

# **Netflip**

## **Proiect Baze de Date**

Florin Craciun

9 iulie 2023

# Cuprins

<b>1</b>	<b>Tema Proiectului</b>	<b>1</b>
1.1	Introducere . . . . .	1
1.2	Entitati si attribute . . . . .	1
1.3	Relatii . . . . .	3
<b>2</b>	<b>Prezentarea constrangerilor impuse asupra modelului</b>	<b>4</b>
<b>3</b>	<b>Descrierea entitatilor</b>	<b>5</b>
<b>4</b>	<b>Descrierea relatiilor</b>	<b>6</b>
<b>5</b>	<b>Descrierea atributelor</b>	<b>7</b>
<b>6</b>	<b>Diagrama entitate-relatie</b>	<b>10</b>
<b>7</b>	<b>Diagrama conceptuala</b>	<b>11</b>
<b>8</b>	<b>Enumerarea schemelor relationale corespunzatoare diagramei conceptuale</b>	<b>12</b>
<b>9</b>	<b>Realizarea normalizarii pana la forma normala 3</b>	<b>13</b>
9.1	Forma normala 1 (FN1) . . . . .	13
9.2	Forma normala 2 (FN2) . . . . .	13
9.3	Forma normala 3 (FN3) . . . . .	13
<b>10</b>	<b>Crearea de tabele SQL si inserarea de date coerente in acestea</b>	<b>14</b>
10.1	Abonamente . . . . .	14
10.2	Utilizator . . . . .	14
10.3	Metode_de_plata . . . . .	15
10.4	Profiluri . . . . .	15
10.5	Filme . . . . .	16
10.6	Seriale . . . . .	17
10.7	Distributie . . . . .	17
10.8	Istoric_Vizionari . . . . .	18
10.9	Favorite . . . . .	19
10.10	Genuri . . . . .	20
10.11	Recenzii . . . . .	21
<b>11</b>	<b>Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele</b>	<b>23</b>
<b>12</b>	<b>Formulati in limbja natural si implementati 5 cereri SQL complexe</b>	<b>24</b>
12.1	Sa se afiseze utilizatorii care nu au abonamentul gratuit. . . . .	24
12.2	Sa se afiseze utilizatorii ce au vizionat un serial thriller in luna iulie . . . . .	24
12.3	Sa se afiseze pentru fiecare film data de lansare si daca este de actiune. . . . .	25
12.4	Sa se afiseze pentru fiecare utilizator numarul de seriale si numarul de filme favorite insumate. (Exp: Daca are 2 seriale favorite, atunci se afiseaza 'doua') . . .	25
12.5	Sa se afiseze numarul de utilizatori ce au vizionat vreun film lansat in anul 2000	26

<b>13 Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri</b>	<b>28</b>
13.1 Sa se schimbe genul serialelor cu evaluarea 5 in Horror . . . . .	28
13.2 Abonamentul utilizatorilor ce au lasat recenzii de mai putin de 70 de caractere sa fie retrogradat la cel gratis . . . . .	28
13.3 Sa se stearga abonamentele care nu sunt achizitionate de niciun utilizator . . . .	29

# 1 Tema Proiectului

## 1.1 Introducere

Proiectul va fi despre un service de streaming pentru seriale si filme online, similar cu platforma Netflix.

## 1.2 Entitati si atribute

### i) Utilizatori

- a. Utilizator\_ID
- b. Abonament\_ID
- c. Prenume
- d. Nume
- e. Email
- f. Parola

### ii) Abonamente

- a. Abonament\_ID
- b. Tip\_Abonament
- c. Pret
- d. Durata
- e. Reducere

### iii) Metode\_de\_plata

- a. Plata\_ID
- b. Utilizator\_ID
- c. Tip\_Plata
- d. Numar\_Card
- e. Data\_Expirare
- f. Cod\_CVV

iv) Profiluri

- a. Profil\_ID
- b. Utilizator\_ID
- c. Nume\_Profil
- d. Categorie\_varsta

v) Filme

- a. Film\_ID
- b. Titlu
- c. Data\_Lansare
- d. Minute
- e. Descriere

vi) Seriale

- a. Serial\_ID
- b. Titlu
- c. Data\_Lansare
- d. Numar\_Sezoane
- e. Numar\_Episoade
- f. Descriere

vii) Caracter

- a. Caracter\_ID
- b. Actor\_ID
- c. Film\_ID
- d. Serial\_ID
- e. Nume\_Caracter

viii) Istoric\_Vizionari

- a. Istoric\_Vizionari\_ID
- b. Utilizator\_ID
- c. Film\_ID
- d. Serial\_ID
- e. Data\_Vizionarii
- f. Durata

ix) Favorite

- a. Favorite\_ID
- b. Utilizator\_ID
- c. Film\_ID
- d. Serial\_ID

x) Recenzii

- a. Recenzie\_ID
- b. Utilizator\_ID
- c. Film\_ID
- d. Serial\_ID
- e. Evaluare
- f. Text\_Evaluare

xi) Genuri

- a. Gen\_ID
- b. Serial\_ID
- c. Film\_ID
- d. Nume\_Gen

xiii) Distributie

- a. Actor\_ID
- b. Nume

## 1.3 Relatii

In proiect vor exista atat relatii one-to-many, cat si many-to-many. Cateva exemple de relatii one-to-many ar fi una dintre Utilizatori si Abonamente, una dintre Filme si Genuri si una dintre Utilizatori si Profiluri. Iar pentru relatiile de tip many-to-many: relatia dintre Filme si Note, cea dintre Filme si Utilizatori prin entitatea Recenzii.

## 2 Prezentarea constrangerilor impuse asupra modelului

- Un utilizator poate avea un singur abonament.
- O metoda de plata ii poate apartine doar unui utilizator.
- Un utilizator poate avea mai multe profiluri.
- Un film sau serial poate avea mai multe distributii.
- Un istoric de vizionari poate avea mai multe filme si mai multe seriale.
- Un utilizator poate avea un singur istoric de vizionari.
- Mai multe filme si seriale apartin favoritelor.
- O recenzie poate fi ori pentru un film ori pentru un serial.
- Orice tabel ce contine coloanele Film\_ID si Serial\_ID poate atribui doar uneia dintre ele o valoare nemula.

### 3 Descrierea entitatilor

- Utilizator: Utilizator\_ID; o persoana inregistrata pe platforma
- Abonamente: Abonament\_ID; abonament pentru accesarea filmelor si seriialelor
- Metode\_de\_plata: Plata\_ID; metoda de plata pentru achizitionarea abonamentului
- Profiluri: Profil\_ID; profilurile create de utilizator
- Filme: Film\_ID; film difuzat prin intermediul platformei
- Seriale: Serial\_ID; serial difuzat prin intermediul platformei
- Distributie: Actor\_ID; actor ce a luat parte la un film sau serial
- Istoric\_Vizionari: Istoric\_Vizionari\_ID; un film sau serial vizionat de un utilizator
- Favorite: Favorite\_ID; un film/serial favorit al unui utilizator
- Recenzii: Recenzie\_ID; o recenzie a unui utilizator pentru un film/serial
- Genuri: Gen\_ID; gen al unui film/serial
- Caracter: Caracter\_ID; caracter jucat de un actor intr-un film sau serial.



