Baza de date pentru Netflix

Tema 2

Mincu Adrian-Lucian

grupa 232

Cuprins

Contents

[Descriere model 3](#_Toc148269185)

[Constrângeri 3](#_Toc148269186)

[Descriere entități 3](#_Toc148269187)

[Descriere relații 5](#_Toc148269188)

[Atribute (\* ⬄ obligatoriu) 7](#_Toc148269189)

[Diagrama entitate-relație 10](#_Toc148269190)

[Transformare many-to-many 10](#_Toc148269191)

[Diagrama conceptuală 11](#_Toc148269192)

[Definire tabele 12](#_Toc148269193)

[SUBSCRIPȚIE 12](#_Toc148269194)

[FILM 13](#_Toc148269195)

[SUBSCRIPȚIE\_FILM 14](#_Toc148269196)

[Inserări 14](#_Toc148269197)

[SUBSCRIPȚIE 14](#_Toc148269198)

[FILM 15](#_Toc148269199)

[SUBSCRIPȚIE\_FILM 15](#_Toc148269200)

[Query 16](#_Toc148269201)

# 

# 

# Descriere model

Un utilizator iși poate alege o singură subscripție si e obligatoriu să aleagă una, altfel nu va putea folosi serviciile oferite, la care se poate utiliza o singură metodă de plată, și anume card, dar utilizatorul poate avea mai multe card-uri și un card poate fi folosit de mai mulți utilizatori. Filmele sau serialele aparțin unei subscripții, iar utilizatorul se poate uita la ele doar daca este abonat la acea subscripție care le conține.

Fiecare film are un director și trebuie să aibă unul, iar actorii aparțin și filmelor și serialelor, dar pot și să nu fie. Fiecare actor are un rol in film/serial, iar fiecare film are propriile sale roluri. Un actor are un salariu setat pentru filmul si rolul specific.

# Constrângeri

* Un utilizator trebuie să iși aleagă o singură subscripție
* Un utilizator poate plăti cu mai multe card-uri dar trebuie sa aibă unul
* Exista 5 subscripții: basic, standard, normal, premium, ultimate. Ultimate cuprinde toate filmele, premium pe cele din normal standard si basic, normal pe cele din standard si basic si standard doar pe cele din basic, fiind o ierarhie. Totodata o subscriptie de nivel mai mare, poate avea filme pe care nu le are subscriptia de nivel mai mic. Analog pentru seriale.
* Un serial poate să aibă mai multe episoade, dar un episod are un singur serial
* Un film trebuie să aibă un director, dar un director poate să regizeze mai multe filme
* Un actor poate să nu aibă niciun rol, sau mai multe, și poate juca in mai multe filme/seriale, sau niciunul.

# Descriere entități

* **UTILIZATOR**
* persoana fizică care iși poate crea cont pe platforma Netflix
* entitate dependentă de **SUBSCRIPȚIE** și **PLATĂ**
* cheie primară: Utilizator\_ID
* **SUBSCRIPȚIE**
* un pachet care contine filme/seriale cu un pret specific
* entitate independentă
* cheie primară: Subscripție\_ID
* **PLATĂ**
* informațiile cardului cu care plătește utilizatorul
* entitate independentă
* cheie primară: Plată\_ID
* **FILM**
* film, la care se poate uita un utilizator
* entitate dependentă de **DIRECTOR**
* cheie primară: Film\_ID
* **SERIAL**
* serial, la care se poate uita un utilizator
* entitate independentă
* cheie primară: Serial\_ID
* **EPISOD**
* episod, care aparține unui serial
* entitate dependentă de **SERIAL**
* cheieprimară: Episod\_ID
* **ACTOR**
* persoană care joacă într-un film/serial
* entitate independentă
* cheie primară: Actor\_ID
* **ROL**
* caracter dintr-un film care poate fi jucat de un/mai mulți actor/i
* entitate independentă
* cheie primară: Rol\_ID
* **DIRECTOR**
* persoană care regizează un film
* entitate independentă
* cheie primară: Director\_ID

