', 'Iustina', 'casier', 3000);
INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Popescu', 'Dragos', 'casier', 3000);
INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Dascalu', 'Andrei', 'tehnician', 4000);
INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Olteanu', 'Maria', 'curatenie', 1500);

```

### TABELA COMANDĂ:

| ID_COMA... | ID_CLIENT | ID_ANGA... | TOTAL | METODA_... | DATA      |
|------------|-----------|------------|-------|------------|-----------|
| 45         | 25        | 42         | 0     | card       | 12-APR-23 |
| 46         | 26        | 41         | 0     | cash       | 15-APR-23 |
| 47         | 27        | 40         | 0     | card       | 23-APR-23 |
| 48         | 28        | 41         | 0     | cash       | 10-MAY-23 |
| 49         | 29        | 42         | 0     | card       | 24-MAY-23 |
| 50         | 30        | 40         | 0     | card       | 25-MAY-23 |
| 51         | 31        | 42         | 0     | cash       | 25-MAY-23 |
| 52         | 32        | 41         | 0     | cash       | 26-MAY-23 |
| 53         | 33        | 40         | 0     | card       | 29-MAY-23 |
| 54         | 34        | 41         | 0     | card       | 01-JUN-23 |

```

CREATE TABLE COMANDA
( id_comanda NUMBER(5) CONSTRAINT PKEY_COMANDA PRIMARY KEY,
  id_client NUMBER(5), CONSTRAINT fk_comanda_client FOREIGN KEY(id_client)
REFERENCES CLIENT(id_client),
  id_angajat NUMBER(5), CONSTRAINT fk_comanda_angajat FOREIGN KEY(id_angajat)
REFERENCES ANGAJAT(id_angajat),
  total NUMBER(10) DEFAULT 0 CONSTRAINT total_comanda NOT NULL,
  metoda_plata VARCHAR(10) DEFAULT 'cash' CONSTRAINT metoda_plata_comanda
CHECK(metoda_plata IN ('cash', 'card')) NOT NULL,
  data DATE CONSTRAINT data_comanda NOT NULL
);

```

```

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0719472600'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Popescu'),
'card', TO_DATE('12-04-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0709012574'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Vasile'),
TO_DATE('15-04-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0799006584'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Ionescu'), 'card',
TO_DATE('23-04-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0771063900'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Vasile'),
TO_DATE('10-05-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0761352678'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Popescu'),
'card', TO_DATE('24-05-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0721123456'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Ionescu'), 'card',
TO_DATE('25-05-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0731987654'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Popescu'),
TO_DATE('25-05-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0761890123'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Vasile'),
TO_DATE('26-05-2023', 'DD-MM-YYYY'));

INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0761352875'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Ionescu'), 'card',
TO_DATE('29-05-2023', 'DD-MM-YYYY'));

```

```
INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0761000678'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Vasile'), 'card',
TO_DATE('01-06-2023', 'DD-MM-YYYY'));
```

### TABELA DETALIU\_COMANDĂ:

| ID_DETALIU_COMANDA | ID_PRODUS | ID_COMANDA | CANTITATE |
|--------------------|-----------|------------|-----------|
| 93                 | 35        | 45         | 2         |
| 94                 | 37        | 45         | 2         |
| 95                 | 38        | 46         | 3         |
| 96                 | 39        | 47         | 1         |
| 97                 | 36        | 47         | 1         |
| 98                 | 39        | 48         | 2         |
| 99                 | 35        | 49         | 4         |
| 100                | 39        | 50         | 1         |
| 101                | 36        | 51         | 2         |
| 102                | 37        | 52         | 2         |
| 103                | 36        | 52         | 1         |
| 104                | 35        | 53         | 1         |
| 105                | 38        | 54         | 2         |
| 106                | 35        | 54         | 1         |

```
CREATE TABLE DETALIU_COMANDA
```

```
( id_detaliu_comanda NUMBER(5) CONSTRAINT PKEY_DETALIU_COMANDA PRIMARY
KEY,
```

```
id_produs NUMBER(5), CONSTRAINT fk_detaliu_comanda_produs FOREIGN
KEY(id_produs) REFERENCES PRODUS(id_produs),
```

```
id_comanda NUMBER(5), CONSTRAINT fk_detaliu_comanda_comanda FOREIGN
KEY(id_comanda) REFERENCES COMANDA(id_comanda),
```

```
cantitate NUMBER(10) CONSTRAINT cantitate_comanda NOT NULL
```

```
);
```

```
INSERT INTO DETALIU_COMANDA
```

```
VALUES (cinema_seq.nextval, (SELECT id_produs FROM PRODUS WHERE nume_produs =
'popcorn'), 45, 2);
```