## 4 Descrierea relatiilor

- Relatia "a cumparat" Utilizator one-to-one Abonamente; Un utilizator cumpara si utilizeaza un singur abonament.
- Relatia "sunt folosite" Profiluri many-to-one Utilizator; Un utilizator foloseste mai multe profiluri in scopul vizionarii de continut;
- Relatia "este detinuta" Metode\_de\_plata one-to-one Utilizator; O metoda de plata poate fi detinuta doar de un utilizator
- Relatia "este de genul" Filme/Seriale one-to-many Genuri; Un film sau serial este de mai multe genuri (de exemplu comedie, drama, thriller etc.)
- Relatia "apare in" Caracter many-to-one Filme/Seriale; Mai multe caractere pot aparea intr-un film/serial
- Relatia "a fost vizionat de" Istoric\_Vizionari many-to-one Utilizator; Un utilizator poate avea mai multe vizionari in istoric
- Relatia "contine" Istoric\_Vizionari many-to-one Filme/Seriale; Mai multe istoricuri pot avea acelasi film/serial
- Relatia "este favorizat" Favorite many-to-one Utilizator; Un utilizator poate avea mai multe favorite
- Relatia "contine" Favorite one-to-one Filme/Seriale; O favorita poate contine un singur serial sau film
- Relatia "a recenzat" Recenzii many-to-one Utilizator; Un utilizator poate avea mai multe recenzii
- Relatia "este despre" Recenzii many-to-one Filme/Seriale; Mai multe recenzii pot fi despre un serial sau un film
- Relatia "joaca" Distributie one-to-one Caracter; Un actor poate juca un caracter.

## 5 Descrierea atributelor

- Entitate: Utilizator

Tabelul 5.1: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Utilizator_ID	INT		
Abonament_ID	INT		
Prenume	VARCHAR	100	
Nume	VARCHAR	50	
Email	VARCHAR	50	
Parola	VARCHAR	50	

- Entitate: Abonamente

Tabelul 5.2: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Tip_Abonament	VARCHAR	50	Premium
Abonament_ID	INT		
Pret	INT		
Durata	INT		120 (minute)
Reducere	INT		

- Entitate: Metode\_de\_plata

Tabelul 5.3: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Plata_ID	INT		
Utilizator_ID	INT		
Tip_Plata	VARCHAR	20	
Numar_Card	VARCHAR	50	
Data_Expirare	DATE		
Cod_CVV	VARCHAR	20	

- Entitate: Profiluri

Tabelul 5.4: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Profil_ID	INT		
Utilizator_ID	INT		
Nume_Profil	VARCHAR	50	

Tabelul 5.5: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Film_ID	INT		
Titlu	VARCHAR	100	
Minute	INT		
Descriere	VARCHAR	300	
Data_Lansare	DATE		

- Entitate: Filme
- Entitate: Seriale

Tabelul 5.6: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Serial_ID	INT		
Titlu	VARCHAR	100	
Numar_Sezoane	INT		
Numar_Episoade	INT		
Descriere	VARCHAR	300	
Data_Lansare	DATE		

- Entitate: Distributie

Tabelul 5.7: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Actor_ID	INT		
Nume	VARCHAR	100	

- Entitate: Caracter

Tabelul 5.8: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Caracter_ID	INT		
Film_ID	INT		
Serial_ID	INT		
Actor_ID	INT		
Nume_Caracter	VARCHAR	100	

- Entitate: Istoric\_Vizionari
- Entitate: Favorite
- Entitate: Recenzii
- Entitate: Genuri

Tabelul 5.9: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Istoric_Vizionari_ID	INT		
Film_ID	INT		
Serial_ID	INT		
Utilizator_ID	INT		
Data_Vizionarii	DATE		

Tabelul 5.10: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Favorite_ID	INT		
Film_ID	INT		
Serial_ID	INT		
Utilizator_ID	INT		

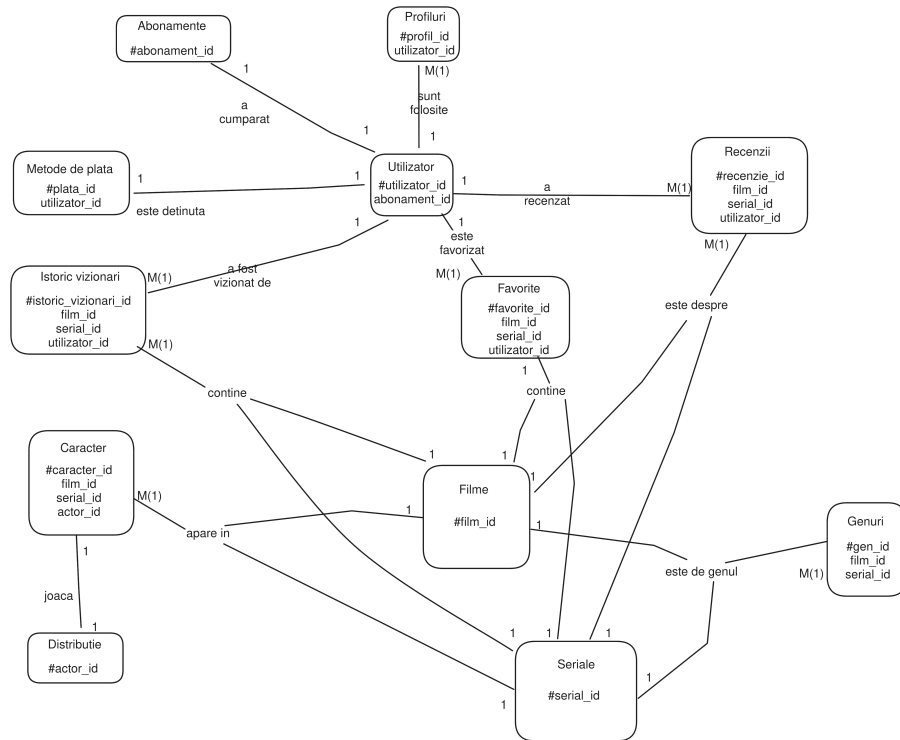
Tabelul 5.11: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Recenzie_ID	INT		
Film_ID	INT		
Serial_ID	INT		
Utilizator_ID	INT		
Evaluare	INT		1, 2, 3, 4, 5
Text_Evaluare	VARCHAR	300	