# Descriere relații

UTILIZATOR\_alege\_o\_SUBSCRIPȚIE = relație între UTILIZATOR și SUBSCRIPȚIE

Un UTILIZATOR poate să aleagă decât o singură SUBSCRIPȚIE

O SUBSCRIPȚIE poate să fie aleasă de zero sau mai mulți UTILIZATORI

* Cardinalitate maximă: M:1
* Cardinalitate minimă: 0:1

UTILIZATOR\_adaugă \_o\_PLATĂ = relație între UTILIZATOR și PLATĂ

Un UTILIZATOR poate să adauge mai multe PLĂȚI

O PLATĂ poate să fie adăugată de 1 sau mai mulți UTILIZATORI

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 1:1

SUBSCRIȚIE\_are \_un\_SERIAL = relație între SUBSCRIPȚIE și SERIAL

O SUBSCRIPȚIE poate să aibă zero sau mai multe SERIALE

Un SERIAL poate să aparțină de zero sau mai multe SUBSCRIPȚII

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

SUBSCRIȚIE\_are \_un\_FILM = relație între SUBSCRIPȚIE și FILM

O SUBSCRIPȚIE poate să aibă zero sau mai multe FILME

Un FILM poate să aparțină de zero sau mai multe SUBSCRIPȚII

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

ACTOR\_joacă\_în \_FILM = relație între ACTOR și FILM

Un ACTOR poate să joace în zero sau mai multe FILME

Într-un FILM poate să joace zero sau mai mulți ACTORI

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

ACTOR\_joacă\_în \_SERIAL = relație între ACTOR și SERIAL

Un ACTOR poate să joace în zero sau mai multe SERIALE

Într-un SERIAL poate să joace zero sau mai mulți ACTORI

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

ACTOR\_are \_ROL = relație între ACTOR și ROL

Un ACTOR poate să aibă zero sau mai multe ROLURI

Un ROL poate să aibă zero sau mai mulți ACTORI

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

FILM\_are \_ROL = relație între FILM și ROL

Un FILM poate să aibă zero sau mai multe ROLURI

Un ROL poate să aibă zero sau mai multe FILME

* Cardinalitate maximă: M:M
* Cardinalitate minimă: 0:0

DIRECTOR\_regizează\_un\_FILM = relație între DIRECTOR și FILM

Un DIRECTOR poate să regizeze în unul sau mai multe FILME

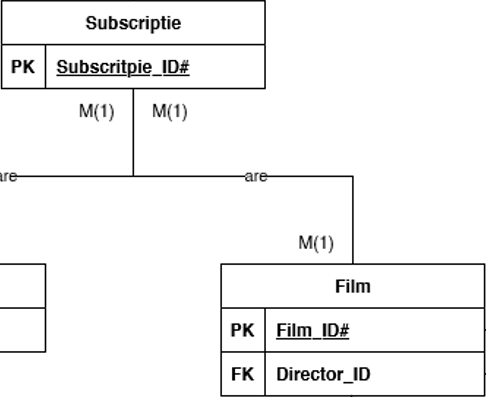
FILM trebuie să fie regizat de un singur DIRECTOR