```
INSERT INTO DETALIU_COMANDA
```

```
VALUES (cinema_seq.nextval, (SELECT id_produs FROM PRODUS WHERE nume_produs =
'suc'), 45, 2);
```

```
INSERT INTO DETALIU_COMANDA
```

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'apa'), 46, 3);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'alune'), 47, 1);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'nachos'), 47, 1);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'alune'), 48, 2);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'popcorn'), 49, 4);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'alune'), 50, 1);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'nachos'), 51, 2);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'suc'), 52, 2);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'nachos'), 52, 1);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'popcorn'), 53, 1);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produ FROM PRODUS WHERE nume\_produ = 'apa'), 54, 2);

INSERT INTO DETALIU\_COMANDA

VALUES (cinema\_seq.nextval, (SELECT id\_produș FROM PRODUS WHERE nume\_produș = 'popcorn'), 54, 1);

## 12. Implementarea a 5 cereri SQL:

1) Afișează titlul, genul, durata și descrierea (în cazul în care există o descriere disponibilă) filmelor și o coloană suplimentară numită "Comparație" care să indice dacă durata fiecărui film este mai lungă, mai scurtă sau egală cu durata medie a filmelor din același gen. Filmele sunt ordonate în funcție de durată, în mod descrescător.

```
SELECT f.titlu_film AS "Titlu film",
       f.gen AS "Gen",
       f.durata AS "Durata",
       NVL(f.descriere, 'Filmul nu are o descriere.') AS "Descriere",
       lungime_medie.durata_medie,
       DECODE(SIGN(f.durata - lungime_medie.durata_medie),
              1, 'Mai lung',
              0, 'Egal',
              -1, 'Mai scurt') AS "Comparatie"
FROM FILM f
JOIN (
  SELECT gen, AVG(durata) AS durata_medie
  FROM FILM
  GROUP BY gen
) lungime_medie ON f.gen = lungime_medie.gen
ORDER BY f.durata DESC;
```

| Titlu film         | Gen     | Durata | Descriere   | DURATA_MEDIE | Comparatie |
|--------------------|---------|--------|---|--------------|------------|
| The Shining        | Horror  | 142    | Filmul nu are o descriere.  | 142          | Egal       |
| Dead Poets Society | Drama   | 128    | The movie explores themes of conformity, self-discovery, and the power of literature.       | 110          | Mai lung   |
| Back to the Future | Sci-Fi  | 116    | Filmul nu are o descriere.  | 116          | Egal       |
| The Terminator     | Action  | 108    | Filmul nu are o descriere.  | 108          | Egal       |
| Ghostbusters       | Comedy  | 107    | The movie features a group of friends who start a ghost-catching business in New York City. | 107          | Egal       |
| Dirty Dancing      | Romance | 100    | The movie showcases a forbidden romance, set against the backdrop of a dance competition.   | 100          | Egal       |
| The Breakfast Club | Drama   | 92     | Filmul nu are o descriere.  | 110          | Mai scurt  |

2) Afișează numele cinematografele și numărul de proiecții ale filmelor în funcție de oră (program matinal, program de zi și program de seara).

```
SELECT c.nume_cinema AS "Nume cinema",
       SUM (CASE WHEN p.ora < '12:00' THEN 1 ELSE 0 END ) AS "Program matinal",
```



```

SUM (CASE WHEN p.ora >= '12:00' AND p.ora < '18:00' THEN 1 ELSE 0 END ) AS "Program de zi",
SUM (CASE WHEN p.ora >= '18:00' THEN 1 ELSE 0 END ) AS "Program de zi"
FROM CINEMA c
JOIN SALA s ON s.id_cinema = c.id_cinema
JOIN PROIECTIE p ON p.id_sala = s.id_sala
GROUP BY c.nume_cinema;

```

| Nume cinema | Program matinal | Program de zi | Program de zi_1 |
|-------------|-----------------|---------------|-----------------|
| Flashback   | 0               | 1             | 0               |
| Starlight   | 0               | 1             | 2               |
| Pixelplex   | 0               | 0             | 1               |
| Retroplex   | 2               | 2             | 0               |
| Cinematica  | 0               | 0             | 1               |

3) Afișează angajații care au funcția de casier și lucrează la un cinema din București, respectiv numărul de comenzi de care s-au ocupat aceștia în anul 2023.