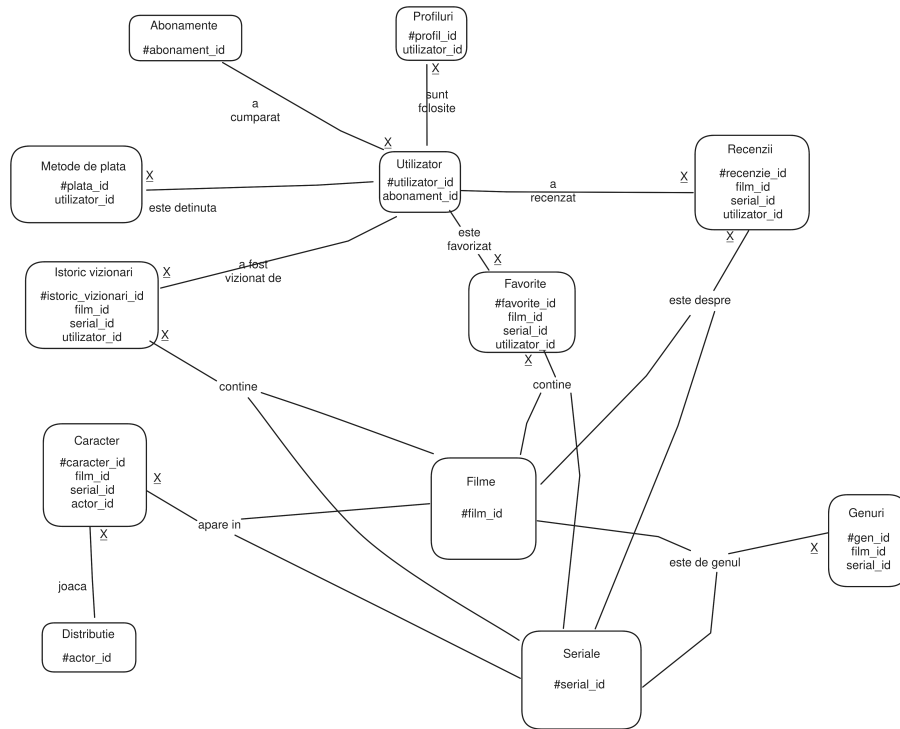
Tabelul 5.12: Atribut

Atribut	Tip	Dimensiune	Valori posibile
Gen_ID	INT		
Film_ID	INT		
Serial_ID	INT		
Nume_Gen	VARCHAR	30	

## 6 Diagrama entitate-relatie



## 7 Diagrama conceptuala



## 8 Enumerarea schemelor relationale corespunzatoare diagramei conceptuale

- Utilizator(#utilizator\_id, abonament\_id, prenume, nume, email, parola)
- Abonamente(#abonament\_id, tip\_abonament, reducere, durata, pret)
- Metode\_de\_plata(#plata\_id, utilizator\_id, tip\_plata, numar\_card, data\_expirare, cod\_cvv)
- Profiluri(#profil\_id, utilizator\_id, nume\_profil)
- Filme(#film\_id, titlu, minute, descriere, data\_lansare)
- Seriale(#serial\_id, titlu, numar\_sezoane, numar\_episoade, descriere, data\_lansare)
- Distributie(#actor\_id, nume)
- Caracter(#caracter\_id, film\_id, serial\_id, actor\_id, nume\_caracter)
- Istoric\_Vizionari(#istoric\_vizionari\_id, film\_id, serial\_id, utilizator\_id, data\_vizionarii)
- Favorite(#favorite\_id, film\_id, serial\_id, utilizator\_id)
- Recenzii(#recenzie\_id, film\_id, serial\_id, utilizator\_id, evaluare, text\_evaluare)
- Genuri(#gen\_id, film\_id, serial\_id, nume\_gen)

## 9 Realizarea normalizarii pana la forma normala 3

### 9.1 Forma normala 1 (FN1)

Forma normala 1 (FN1) inseamna ca fiecare atribut dintr-o relatie are o valoare atomica si fiecare inregistrare sa fie identificata in mod unic printr-o cheie primara.

Initial, tabelurile Seriale si Filme aveau coloana numita "Genuri". In acest caz, problema apare cand serialul/filmul apartine de mai multe genuri, cum ar fi comedie si drama, si ar incalca forma normala 1, deoarece coloanei "Genuri" nu i-ar fi corespuns o valoare atomica.

Pentru a rezolva aceasta problema am creat un nou tabel, numit "Genuri" cu coloanele (Gen\_ID, Film\_ID, Serial\_ID, Nume\_Gen). Astfel, se creeaza o relatie one-to-many intre tabelurile Filme/Seriale si tabelul Genuri.

### 9.2 Forma normala 2 (FN2)

Forma normala 2 (FN2) inseamna ca o relatie se afla in Forma normala 1 (FN1) si fiecare atribut care nu este cheie primara este dependent de intreaga cheie primara.

Intr-o forma non-FN2, tabelul Distributie avea schema conceptuala Distributie(#actor\_id, film\_id, serial\_id, nume, nume\_caracter). Se observa o dependenta directa intre actor\_id si film\_id/serial\_id, insa o dependenta partiala pentru atributul nume\_caracter, deoarece acesta depinde si de film\_id/serial\_id, dar si de actor\_id.

Solutie: Am creat un nou tabel, Caracter, cu schema conceptuala Caracter(#caracter\_id, nume\_caracter, film\_id, serial\_id, actor\_id), iar tabelul Distributie este modificat astfel incat schema conceptuala a sa devine Distributie(#actor\_id, nume).

### 9.3 Forma normala 3 (FN3)

Forma normala 3 (FN3) inseamna ca o relatie se afla in Forma normala 2 (FN2) si fiecare atribut care nu este cheie, depinde direct de cheia primara.

Initial, tabelurile Filme si Seriale aveau doua coloane in plus, numite Nume\_Nota si Text\_Evaluare. Coloana Nume\_Nota depindea direct de cheia primara corespunzatoare tabelului (#film\_id, respectiv #serial\_id), insa coloana Text\_Evaluare depindea doar de nume\_nota, nu si de cheia primara a tabelului.

Astfel, am creat tabelul Note cu schema conceptuala Note(#nota\_id, film\_id, serial\_id, nume\_nota, text\_evaluare), iar din schemele conceptuale ale tabelurilor Filme, respectiv Seriale vor fi eliminate coloanele Nume\_Nota si Text\_Evaluare.



## 10 Crearea de tabele SQL si inserarea de date coerente in acestea

### 10.1 Abonamente

```
CREATE TABLE ABONAMENTE ( Abonament_ID INT CONSTRAINT PKEY_ABONAMENT  
PRIMARY KEY, Tip_Abonament VARCHAR(50) CONSTRAINT Tip_Abonament NOT NULL,  
Pret INT CONSTRAINT Pret_Conducere NOT NULL, Durata INT CONSTRAINT Durata_Abonament  
NOT NULL, Reducere INT );
```

```
INSERT INTO ABONAMENTE VALUES (1,'Gratis',0,0,0);
```

```
INSERT INTO ABONAMENTE VALUES (2,'Standard',25,30,0);
```

```
INSERT INTO ABONAMENTE VALUES (3,'Premium',50,30,0);
```

```
INSERT INTO ABONAMENTE VALUES (4,'Familie',75,30,5);
```

```
INSERT INTO ABONAMENTE VALUES (5,'Familie4K',100,30,15);
```

```
SELECT * FROM ABONAMENTE;
```

	ABONAMENT_ID	TIP_ABONAMENT	PRET	DURATA	REDUCERE
1	1	Gratis	0	0	0
2	2	Standard	25	30	0
3	3	Premium	50	30	0
4	4	Familie	75	30	5
5	5	Familie4K	100	30	15

### 10.2 Utilizator

```
CREATE TABLE UTILIZATOR ( Utilizator_ID INT CONSTRAINT PKEY_ABONAMENT  
PRIMARY KEY, Prenume VARCHAR(100) CONSTRAINT Prenume NOT NULL, Nume VAR-  
CHAR(50) CONSTRAINT Nume NOT NULL, Email VARCHAR(50) CONSTRAINT Email  
UNIQUE, Parola VARCHAR(50) CONSTRAINT Parola NOT NULL, Abonament_ID INT,  
CONSTRAINT FK_ABONAMENT FOREIGN KEY(Abonament_ID) REFERENCES ABO-  
NAMENTE(Abonament_ID) );
```

```
INSERT INTO UTILIZATOR  
VALUES (1,'Ion','Badea','ion.badea@gmail.com','ionBadEa14',1);
```

```
INSERT INTO UTILIZATOR  
VALUES (2,'Octavian','Baltag','octavian.baltag@gmail.com','oktavian.baltg@gmail.com',3);
```

```
INSERT INTO UTILIZATOR  
VALUES (3,'Paul','Bechet','bechet.paul@gmail.com','bechetpaulll9098',4);
```

```
INSERT INTO UTILIZATOR
VALUES (4,'Catalin','Boltasu','boltasu.catalin@gmail.com','catalinn7',2);
```

```
INSERT INTO UTILIZATOR
VALUES (5,'Mihai','Cernat','cernat.mihai@gmail.com','mihaiii9',5);
```

```
SELECT * FROM UTILIZATOR;
```