* Cardinalitate maximă: 1:M
* Cardinalitate minimă: 1:1

# Atribute (\* ⬄ obligatoriu)

* **UTILIZATOR**
* Utilizator\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru întregistrarea contului unui utilizator
* Subscripție \_ID\* = cheie străină din tabela Subscripție
* Poreclă\* = variabilă de tip varchar2, de lungime 50
* Mail\* = variabilă de tip varchar2 de lungime 255
* Parolă\* = variabilă de tip varchar2 de lungime 255
* DatăCreare = variabilă de tip date (valoare implicită = dată curentă)
* Data\_Exp\_Sub\* = date, cand utilizatorul trebuie sa plătească iar
* **SUBSCRIPȚIE**
* Subscriptie\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru pachetele cu filme/seriale
* Tip\* = varchar2(20) dintre valorile: basic, simplu, normal, premium, ultimate, basic e implicită
* Cost = variabilă de tip number, de 2 cifre
* **PLATĂ**
* Plată\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru evidența plăților
* Nume\* = varchar2(50)
* Prenume\* = varchar2(50)
* Cod\* = number(16)
* DatăExp\* = date
* CVV\* = number(3), obligatoriu de 3 cifre
* **FILM**
* Film\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru evidența filmelor
* Director\_ID\* = cheie străină din tabela Director
* Denumire\* = varchar2(50)
* Notă = number(4,2), număr între 1-10
* DatăApariție = date
* **SERIAL**
* Serial\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru evidența serialelor
* Denumire\* = varchar2(50)
* Notă = number(4,2), număr între 1-10
* DatăApariție = date
* **EPISOD**
* Episod\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru evidența episoadelor
* Serial\_ID\* = cheie străină din tabela Serial
* Număr = number(2), numărul episodului în serial
* Denumire\* = varchar2(50)
* Durată\* = number(3), unitate de măsură = minut
* **ACTOR**
* Actor\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru fiecare actor
* Nume\* = varchar2(50)
* Prenume\* = varchar2(50)
* DatăNaștere = date
* **ROL**
* Rol\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru evidența rolurilor
* Tip\* = varchar2(50)
* Descriere = varchar2(255)
* Importanță = number(4,2), număr între 1-10
* **DIRECTOR**
* Director\_ID\* = cod numeric unic, de tip number, de 6 cifre pentru fiecare director
* Nume\* = varchar2(50)
* Prenume = varchar2(50)
* DatăNaștere = date
* Notă = number(4,2), număr între 1-10
* **Subscripție\_Serial**
* SubscripțieSerial\_ID = cod numeric unic, de tip number, de 6 cifre
* Serial\_ID = cheie străină din tabela Serial
* Subscripție\_ID = cheie străină din tabela Subsripție
* **Subscripție\_Film**
* SubscripțieFilm\_ID = cod numeric unic, de tip number, de 6 cifre
* Film\_ID = cheie străină din tabela Film
* Subscripție\_ID = cheie străină din tabela Subsripție
* **Serial\_Actor**
* SerialActor\_ID = cod numeric unic, de tip number, de 6 cifre
* Serial\_ID = cheie străină din tabela Serial
* Actor\_ID = cheie străină din tabela Actor
* **Rol\_Jucat**
* FilmRolActor\_ID = cod numeric unic, de tip number, de 6 cifre
* Film\_ID = cheie străină din tabela FILM
* Rol\_ID = cheie străină din tabela ROL
* Actor\_ID = cheie străină din tabela ACTOR
* Timp\_Ecran = number(3), unitate de măsură = minut
* Salariu = number(7), valoare implicită = 3000

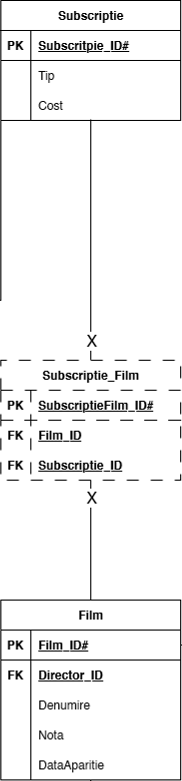
# Diagrama entitate-relație

****

# Transformare many-to-many

Între tabela Subscripție si tabela Film există o relație de tip MANY-TO-MANY, pentru a rezolva această relație m-am folosit de tabela asociativă Subscripție\_Film care are ca si cheie primară Subscripție\_ID, incrementată automat cu ajutorul unei secvențe, iar ca atribute conține cheile străine Film\_ID și Subscripție\_ID, cu un constraint de UNIQUE pentru tuplul (Film\_ID, Subscripție\_ID).