```

SELECT a.nume_angajat AS "Nume", a.prenume_angajat AS "Prenume",
(
SELECT COUNT(*)
FROM COMANDA cmd
JOIN BILET b ON b.id_comanda = cmd.id_comanda
JOIN PROIECTIE p ON b.id_roiectie = p.id_proiectie
JOIN SALA s ON s.id_sala = p.id_sala
JOIN CINEMA c ON c.id_cinema = s.id_cinema
WHERE INITCAP(c.oras) = 'Bucuresti' AND TO_CHAR(cmd.data, 'YYYY') = 2023
AND cmd.id_angajat = a.id_angajat
) AS Numarcomenzi
FROM ANGAJAT a
WHERE LOWER(a.functie) = 'casier'
ORDER BY NumarComenzi DESC;

```

| Nume    | Prenume | NUMARCOMENZI |
|---------|---------|--------------|
| Vasile  | Iustina | 4            |
| Popescu | Dragos  | 3            |
| Ionescu | Mihai   | 3            |

4) Afisează codul și numele angajaților care au procesat cel puțin o comandă cu mai mult de un produs.

```
SELECT id_angajat AS "Cod angajat", CONCAT(UPPER(num_ angajat), CONCAT(' ', prenume_angajat))  
AS "Nume angajat"
```

```
FROM ANGAJAT
```

```
WHERE id_angajat IN (
```

```
    SELECT DISTINCT c.id_angajat
```

```
    FROM COMANDA c
```

```
    JOIN DETALIU_COMANDA dc ON dc.id_comanda = c.id_comanda
```

```
    GROUP BY c.id_comanda, c.id_angajat
```

```
    HAVING COUNT(dc.id_produ_s) > 1 );
```

| Cod angajat | Nume angajat   |
|-------------|----------------|
| 40          | IONESCU Mihai  |
| 41          | VASILE Iustina |
| 42          | POPESCU Dragos |

5) Pentru fiecare produs care are prețu\_l maxim să se afișeze toate comenzile în care apare.

```
WITH PretMaxim AS (
```

```
    SELECT MAX(pret) AS pret_maxim
```

```
    FROM PRODUS
```

```
), ComenziCuPretMaxim AS (
```

```
    SELECT c.id_comanda, c.id_client, c.id_angajat, c.total, c.metoda_plata, c.data
```