	UTILIZATOR_ID	PRENUME	NUME	EMAIL	PAROLA	ABONAMENT_ID
1	1	Ion	Badea	ion.badea@gmail.com	ionBadEa14	1
2	2	Octavian	Baltag	octavian.baltag@gmail.com	oktavian.baltg@gmail.com	3
3	3	Paul	Bechet	bechet.paul@gmail.com	bechetpaul119098	4
4	4	Catalin	Boltasu	boltasu.catalin@gmail.com	catalinn7	2
5	5	Mihai	Cernat	cernat.mihai@gmail.com	mihaiii9	5

### 10.3 Metode\_de\_plata

```
CREATE TABLE METODE_DE_PLATA ( Plata_ID INT CONSTRAINT PKEY_PLATA
PRIMARY KEY, Tip_Plata VARCHAR(20) CONSTRAINT Tip_Plata NOT NULL, Nu-
mar_Card VARCHAR(50) CONSTRAINT Numar_Card NOT NULL, Cod_CVV VARCHAR(20)
CONSTRAINT Cod_CVV UNIQUE, Data_Expirare DATE CONSTRAINT Data_Expirare
NOT NULL, Utilizator_ID INT, CONSTRAINT FK_UTILIZATOR FOREIGN KEY(Utilizator_ID)
REFERENCES Utilizator(Utilizator_ID) );
```

```
INSERT INTO METODE_DE_PLATA VALUES
(1,'mastercard','4867-0217-4985-2000','505',TO_DATE('november-2026', 'mm-yy'),1);
```

```
INSERT INTO METODE_DE_PLATA VALUES
(2,'visa','4490-5670-6378-5650','614',TO_DATE('april-2024', 'mm-yy'),2);
```

```
INSERT INTO METODE_DE_PLATA VALUES
(3,'visa','4318-4950-0056-4188','302',TO_DATE('january-2025', 'mm-yy'),3);
```

```
INSERT INTO METODE_DE_PLATA VALUES
(4,'mastercard','4726-0753-1627-2159','427',TO_DATE('june-2024', 'mm-yy'),4);
```

```
INSERT INTO METODE_DE_PLATA VALUES
(5,'visa','4134-7202-7223-2335','117',TO_DATE('august-27', 'mm-yy'),5);
```

```
SELECT * FROM METODE_DE_PLATA;
```

	PLATA_ID	TIP_PLATA	NUMAR_CARD	COD_CVV	DATA_EXPIRARE	UTILIZATOR_ID
1	5	visa	4134-7202-7223-2335	117	2027-08-01	5
2	1	mastercard	4867-0217-4985-2000	505	2026-11-01	1
3	2	visa	4490-5670-6378-5650	614	2024-04-01	2
4	3	visa	4318-4950-0056-4188	302	2025-01-01	3
5	4	mastercard	4726-0753-1627-2159	427	2024-06-01	4

### 10.4 Profiluri

```
CREATE TABLE PROFILURI ( Profil_ID INT CONSTRAINT PKEY_Profil PRIMARY
KEY, Nume_Profil VARCHAR(50) CONSTRAINT Nume_Profil NOT NULL, Utilizator_ID
INT, CONSTRAINT FK_UTILIZATOR_PROFIL FOREIGN KEY(Utilizator_ID) REFE-
RENCES Utilizator(Utilizator_ID) );
```

```
INSERT INTO PROFILURI VALUES (1,'Ion',1);
```

```
INSERT INTO PROFILURI VALUES (2,'Octavian',2);
```

```
INSERT INTO PROFILURI VALUES (3,'Paul',3);
```

```
INSERT INTO PROFILURI VALUES (4,'Catalin',4);
```

```
INSERT INTO PROFILURI VALUES (5,'Mihai',5);
```

```
SELECT * FROM PROFILURI;
```

	PROFIL_ID	NUME_PROFIL	UTILIZATOR_ID
1	1	Ion	1
2	2	Octavian	2
3	3	Paul	3
4	4	Catalin	4
5	5	Mihai	5

## 10.5 Filme

```
CREATE TABLE FILME ( Film_ID INT CONSTRAINT PKEY_Film PRIMARY KEY, Titlu VARCHAR(100) CONSTRAINT Titlu_Film NOT NULL, Minute INT CONSTRAINT Minute_Film NOT NULL, Descriere VARCHAR(300) CONSTRAINT Descriere_Film NOT NULL, Data_Lansare DATE CONSTRAINT Lansare_Film NOT NULL );
```

```
INSERT INTO FILME VALUES (1,'Se7en',127,'Doi detectivi, un începător și un veteran, vânează un ucigaș în serie care se folosește de cele șapte păcate de moarte ca motivații',TO_DATE('15-september-1995','dd-month-yyyy'));
```

```
INSERT INTO FILME VALUES (2,'The Prestige',130,'După un accident tragic, doi magicieni de scenă din Londra anilor 1890 se angajează într-o luptă pentru a crea iluzia supremă, sacrificând tot ce au pentru a se păcăli unul pe celălalt',TO_DATE('17-october-2006','dd-month-yyyy'));
```

```
INSERT INTO FILME VALUES (3,'Shutter Island',138,'Teddy Daniels și Chuck Aule, doi șerifi americani, sunt trimiși la un azil de pe o insulă îndepărtată pentru a investiga dispariția unui pacient, unde Teddy descoperă un adevăr șocant despre acest loc',TO_DATE('13-february-2010','dd-month-yyyy'));
```

```
INSERT INTO FILME VALUES (4,'The Green Mile',189,'O poveste care se petrece în închisoarea din sudul țării, unde uriașul blând John are puterea misterioasă de a vindeca bolile oamenilor. Când gardianul principal, Paul, recunoaște darul lui John, încearcă să ajute la evitarea execuției condamnatului',TO_DATE('6-december-1999','dd-month-yyyy'));
```

```
INSERT INTO FILME VALUES (5,'Memento',113,'Un bărbat cu pierderi de memorie pe termen scurt încearcă să dea de urma ucigașului soției sale',TO_DATE('5-september-2000','dd-month-yyyy'));
```

```
INSERT INTO FILME VALUES (0,'Niciun Film',0,'N',TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
SELECT * FROM FILME;
```

FILM_ID	TITLU	MINUTE	DESCRIERE	DATA_LANSARE
1	5 Memento	113	Un bărbat cu pierderi de memorie pe termen scurt încearcă să dea...	2000-09-05
2	1 Se7en	127	Doi detectivi, un începător și un veteran, vânează un ucigaș în ...	1995-09-15
3	2 The Prestige	130	După un accident tragic, doi magicieni de scenă din Londra anilo...	2006-10-17
4	3 Shutter Island	138	Teddy Daniels și Chuck Aule, doi șerifi americani, sunt trimiși ...	2010-02-13
5	4 The Green Mile	189	0 poveste care se petrece în închisoarea din sudul țării, unde u...	1999-12-06
6	0 Niciun Film	0 N/A		2023-07-07