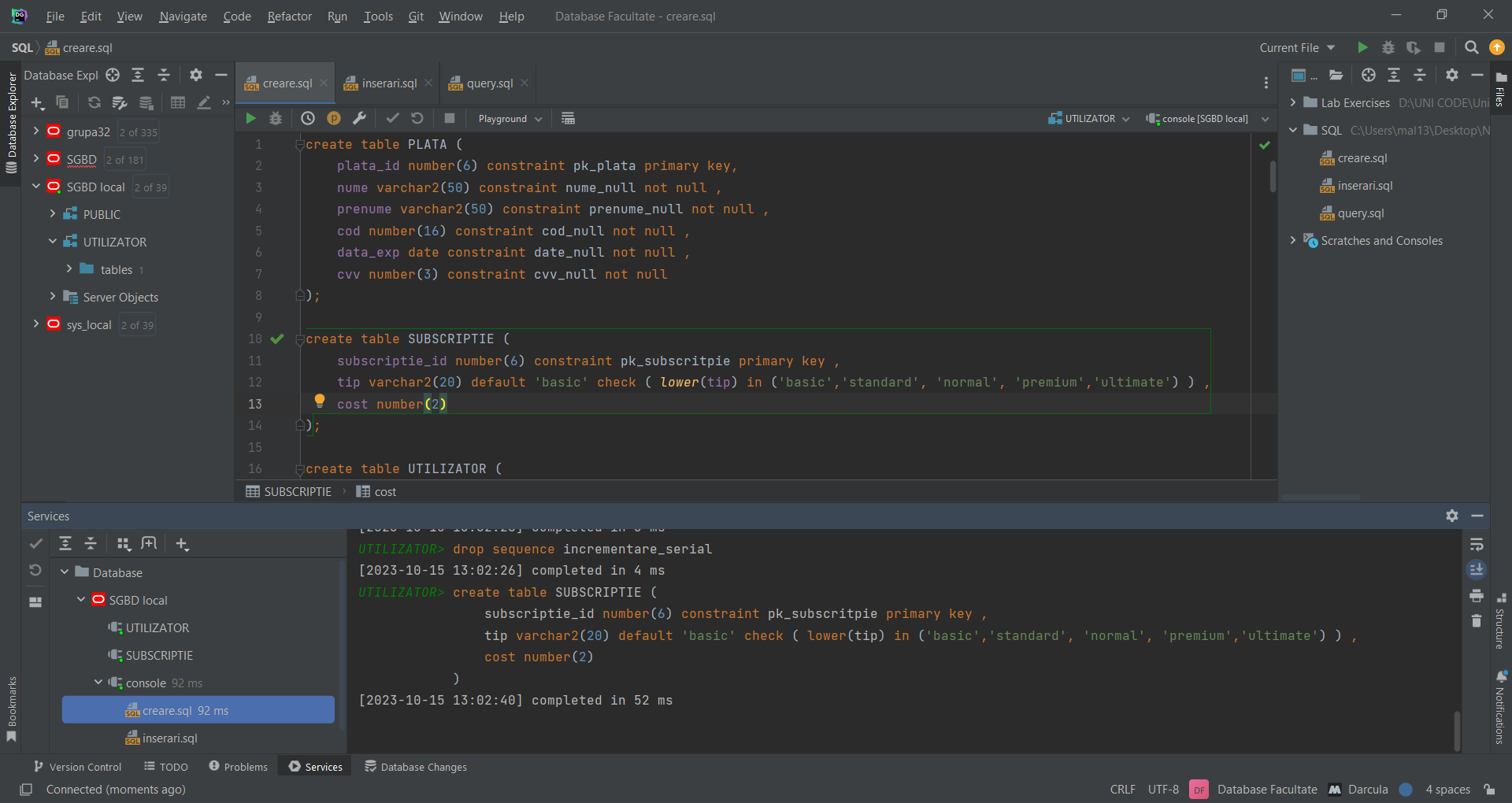
# Diagrama conceptuală



# Definire tabele

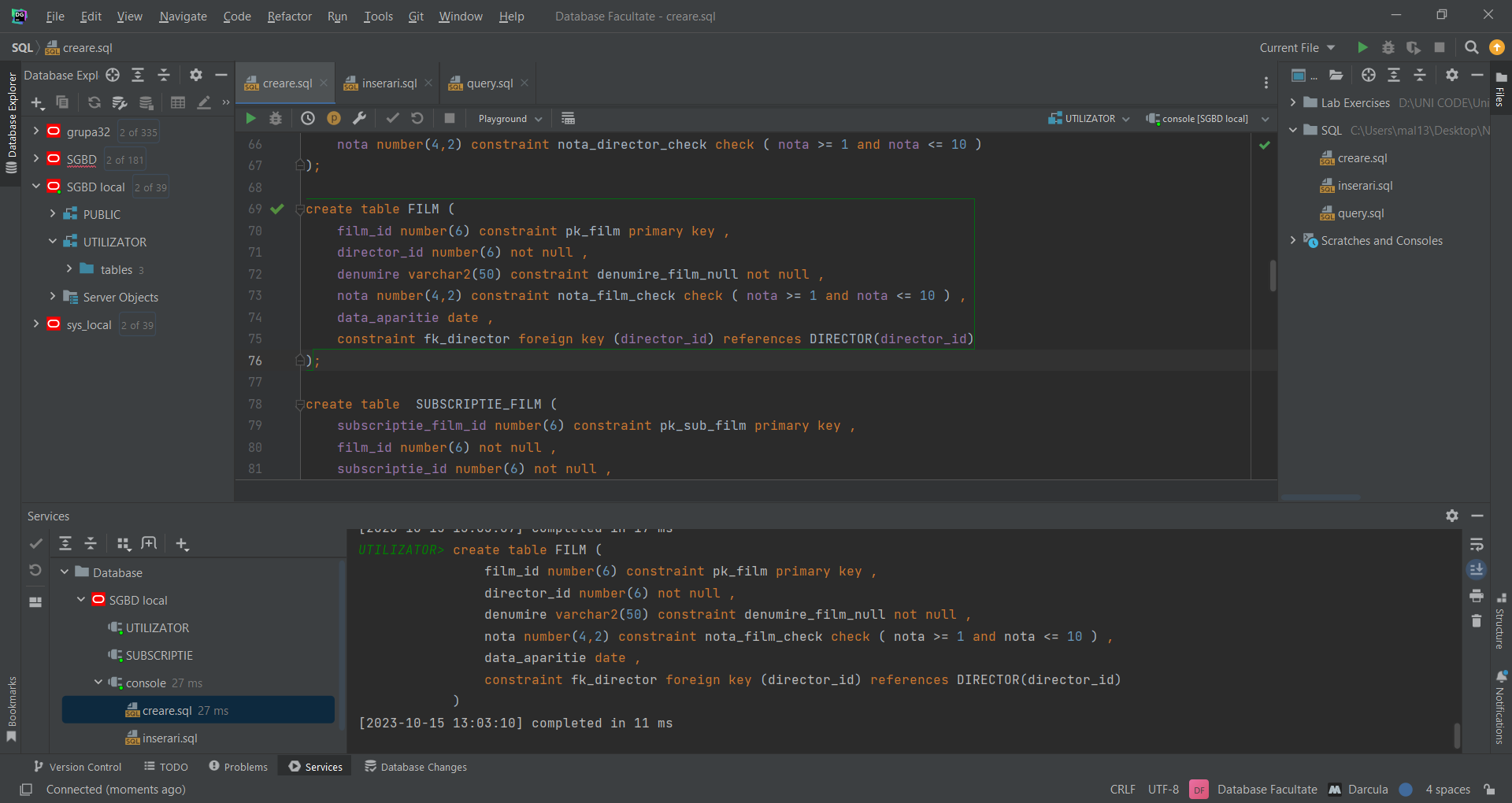
## SUBSCRIPȚIE

create table SUBSCRIPTIE (  
 subscriptie\_id number(6) constraint pk\_subscritpie primary key ,  
 tip varchar2(20) default 'basic' check ( *lower*(tip) in ('basic','standard', 'normal', 'premium','ultimate') ) ,  
 cost number(2)  
);