```
    FROM COMANDA c
```

```
    JOIN DETALIU_COMANDA dc ON c.id_comanda = dc.id_comanda
```

```
    JOIN PRODUS p ON dc.id_produ_s = p.id_produ_s
```

```
    WHERE p.pret = (
```

```
        SELECT pret_maxim
```

```
        FROM PretMaxim ) )
```

```
SELECT *
```

```
FROM ComenziCuPretMaxim;
```

| ID_COMANDA | ID_CLIENT | ID_ANGAJAT | TOTAL | METODA_PLATA | DATA      |
|------------|-----------|------------|-------|--------------|-----------|
| 47         | 27        | 40         | 40    | card         | 23-APR-23 |
| 51         | 31        | 42         | 50    | cash         | 25-MAY-23 |
| 52         | 32        | 41         | 45    | cash         | 26-MAY-23 |
| 54         | 34        | 41         | 30    | card         | 01-JUN-23 |

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

1) - UPDATE - Adauga pretului total pentru comenzi.

UPDATE COMANDA c

SET c.total = (

SELECT SUM(p.pret \* dc.cantitate) as total

FROM DETALIU\_COMANDA dc

JOIN PRODUS p ON dc.id\_produș = p.id\_produș

WHERE dc.id\_comanda = c.id\_comanda

);

| ID_COMA... | ID_CLIENT | ID_ANGA... | TOTAL | METODA_... | DATA      |
|------------|-----------|------------|-------|------------|-----------|
| 45         | 25        | 42         | 0     | card       | 12-APR-23 |
| 46         | 26        | 41         | 0     | cash       | 15-APR-23 |
| 47         | 27        | 40         | 0     | card       | 23-APR-23 |
| 48         | 28        | 41         | 0     | cash       | 10-MAY-23 |
| 49         | 29        | 42         | 0     | card       | 24-MAY-23 |
| 50         | 30        | 40         | 0     | card       | 25-MAY-23 |
| 51         | 31        | 42         | 0     | cash       | 25-MAY-23 |
| 52         | 32        | 41         | 0     | cash       | 26-MAY-23 |
| 53         | 33        | 40         | 0     | card       | 29-MAY-23 |
| 54         | 34        | 41         | 0     | card       | 01-JUN-23 |



| ID_COMA... | ID_CLIENT | ID_ANGA... | TOTAL | METODA_... | DATA      |
|------------|-----------|------------|-------|------------|-----------|
| 45         | 25        | 42         | 60    | card       | 12-APR-23 |
| 46         | 26        | 41         | 15    | cash       | 15-APR-23 |
| 47         | 27        | 40         | 40    | card       | 23-APR-23 |
| 48         | 28        | 41         | 30    | cash       | 10-MAY-23 |
| 49         | 29        | 42         | 80    | card       | 24-MAY-23 |
| 50         | 30        | 40         | 15    | card       | 25-MAY-23 |
| 51         | 31        | 42         | 50    | cash       | 25-MAY-23 |
| 52         | 32        | 41         | 45    | cash       | 26-MAY-23 |
| 53         | 33        | 40         | 20    | card       | 29-MAY-23 |
| 54         | 34        | 41         | 30    | card       | 01-JUN-23 |

2) - UPDATE - Crește salariul cu 100 pentru angajații care au vândut mai mult de două tipuri de produse diferite.

UPDATE ANGAJAT a

SET a.salariu = a.salariu + 100

WHERE a.id\_angajat IN (

SELECT DISTINCT c.id\_angajat

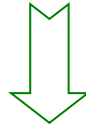
FROM COMANDA c

JOIN DETALIU\_COMANDA dc ON c.id\_comanda = dc.id\_comanda

GROUP BY c.id\_angajat, c.id\_comanda

HAVING COUNT(\*) > 2);

| ID_ANGAJAT | NUME_ANGAJAT | PRENUME_ANGAJAT | FUNCTIE   | SALARIU |
|------------|--------------|-----------------|-----------|---------|
| 40         | Ionescu      | Mihai           | casier    | 3000    |
| 41         | Vasile       | Iustina         | casier    | 3000    |
| 42         | Popescu      | Dragos          | casier    | 3000    |
| 43         | Dascalu      | Andrei          | tehnician | 4000    |
| 44         | Olteanu      | Maria           | curatenie | 1500    |



| ID_ANGAJAT | NUME_ANGAJAT | PRENUME_ANGAJAT | FUNCTIE   | SALARIU |
|------------|--------------|-----------------|-----------|---------|
| 40         | Ionescu      | Mihai           | casier    | 3000    |
| 41         | Vasile       | Iustina         | casier    | 3100    |
| 42         | Popescu      | Dragos          | casier    | 3000    |
| 43         | Dascalu      | Andrei          | tehnician | 4000    |
| 44         | Olteanu      | Maria           | curatenie | 1500    |

3) - DELETE - Șterge clientul care nu a plasat comenzi.

DELETE CLIENT c

WHERE NOT EXISTS (

SELECT \*

FROM COMANDA cmd

WHERE cmd.id\_client = c.id\_client );

| ID_CLIENT | NUME_CLIENT | PRENUME_CLIENT | TELEFON    | EMAIL                    |
|-----------|-------------|----------------|------------|--------------------------|
| 25        | Stan        | Bianca         | 0719472600 | bianastan123@gmail.com   |
| 26        | Dumitrescu  | Eric           | 0709012574 | dumitrescueric@yahoo.com |
| 27        | Dobre       | Stefan         | 0799006584 | stefan27@gmail.com       |
| 28        | Paun        | Raluca         | 0771063900 | ralucapaun@gmail.com     |
| 29        | Munteanu    | Laura          | 0761352678 | lalamunteanu@gmail.com   |
| 30        | Popescu     | Andrei         | 0721123456 | andrei.popescu@gmail.com |
| 31        | Ionescu     | Maria          | 0731987654 | maria.ionescu@gmail.com  |
| 32        | Stoica      | Larisa         | 0761890123 | sstoica@gmail.com        |
| 33        | Ciurnea     | Ioan           | 0761352875 | ioan1234@yahoo.com       |
| 34        | Pestritu    | Monica         | 0761000678 | monicapestritu@gmail.com |
| 109       | Calinescu   | Florin         | 0763000779 | fflorinn@yahoo.com       |



| ID_CLIENT | NUME_CLIENT | PRENUME_CLIENT | TELEFON    | EMAIL                    |
|-----------|-------------|----------------|------------|--------------------------|
| 25        | Stan        | Bianca         | 0719472600 | bianastan123@gmail.com   |
| 26        | Dumitrescu  | Eric           | 0709012574 | dumitrescueric@yahoo.com |
| 27        | Dobre       | Stefan         | 0799006584 | stefan27@gmail.com       |
| 28        | Paun        | Raluca         | 0771063900 | ralucapaun@gmail.com     |
| 29        | Munteanu    | Laura          | 0761352678 | lalamunteanu@gmail.com   |
| 30        | Popescu     | Andrei         | 0721123456 | andrei.popescu@gmail.com |
| 31        | Ionescu     | Maria          | 0731987654 | maria.ionescu@gmail.com  |
| 32        | Stoica      | Larisa         | 0761890123 | sstoica@gmail.com        |
| 33        | Ciurnea     | Ioan           | 0761352875 | ioan1234@yahoo.com       |
| 34        | Pestritu    | Monica         | 0761000678 | monicapestritu@gmail.com |

! Pentru exercițiile de mai jos am introdus următoarele înregistrări:

INSERT INTO BILET

VALUES (cinema\_seq.nextval, 56, 111, 'C', 6);