## 10.6 Seriale

```
CREATE TABLE SERIALE (
  Seriale_ID INT CONSTRAINT PKEY_Serial PRIMARY KEY,
  Titlu VARCHAR(100) CONSTRAINT Titlu_Serial NOT NULL,
  Numar_Sezoane INT CONSTRAINT Numar_Sezoane NOT NULL,
  Numar_Episoade INT CONSTRAINT Numar_Episoade NOT NULL,
  Descriere VARCHAR(300) CONSTRAINT Descriere_Serial NOT NULL,
  Data_Lansare DATE CONSTRAINT Lansare_Serial NOT NULL );
```

INSERT INTO SERIALE VALUES (1,'Square Mile of Murder',1,6,'Acest serial dramatic în șase părți s-a bazat pe cartea cu același nume scrisă de Jack House. Descriind crime reale care au avut loc în zona Charing Cross din Glasgow. Fiecare episod prezintă o crimă misterioasă diferită',TO\_DATE('12-june-1980','dd-month-yyyy'));

INSERT INTO SERIALE VALUES (2,'Black Mirror',6,27,'O serie antologică care explorează un multivers întortocheat de înaltă tehnologie în care cele mai mari inovații și cele mai întunecate instincte ale umanității se ciocnesc',TO\_DATE('4-december-2011','dd-month-yyyy'));

INSERT INTO SERIALE VALUES (3,'Manifest',4,62,'Atunci când un avion de linie reappare brusc după ce a dispărut timp de cinci ani cei de la bord trebuie să se reintegreze în societate',TO\_DATE('23-september-2018','dd-month-yyyy'));

INSERT INTO SERIALE VALUES (4,'The Blacklist',10,218,'Un nou profiler FBI, Elizabeth Keen își vede întreaga viață bulversată atunci când un criminal misterios Raymond Reddington care a scăpat de capturare timp de decenii se predă și insistă să vorbească doar cu ea',TO\_DATE('23-september-2013','dd-month-yyyy'));

INSERT INTO SERIALE VALUES (5,'True Detective',4,25,'Serie antologică sezonieră în care investigațiile polițienști scot la iveală secretele personale și profesionale ale celor implicați, atât în interiorul, cât și în afara legii',TO\_DATE('12-january-2014','dd-month-yyyy'));

INSERT INTO SERIALE VALUES (0,'Niciun Serial',0,0,'N',TO\_DATE('7-july-2023','dd-month-yyyy'));

```
SELECT * FROM SERIALE;
```

SERIALE_ID	TITLU	NUMAR_SEZOANE	NUMAR_EPISOADE	DESCRIERE	DATA_LANSARE
1	5 True Detective	4	25	Serie antologică sezonieră în care investigațiile polițienști s...	2014-01-12
2	1 Square Mile of Murder	1	6	Acest serial dramatic în șase părți s-a bazat pe cartea cu acela...	1980-06-12
3	2 Black Mirror	6	27	O serie antologică care explorează un multivers întortocheat de ...	2011-12-04
4	3 Manifest	4	62	Atunci când un avion de linie reappare brusc după ce a dispărut t...	2018-09-23
5	4 The Blacklist	10	218	Un nou profiler FBI, Elizabeth Keen își vede întreaga viață bulv...	2013-09-23
6	0 Niciun Serial	0	0	N/A	2023-07-07

## 10.7 Distributie

```
CREATE TABLE DISTRIBUTIE (
  Actor_ID INT CONSTRAINT PKEY_Actor PRIMARY KEY,
  Nume VARCHAR(100) CONSTRAINT Actor NOT NULL );
```

```

INSERT INTO DISTRIBUTIE VALUES (1,'Guy Pearce');
INSERT INTO DISTRIBUTIE VALUES (2,'Brad Pitt');
INSERT INTO DISTRIBUTIE VALUES (3,'Christian Bale');
INSERT INTO DISTRIBUTIE VALUES (4,'Leonardo DiCaprio');
INSERT INTO DISTRIBUTIE VALUES (5,'Tom Hanks');
INSERT INTO DISTRIBUTIE VALUES (6,'Mahershala Ali');
INSERT INTO DISTRIBUTIE VALUES (7,'George Baker');
INSERT INTO DISTRIBUTIE VALUES (8,'Hayley Atwell');
INSERT INTO DISTRIBUTIE VALUES (9,'Melissa Roxburgh');
INSERT INTO DISTRIBUTIE VALUES (10,'James Spader');

SELECT * FROM DISTRIBUTIE;

```

	ACTOR_ID	NUME
1	1	Guy Pearce
2	2	Brad Pitt
3	3	Christian Bale
4	4	Leonardo DiCaprio
5	5	Tom Hanks
6	6	Mahershala Ali
7	7	George Baker
8	8	Hayley Atwell
9	9	Melissa Roxburgh
10	10	James Spader

## 10.8 Istoric\_Vizionari

```

CREATE TABLE ISTORIC_VIZIONARI ( Istorice_Vizionari_ID INT CONSTRAINT PKEY_Istoric
PRIMARY KEY, Film_ID INT, CONSTRAINT FK_Istoric_Film FOREIGN KEY(Film_ID)
REFERENCES FILME(Film_ID), Serial_ID INT, CONSTRAINT FK_Istoric_Serial FOREIGN
KEY(Serial_ID) REFERENCES SERIALE(Seriale_ID), Utilizator_ID INT, CONSTRAINT
FK_Istoric_Utilizator FOREIGN KEY(Utilizator_ID) REFERENCES Utilizator(Utilizator_ID),
Data_Vizioarii DATE CONSTRAINT Istoric_Data NOT NULL );

```

```

INSERT INTO ISTORIC_VIZIONARI VALUES (1,1,0,1,TO_DATE('7-july-2023','dd-
month-yyyy'));

```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (2,2,0,2,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (3,3,0,3,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (4,4,0,4,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (5,5,0,5,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (6,0,1,1,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (7,0,2,2,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (8,0,3,3,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (9,0,4,4,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
INSERT INTO ISTORIC_VIZIONARI VALUES (10,0,5,5,TO_DATE('7-july-2023','dd-month-yyyy'));
```

```
SELECT * FROM ISTORIC_VIZIONARI;
```

	ISTORIC_VIZIONARI_ID	FILM_ID	SERIAL_ID	UTILIZATOR_ID	DATA_VIZIIONARI
1	1	1	0	1	2023-07-07
2	2	2	0	2	2023-07-07
3	3	3	0	3	2023-07-07
4	4	4	0	4	2023-07-07
5	5	5	0	5	2023-07-07
6	6	0	1	1	2023-07-07
7	7	0	2	2	2023-07-07
8	8	0	3	3	2023-07-07
9	9	0	4	4	2023-07-07
10	10	0	5	5	2023-07-07

## 10.9 Favorite

```
CREATE TABLE FAVORITE ( Favorite_ID INT CONSTRAINT PKEY_Favorite PRIMARY KEY, Film_ID INT, CONSTRAINT FK_Favorite_Film FOREIGN KEY(Film_ID) REFERENCES FILME(Film_ID), Serial_ID INT, CONSTRAINT FK_Favorite_Serial FOREIGN KEY(Serial_ID) REFERENCES SERIALE(Seriale_ID), Utilizator_ID INT, CONSTRAINT FK_Favorite_Utilizator FOREIGN KEY(Utilizator_ID) REFERENCES Utilizator(Utilizator_ID) );
```

```
INSERT INTO FAVORITE VALUES (1,1,0,1);
```

```
INSERT INTO FAVORITE VALUES (2,2,0,2);
```

```
INSERT INTO FAVORITE VALUES (3,3,0,3);
```

```
INSERT INTO FAVORITE VALUES (4,4,0,4);
```

```

INSERT INTO FAVORITE VALUES (5,5,0,5);

INSERT INTO FAVORITE VALUES (6,0,1,1);

INSERT INTO FAVORITE VALUES (7,0,2,2);

INSERT INTO FAVORITE VALUES (8,0,3,3);

INSERT INTO FAVORITE VALUES (9,0,4,4);

INSERT INTO FAVORITE VALUES (10,0,5,5);

SELECT * FROM FAVORITE;

```

	FAVORITE_ID	FILM_ID	SERIAL_ID	UTILIZATOR_ID
1	1	1	0	1
2	2	2	0	2
3	3	3	0	3
4	4	4	0	4
5	5	5	0	5
6	6	0	1	1
7	7	0	2	2
8	8	0	3	3
9	9	0	4	4
10	10	0	5	5

## 10.10 Genuri

```

CREATE TABLE GENURI ( Gen_ID INT CONSTRAINT PKEY_Gen PRIMARY KEY,
Film_ID INT, CONSTRAINT FK_GenFilm FOREIGN KEY (Film_ID) REFERENCES FILME (Film_ID));

INSERT INTO GENURI VALUES (1,1,0,'Action');

INSERT INTO GENURI VALUES (2,2,0,'Adventure');

INSERT INTO GENURI VALUES (3,3,0,'Action');

INSERT INTO GENURI VALUES (4,4,0,'Mystery');

INSERT INTO GENURI VALUES (5,5,0,'Horror');

INSERT INTO GENURI VALUES (6,0,1,'Thriller');

INSERT INTO GENURI VALUES (7,0,2,'Comedy');

INSERT INTO GENURI VALUES (8,0,3,'Horror');

INSERT INTO GENURI VALUES (9,0,4,'Action');

INSERT INTO GENURI VALUES (10,0,5,'Adventure');

SELECT * FROM GENURI;

```

	GEN_ID	FILM_ID	SERIAL_ID	NUME_GEN
1	1	1	0	Action
2	2	2	0	Adventure
3	3	3	0	Action
4	4	4	0	Mystery
5	5	5	0	Horror
6	6	0	1	Thriller
7	7	0	2	Comedy
8	8	0	3	Horror
9	9	0	4	Action
10	10	0	5	Adventure

## 10.11 Recenzii

```
CREATE TABLE RECENZII ( Recenzie_ID INT CONSTRAINT PKEY_Recenzie PRIMARY
KEY, Film_ID INT, CONSTRAINT FK_Recenzie_Film FOREIGN KEY(Film_ID) REFE-
RENCES FILME(Film_ID), Serial_ID INT, CONSTRAINT FK_Recenzie_Serial FOREIGN
KEY(Serial_ID) REFERENCES SERIALE(Seriale_ID), Utilizator_ID INT, CONSTRAINT
FK_Recenzie_Utilizator FOREIGN KEY(Utilizator_ID) REFERENCES Utilizator(Utilizator_ID),
Evaluare INT CONSTRAINT Nota_Evaluare NOT NULL, Text_Evaluare VARCHAR(300)
CONSTRAINT Text_Evaluare NOT NULL );
```

INSERT INTO RECENZII VALUES

(1,1,0,1,4,'Poate că toate trupele fantastice de aici sunt generice, dar ei știu asta și poartă un zâmbet de satisfacție');

INSERT INTO RECENZII VALUES

(2,2,0,2,5,'Cavill ar putea pleca într-o notă relativ bună, datorită unui al treilea sezon mai strâns și mai concentrat, cel puțin inițial, decât a oferit cel de-al doilea');

INSERT INTO RECENZII VALUES

(3,3,0,3,3,'Cu intrigi care încep în sfârșit să se coaguleze, acesta este cel mai strâns și mai plin de acțiune sezon din The Witcher de până acum');

INSERT INTO RECENZII VALUES

(4,4,0,4,2,'O continuare mediocră a unei francize care își pierde rapid farmecul');

INSERT INTO RECENZII VALUES

(5,5,0,5,5,'S-a terminat. Totul este foarte rău. Începând de la selecția noilor actori și terminând cu scenariul');

INSERT INTO RECENZII VALUES

(6,0,1,1,1,'Depart de cărți');

INSERT INTO RECENZII VALUES

(7,0,2,2,4,'Poate mai ambițios, dar și mai puțin reușit, la care se adaugă plecarea lui Cavill');

INSERT INTO RECENZII VALUES

(8,0,3,3,5,'Din păcate, scenariștii petrec mult prea mult timp dezvoltând scenariul și tind să se bazeze prea des pe răsturnări de situație stupide și pe tropi de spionaj exagerat.');

INSERT INTO RECENZII VALUES

(9,0,5,5,5,'Cvartetul de actrițe experimentate își aduc jocul lor de nivel A pentru a face ca acest film de împlinire a dorințelor, film de bun-simț, să fie foarte distractiv de urmărit.');



INSERT INTO RECENZII VALUES

(10,0,4,4,4,'Nu-mi vine să cred că acest film există... M-am simțit foarte jenată în mare parte din timp pentru aceste femei');

SELECT \* FROM RECENZII;

#	RECENZIE_ID	FILM_ID	SERIAL_ID	UTILIZATOR_ID	EVALUARE	TEXT_EVALUARE
1	1	1	0	1	4	Poate că toate trupele fantastice de aici sunt generice, dar ei...
2	2	2	0	2	5	Cavill ar putea pleca într-o notă relativ bună, datorită unui al...
3	3	3	0	3	3	Cu intrigi care încep în sfârșit să se coaguleze, acesta este c...
4	4	4	0	4	2	0 continuare mediocră a unei francize care își pierde rapid farm...
5	5	5	0	5	5	S-a terminat. Totul este foarte rău. Începând de la selecția noi...
6	6	0	1	1	1	Departee de cârți
7	7	0	2	2	4	Poste mai ambițios, dar și mai puțin reușit, la care se adaugă...
8	8	0	3	3	5	Din păcate, scenariștii petrec mult prea mult timp dezvoltând sc...
9	9	0	5	5	5	Cvartetul de actrițe experimentate își aduc jocul lor de nivel A...
10	10	0	4	4	4	Nu-mi vine să cred că acest film există... M-am simțit foarte je...

## 11 Crearea unei secvente ce va fi utilizata in inserarea inregistrarilor in tabele