```
INSERT INTO COMANDA(id_comanda, id_client, id_angajat, metoda_plata, data)
VALUES (cinema_seq.nextval, (SELECT id_client FROM CLIENT WHERE telefon =
'0719472600'), (SELECT id_angajat FROM ANGAJAT WHERE nume_angajat = 'Popescu'),
'card', TO_DATE('12-05-2023', 'DD-MM-YYYY'));
```

```
INSERT INTO ANGAJAT
VALUES (cinema_seq.nextval, 'Muntean', 'Cristian', 'casier', 3000);
```

#### 14. Crearea unei vizualizari complexe. Exemplu de operație LMD permisă și nepermisă:

Operația LMD (Limbaj de Manipulare al Datelor) se referă la posibilitatea de a modifica sau șterge datele unei entități într-o bază de date. Pentru a exemplifica o vizualizare complexă, să presupunem că dorim să obținem o listă a tuturor proiecțiilor unui anumit film dintr-un cinematograful specific. Aceasta implică conectarea mai multor entități și interogarea bazei de date pentru a obține rezultatele dorite. Iată un exemplu de interogare pentru această vizualizare.

```
CREATE OR REPLACE VIEW COMENZI_CLIENT AS
SELECT c.nume_client, c.prenume_client, c.telefon, c.email, cmd.total, cmd.metoda_plata, cmd.data
FROM CLIENT c
INNER JOIN COMANDA cmd ON cmd.id_client = c.id_client;
```

#### ➤ Exemplu de operație LMD permisă:

```
SELECT * FROM COMENZI_CLIENT;
```

| NUME_CLIENT | PRENUME_CLIENT | TELEFON    | EMAIL                    | TOTAL | METODA_PLATA | DATA      |
|-------------|----------------|------------|--------------------------|-------|--------------|-----------|
| Stan        | Bianca         | 0719472600 | bianastan123@gmail.com   | 60    | card         | 12-APR-23 |
| Dumitrescu  | Eric           | 0709012574 | dumitrescueric@yahoo.com | 15    | cash         | 15-APR-23 |
| Dobre       | Stefan         | 0799006584 | stefan27@gmail.com       | 40    | card         | 23-APR-23 |
| Paun        | Raluca         | 0771063900 | ralucapaun@gmail.com     | 30    | cash         | 10-MAY-23 |
| Munteanu    | Laura          | 0761352678 | lalamunteanu@gmail.com   | 80    | card         | 24-MAY-23 |
| Popescu     | Andrei         | 0721123456 | andrei.popescu@gmail.com | 15    | card         | 25-MAY-23 |
| Ionescu     | Maria          | 0731987654 | maria.ionescu@gmail.com  | 50    | cash         | 25-MAY-23 |
| Stoica      | Larisa         | 0761890123 | sstoica@gmail.com        | 45    | cash         | 26-MAY-23 |
| Ciurnea     | Ioan           | 0761352875 | ioan1234@yahoo.com       | 20    | card         | 29-MAY-23 |
| Pestritu    | Monica         | 0761000678 | monicapestritu@gmail.com | 30    | card         | 01-JUN-23 |

### ➤ Exemplu de operație LMD nepermisă:

Nu putem efectua operații LMD de modificare (INSERT, UPDATE, DELETE) direct asupra înregistrărilor din vizualizarea complexă. Vizualizarea complexă este o reprezentare virtuală a datelor din baza de date, iar operațiile LMD de modificare sunt permise numai prin intermediul triggerilor INSTEAD OF care definesc cum vor fi afectate tabelele care compun view-ul de operația LMD respectivă.

```
UPDATE COMENZI_CLIENT  
SET metoda_plata = "cash"  
WHERE nume_client = "Stan";
```

Eroarea:

```
Error at Command Line : 10 Column : 21  
Error report -  
SQL Error: ORA-00904: "Stan": invalid identifier  
00904. 00000 - "%s: invalid identifier"  
*Cause:  
*Action:
```

## 15. Formați în limbaj natural și implementați în SQL:

- Exemplu de cerere ce utilizează operația OUTER JOIN. Afișează lista de casieri care nu au vândut bilete în luna MAI pentru proiecții programate după ora 18:00.