```
SELECT * FROM PROFILURI;
```

```
CREATE SEQUENCE ADD1 START WITH 6 INCREMENT BY 1 MAXVALUE 50  
NOCYCLE;
```

```
CREATE SEQUENCE ADD2 START WITH 1 INCREMENT BY 1 MAXVALUE 5 NO-  
CYCLE;
```

```
INSERT INTO PROFILURI VALUES (ADD1.NEXTVAL,'Andrei', ADD2.NEXTVAL);
```

```
SELECT * FROM PROFILURI;
```

	PROFIL_ID	NUME_PROFIL	UTILIZATOR_ID
1	1	Ion	1
2	2	Octavian	2
3	3	Paul	3
4	4	Catalin	4
5	5	Mihai	5

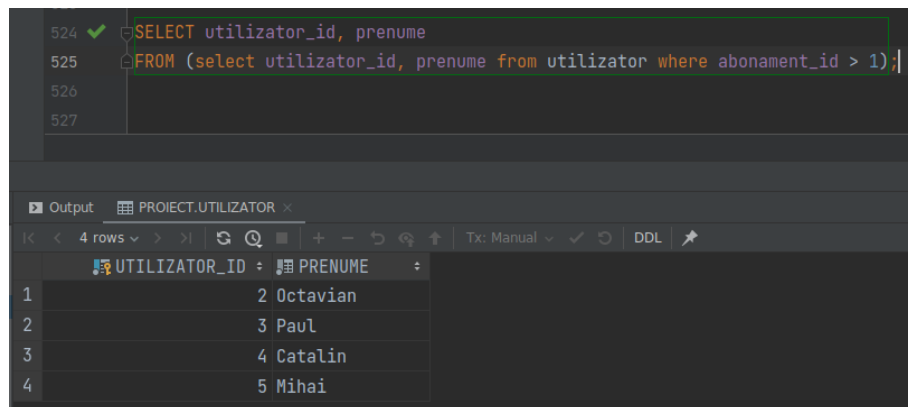
	PROFIL_ID	NUME_PROFIL	UTILIZATOR_ID
1	1	Ion	1
2	2	Octavian	2
3	3	Paul	3
4	4	Catalin	4
5	5	Mihai	5
6	6	Andrei	1

## 12 Formulati in limbja natural si implementati 5 cereri SQL complexe

### 12.1 Sa se afiseze utilizatorii care nu au abonamentul gratuit.

- subcereri nesincronizate in clauza FROM

```
SELECT utilizator_id, prenume  
FROM (select utilizator_id, prenume from utilizator where abonament_id > 1);
```



### 12.2 Sa se afiseze utilizatorii ce au vizionat un serial thriller in luna iulie

- 2 functii pe siruri de caractere, 1 functie pe date calendaristice, subcereri sincronizate in care intervin cel putin 3 tabele

```
select u.utilizator_id, upper(concat(u.prenume, concat(' ', u.num)))  
from utilizator u  
where u.utilizator_id in  
(select iv.utilizator_id  
from istoric_vizionari iv  
where extract(month from iv.data_vizioarii) = 7  
and u.utilizator_id = iv.utilizator_id  
and iv.serial_id in  
(select s.seriale_id  
from seriale s  
where s.seriale_id = iv.serial_id  
and s.seriale_id in (select g.serial_id  
from genuri g  
where g.serial_id = s.seriale_id  
and g.num_gen = 'Thriller')));
```

```

538 ✓ select u.utilizator_id, upper(concat(u.prenume, concat(' ', u.num)))
539       from utilizator u
540      where u.utilizator_id in
541             (select iv.utilizator_id
542              from istoric_vizionari iv
543              where extract(month from iv.data_viziorii) = 7
544                    and u.utilizator_id = iv.utilizator_id
545                    and iv.serial_id in
546                           (select s.serial_id
547                            from seriale s
548                            where s.serial_id = iv.serial_id
549                                  and s.serial_id in (select g.serial_id
550                                                       from genuri g
551                                                       where g.serial_id = s.serial_id
552                                                             and g.num_gen = 'Thriller')));
553

```

Output Result 3 x

UTILIZATOR\_ID : UPPER(CONCAT(U.PRENUME, CONCAT(' ', U.NUM)))

1	1 ION BADEA
---	-------------

### 12.3 Sa se afiseze pentru fiecare film data de lansare si daca este de actiune.

- ordonari si nvl si decode

```

select f.titlu,
f.data_lansare,
nvl(decode(g.num_gen, 'Action', 'Film de actiune', null), 'Film non-actiune') as Tip_Film
from filme f
join genuri g on g.film_id = f.film_id and f.film_id != 0;

```

```

558 ✓ select f.titlu,
559        f.data_lansare,
560        nvl(decode(g.num_gen, 'Action', 'Film de actiune', null), 'Film non-actiune') as Tip_Film
561       from filme f
562      join genuri g on g.film_id = f.film_id and f.film_id != 0
563     order by f.film_id;
564

```

Output ordonari si nvl si decode x

5 rows

TITLU	DATA_LANSARE	TIP_FILM
1 Se7en	1995-09-15	Film non-actiune
2 The Prestige	2006-10-17	Film non-actiune
3 Shutter Island	2010-02-13	Film non-actiune
4 The Green Mile	1999-12-06	Film non-actiune
5 Memento	2000-09-05	Film non-actiune

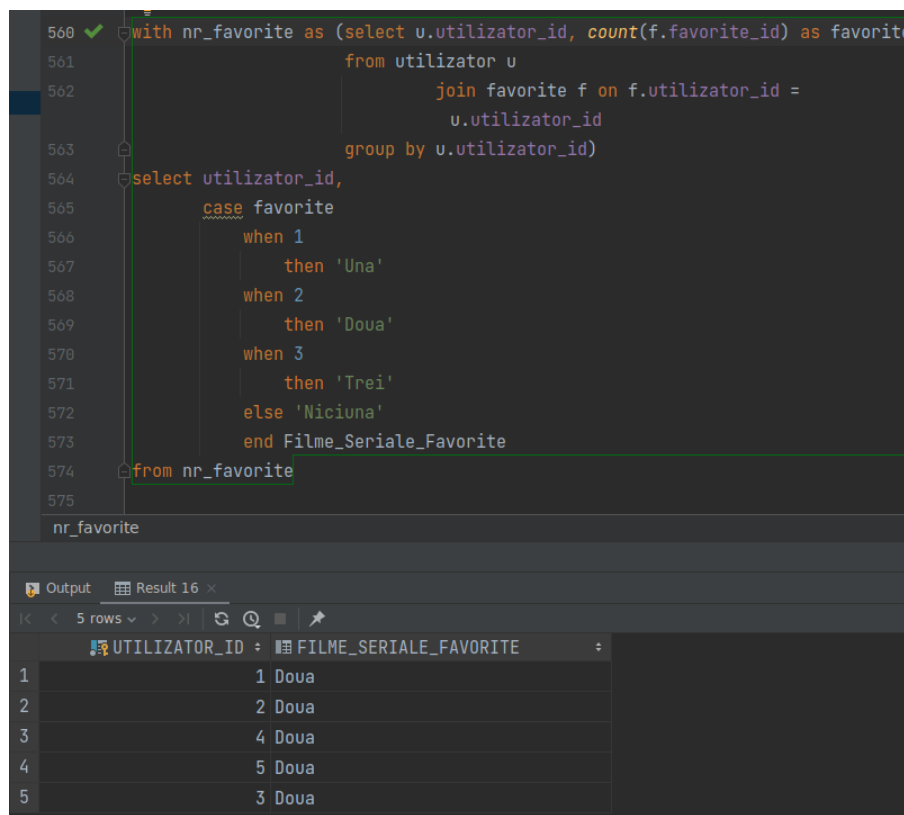
### 12.4 Sa se afiseze pentru fiecare utilizator numarul de seriale si numarul de filme favorite insumate. (Exp: Daca are 2 seriale favorite, atunci se afiseaza 'doua')

- clauza with si expresia case

```

with nr_favorite as (select u.utilizator_id, count(f.favorite_id) as favorite
from utilizator u
join favorite f on f.utilizator_id = u.utilizator_id
group by u.utilizator_id)
select utilizator_id,
case favorite
when 1
then 'Una'
when 2
then 'Doua'
when 3
then 'Trei'
else 'Niciuna'
end Filme_Seriale_Favorite
from nr_favorite

```



```

560 with nr_favorite as (select u.utilizator_id, count(f.favorite_id) as favorite
561                        from utilizator u
562                        join favorite f on f.utilizator_id =
563                        u.utilizator_id
564                        group by u.utilizator_id)
565 select utilizator_id,
566        case favorite
567        when 1
568        then 'Una'
569        when 2
570        then 'Doua'
571        when 3
572        then 'Trei'
573        else 'Niciuna'
574        end Filme_Seriale_Favorite
575 from nr_favorite

```

UTILIZATOR_ID	FILME_SERIALE_FAVORITE
1	Doua
2	Doua
3	Doua
4	Doua
5	3 Doua

## 12.5 Sa se afiseze numarul de utilizatori ce au vizionat vreun film lansat in anul 2000

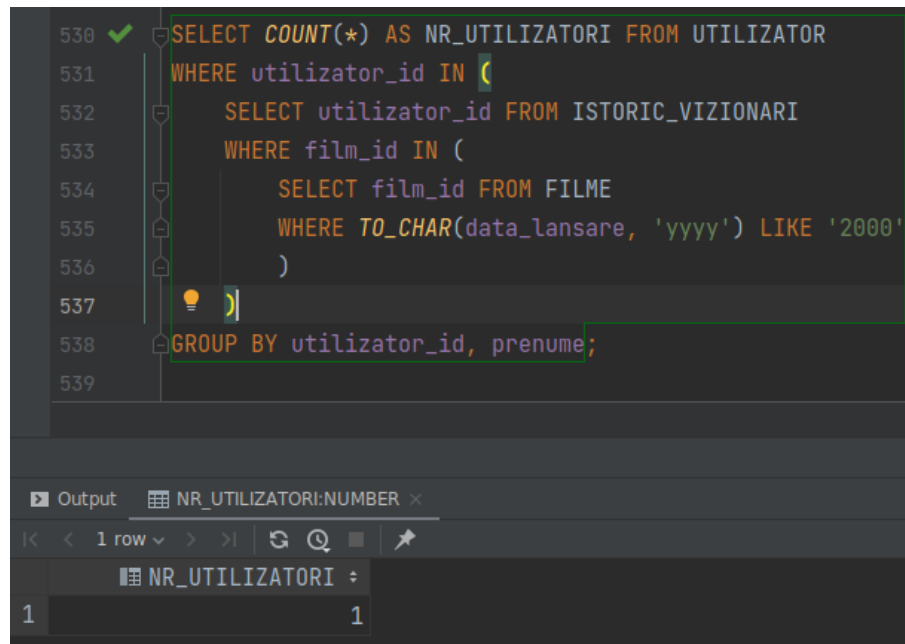
- grupari de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, functii grup, filtrare la nivel de grupuri, o functie pe date calendaristice

```

SELECT COUNT(*) AS NR_UTILIZATORI FROM UTILIZATOR
WHERE utilizator_id IN (
SELECT utilizator_id FROM ISTORIC_VIZIONARI
WHERE film_id IN (
SELECT film_id FROM FILME
WHERE TO_CHAR(data_lansare, 'yyyy') LIKE '2000'

```

```
)
)
GROUP BY utilizator_id, prenume;
```



The screenshot shows a SQL IDE with a query editor and an output window. The query is as follows:

```
530 SELECT COUNT(*) AS NR_UTILIZATORI FROM UTILIZATOR
531 WHERE utilizator_id IN (
532     SELECT utilizator_id FROM ISTORIC_VIZIONARI
533     WHERE film_id IN (
534         SELECT film_id FROM FILME
535         WHERE TO_CHAR(data_lansare, 'yyyy') LIKE '2000'
536     )
537 )
538 GROUP BY utilizator_id, prenume;
```

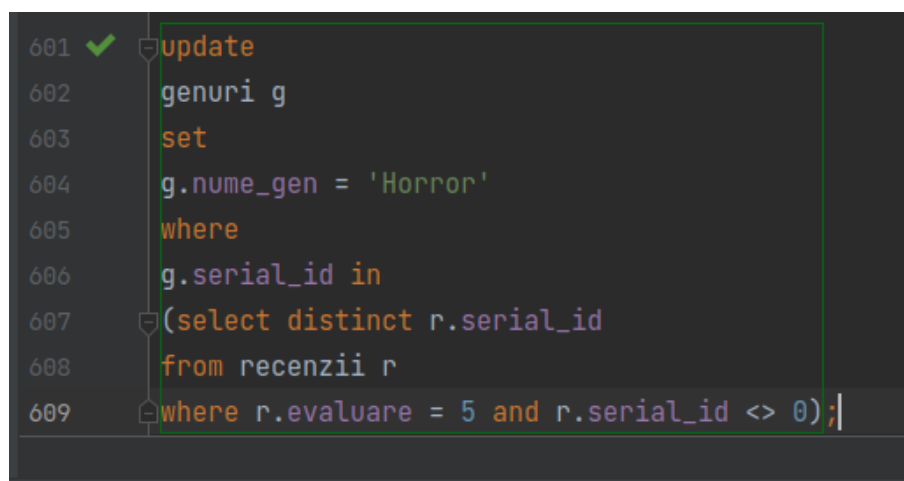
The output window shows the result of the query:

NR_UTILIZATORI
1

## 13 Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri

### 13.1 Sa se schimbe genul serialelor cu evaluarea 5 in Horror

```
update
genuri g
set
g.nume_gen = 'Horror'
where
g.serial_id in
(select distinct r.serial_id
from recenzii r
where r.evaluare = 5 and r.serial_id <> 0);
```

A screenshot of a SQL query in a code editor. The query is: `update genuri g set g.nume_gen = 'Horror' where g.serial_id in (select distinct r.serial_id from recenzii r where r.evaluare = 5 and r.serial_id <> 0);`. The code is color-coded: keywords in orange, table names in green, and string literals in purple. A green box highlights the entire query. On the left, line numbers 601 to 609 are visible, with a green checkmark next to line 601.

### 13.2 Abonamentul utilizatorilor ce au lasat recenzii de mai putin de 70 de caractere sa fie retrogradat la cel gratis

```
update
utilizator u
set
abonament_id = 1
where
u.utilizator_id in (select r.utilizator_id from recenzii r where
length(r.text_evaluare) < 70);
```

```
592 ✓ update
593     utilizator u
594     set
595         abonament_id = 1
596     where
597         u.utilizator_id in (select r.utilizator_id from recenzii r where length(r.text_evaluare) < 70);
598
599
600
601
```

### 13.3 Sa se stearga abonamentele care nu sunt achizitionate de niciun utilizator

delete

from abonamente

where abonament\_id not in (select distinct abonament\_id from utilizator);

```
582 delete
583     from abonamente
584     where abonament_id not in (select distinct abonament_id from utilizator);
585
586 ✓ select * from abonamente;
587
```

Output PROJECT.ABONAMENTE

	ABONAMENT_ID	TIP_ABONAMENT	PRET	DURATA	REDUCERE
1	1	Gratis	0	0	0
2	3	Premium	50	30	0
3	4	Familie	75	30	5
4	5	Familie4K	100	30	15