```
SELECT a.id_angajat AS "Cod angajat", CONCAT(UPPER(a.nume_angajat), CONCAT(' ',  
a.prenume_angajat)) AS "Nume angajat"  
FROM ANGAJAT a  
LEFT OUTER JOIN COMANDA c ON c.id_angajat = a.id_angajat AND TO_CHAR(c.data, 'MON') =  
'MAY'  
LEFT OUTER JOIN BILET b ON b.id_comanda = c.id_comanda  
LEFT OUTER JOIN PROIECTIE p ON p.id_proiectie = b.id_proiectie AND p.ora >= '18:00'  
WHERE a.functie = 'casier' AND p.id_proiectie is null  
AND a.id_angajat NOT IN (  
    SELECT c.id_angajat  
    FROM COMANDA c  
    JOIN BILET b ON b.id_comanda = c.id_comanda  
    JOIN PROIECTIE p ON p.id_proiectie = b.id_proiectie  
    WHERE TO_CHAR(c.data, 'MON') = 'MAY' AND p.ora >= '18:00'  
)  
GROUP BY a.id_angajat, a.nume_angajat, a.prenume_angajat;
```

| Cod angajat | Nume angajat     |
|-------------|------------------|
| 42          | POPESCU Dragos   |
| 110         | MUNTEAN Cristian |

- Exemplu de cerere ce utilizează operația DIVISION. Afișează angajații care nu au vândut toate tipurile de produse din inventar.

```
SELECT a.id_angajat AS "Cod angajat", (a.nume_angajat || ' ' || a.preume_angajat) AS "Nume angajat"
FROM ANGAJAT a
WHERE NOT EXISTS (
  SELECT p.id_produs
  FROM PRODUS p
  WHERE NOT EXISTS (
    SELECT c.id_angajat
    FROM COMANDA c
    JOIN DETALIU_COMANDA dc ON c.id_comanda = dc.id_comanda
    WHERE c.id_angajat = a.id_angajat AND p.id_produs = dc.id_produs
  )
);
```

| Cod angajat | Nume angajat   |
|-------------|----------------|
| 41          | Vasile Iustina |

- Exemplu de cerere ce implementează analiza TOP-N. Afișează top 3 angajați care au vândut bilete programate după ora 15:00.

```
SELECT * FROM (
  SELECT a.id_angajat AS "Cod angajat", (a.nume_angajat || ' ' || a.preume_angajat) AS "Nume angajat", COUNT(c.id_comanda) AS "Nr. comenzi"
  FROM ANGAJAT a
  JOIN COMANDA c ON c.id_angajat = a.id_angajat
  WHERE TO_CHAR(c.data, 'MON') = 'MAY'
  GROUP BY a.id_angajat, a.nume_angajat, a.preume_angajat
  ORDER BY "Nr. comenzi" DESC
)
WHERE ROWNUM <= 3;
```

| Cod angajat | Nume angajat   | Nr. comenzi |
|-------------|----------------|-------------|
| 42          | Popescu Dragos | 3           |
| 40          | Ionescu Mihai  | 2           |
| 41          | Vasile Iustina | 2           |

## 16. Optimizarea unei cereri, aplicând reguli ce derivă din proprietățile operatorilor algebrei relaționale.

### ➤ Neoptimizat:

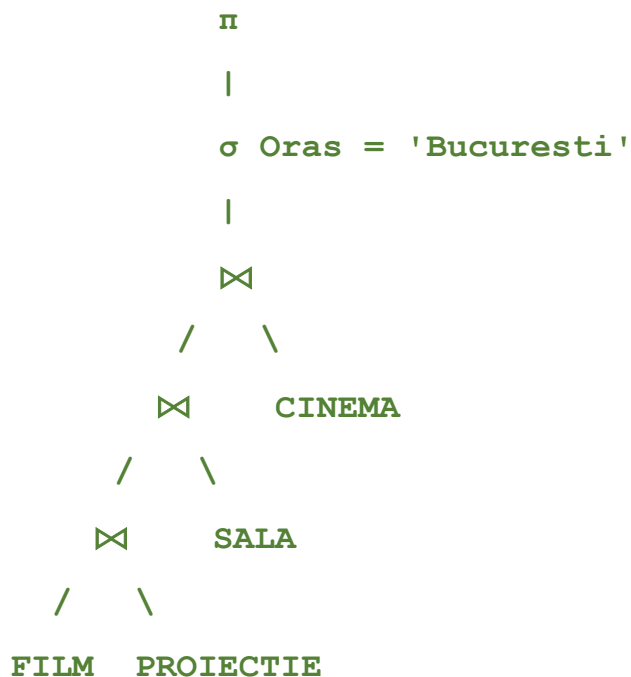
- Afișarea filmelor care sunt proiectate în cinematografe din București:

```
SELECT f.titlu_film AS "Titlu film", f.gen AS "Gen film", p.ora AS "Ora"  
FROM FILM f  
JOIN PROIECTIE p ON f.id_film = p.id_film  
JOIN SALA s ON s.id_sala = p.id_sala  
JOIN CINEMA c ON c.id_cinema = s.id_cinema  
WHERE c.oras = 'Bucuresti';
```

- Expresia algebrică:

$\pi$  titlu\_film AS "Titlu film", gen AS "Gen film", ora AS "Ora" (  
     $(\sigma \text{ oras} = \text{'Bucuresti'}(\text{FILM} \bowtie \text{PROIECTIE} \bowtie \text{SALA} \bowtie \text{CINEMA}))$   
)

- Arbore algebric:





### ➤ Optimizarea:

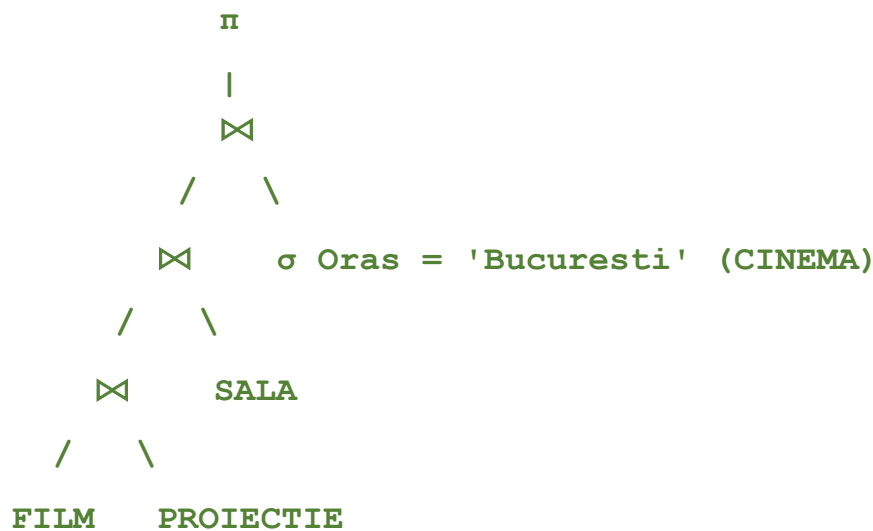
```
SELECT f.titlu_film AS "Titlu film", f.gen AS "Gen film", p.ora AS "Ora"
FROM FILM f
JOIN PROIECTIE p ON f.id_film = p.id_film
JOIN SALA s ON s.id_sala = p.id_sala
JOIN (SELECT * FROM CINEMA WHERE oras = 'Bucuresti') c ON c.id_cinema = s.id_cinema;
```

#### ▪ Expresia algebrică:

```
 $\pi$  titlu_film AS "Titlu film", gen AS "Gen film", ora AS "Ora" (
  FILM  $\bowtie$  PROIECTIE  $\bowtie$  (SALA  $\bowtie$  ( $\sigma$  oras = 'Bucuresti' (CINEMA)))
)
```

- Optimizarea constă în mutarea filtrului ( $\sigma$ ) mai aproape de operația de împerechere ( $\bowtie$ ), astfel încât să se reducă numărul de înregistrări prelucrate înainte de aplicarea filtrului.

#### ▪ Arbore algebric:



## 17. Optimizarea a două cereri utilizând indexare:

- **Exemplu 1** – (Afișarea filmelor care sunt proiectate în cinematografe din București)
- Fără indecși:

```
SELECT f.titlu_film AS "Titlu film", f.gen AS "Gen film", p.ora AS "Ora"
```

```

FROM FILM f
JOIN PROIECTIE p ON f.id_film = p.id_film
JOIN SALA s ON s.id_sala = p.id_sala
JOIN CINEMA c ON c.id_cinema = s.id_cinema
WHERE c.oras = 'Bucuresti';

```



SQL | All Rows Fetched: 7 in 0.005 seconds

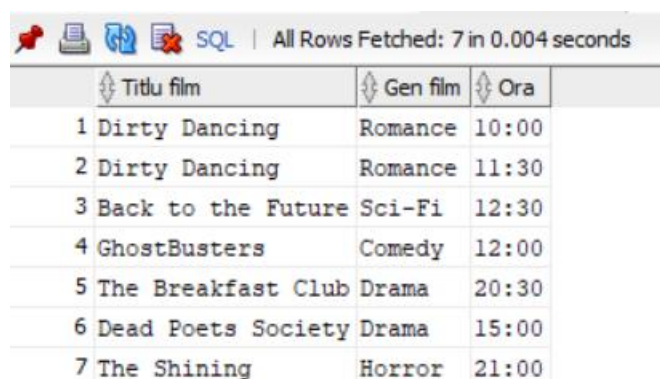
|   | Titlu film         | Gen film | Ora   |
|---|--------------------|----------|-------|
| 1 | Dirty Dancing      | Romance  | 10:00 |
| 2 | Dirty Dancing      | Romance  | 11:30 |
| 3 | Back to the Future | Sci-Fi   | 12:30 |
| 4 | GhostBusters       | Comedy   | 12:00 |
| 5 | The Breakfast Club | Drama    | 20:30 |
| 6 | Dead Poets Society | Drama    | 15:00 |
| 7 | The Shining        | Horror   | 21:00 |

- Cu indecși:

```

CREATE INDEX idx_cinema_oras ON CINEMA (oras);
CREATE INDEX idx_proiectie_id_film ON PROIECTIE (id_film);
CREATE INDEX idx_proiectie_id_sala ON PROIECTIE (id_sala);
CREATE INDEX idx_sale_id_cinema ON SALA (id_cinema);

```



SQL | All Rows Fetched: 7 in 0.004 seconds

|   | Titlu film         | Gen film | Ora   |
|---|--------------------|----------|-------|
| 1 | Dirty Dancing      | Romance  | 10:00 |
| 2 | Dirty Dancing      | Romance  | 11:30 |
| 3 | Back to the Future | Sci-Fi   | 12:30 |
| 4 | GhostBusters       | Comedy   | 12:00 |
| 5 | The Breakfast Club | Drama    | 20:30 |
| 6 | Dead Poets Society | Drama    | 15:00 |
| 7 | The Shining        | Horror   | 21:00 |

- **Exemplu 2** – (Afișarea casierilor și numărului de comenzi pe care le-au făcut în luna mai, ordonați descrescător în funcție de numărul de comenzi)
- Fără indecși:

```

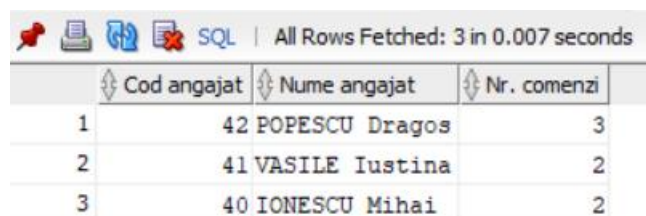
SELECT a.id_angajat AS "Cod angajat", CONCAT(UPPER(a.nume_angajat), CONCAT(' ', a.prenume_angajat)) AS "Nume angajat", COUNT(c.id_comanda) AS "Nr. comenzi"

```

```

FROM ANGAJAT a
JOIN COMANDA c ON c.id_angajat = a.id_angajat
WHERE TO_CHAR(c.data, 'MON') = 'MAY'
GROUP BY a.id_angajat, a.ume_angajat, a.prenume_angajat
ORDER BY "Nr. comenzi" DESC;

```



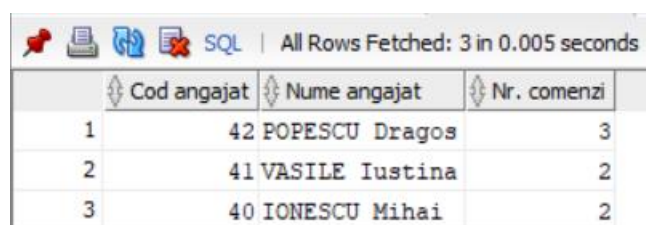
|   | Cod angajat | Nume angajat   | Nr. comenzi |
|---|-------------|----------------|-------------|
| 1 | 42          | POPESCU Dragos | 3           |
| 2 | 41          | VASILE Iustina | 2           |
| 3 | 40          | IONESCU Mihai  | 2           |

- Cu indecși:

```

CREATE INDEX idx_comanda_id_angajat ON COMANDA (id_angajat);
CREATE INDEX idx_comanda_data ON COMANDA (data);

```



|   | Cod angajat | Nume angajat   | Nr. comenzi |
|---|-------------|----------------|-------------|
| 1 | 42          | POPESCU Dragos | 3           |
| 2 | 41          | VASILE Iustina | 2           |
| 3 | 40          | IONESCU Mihai  | 2           |

## 18. Realizarea normalizării:

- Forma normală Boyce-Codd (BCNF):

Forma normală Boyce-Codd se bazează pe dependențele funcționale care iau în considerare toate cheile candidat dintr-o relație. Formele FN3 și BCNF sunt echivalente pentru relațiile cu o singură cheie candidat.

- Forma normală 4 (FN4):

Forma normală 4 elimină redundanțele datorate relațiilor many-to-many, adică datorate dependenței multiple. Pentru ca o relație să fie în FN4 trebuie să fie în BCNF și să nu conțină relații many-to-many independente.

- Forma normală 5 (FN5):

O relație este în FN5 dacă și numai dacă orice dependență de uniune a relației este o consecință a unei chei candidat a relației. Orice relație care este în FN5 este și în FN4. Pentru a preciza dacă o relație este în FN5, trebuie să cunoaștem cheile

candidate si toate dependentele de uniune ale relației. Aducerea în FN5 înseamnă eliminarea join dependențelor.

➤ Denormalizarea:

Denormalizarea este procesul invers procesului de normalizare și are rolul de a realiza executarea mai rapidă a interogărilor prin introducerea redundanței. Astfel, se pune accentul pe rapiditatea analizei și se scade numărul de tabele.