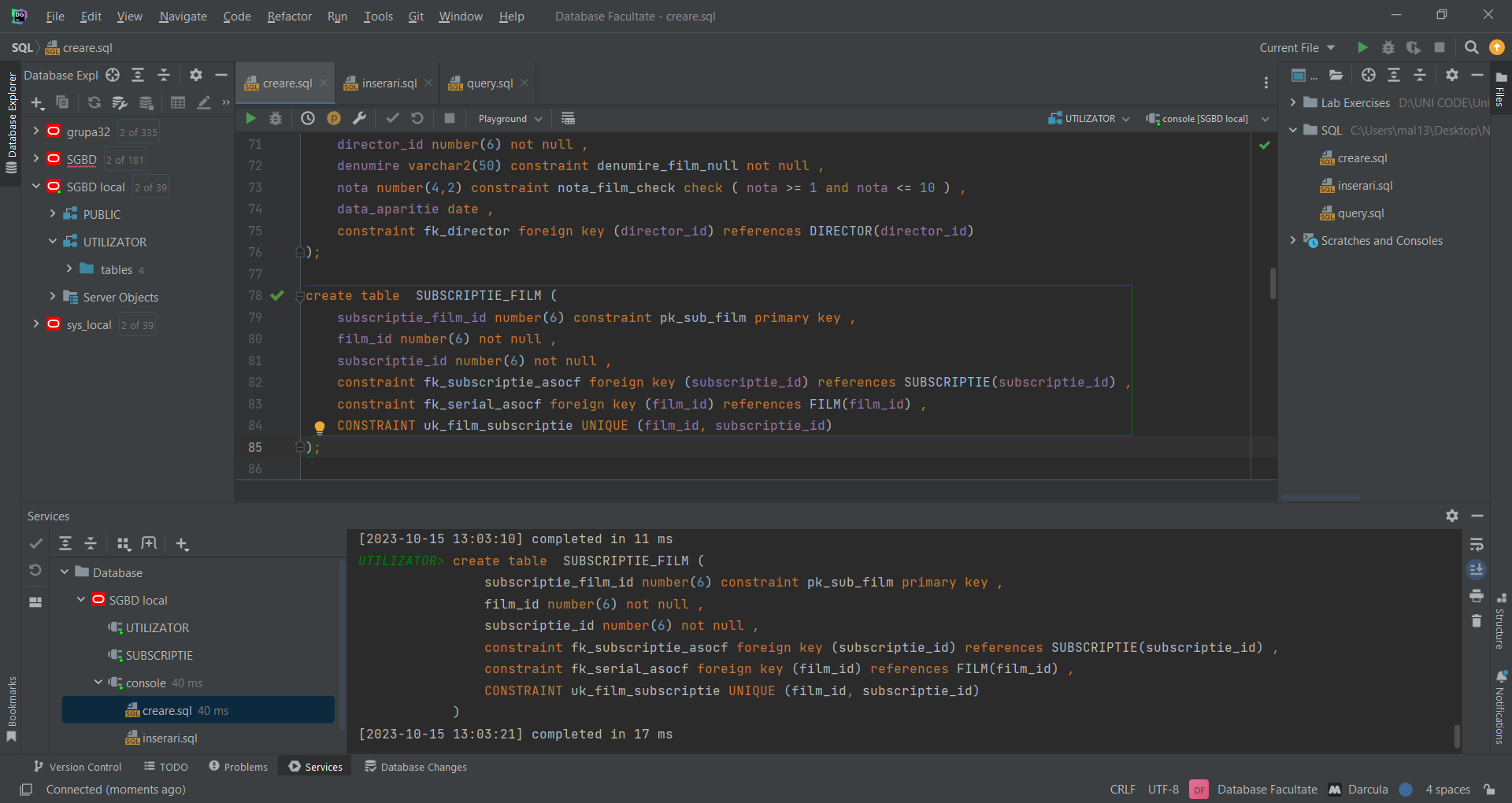
## FILM

create table FILM (  
 film\_id number(6) constraint pk\_film primary key ,  
 director\_id number(6) not null ,  
 denumire varchar2(50) constraint denumire\_film\_null not null ,  
 nota number(4,2) constraint nota\_film\_check check ( nota >= 1 and nota <= 10 ) ,  
 data\_aparitie date ,  
 constraint fk\_director foreign key (director\_id) references DIRECTOR(director\_id)  
);



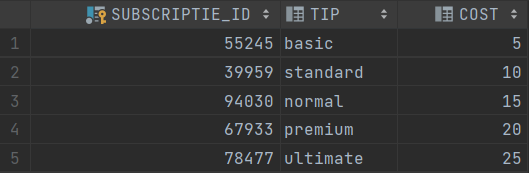
## SUBSCRIPȚIE\_FILM

create table SUBSCRIPTIE\_FILM (  
 subscriptie\_film\_id number(6) constraint pk\_sub\_film primary key ,  
 film\_id number(6) not null ,  
 subscriptie\_id number(6) not null ,  
 constraint fk\_subscriptie\_asocf foreign key (subscriptie\_id) references SUBSCRIPTIE(subscriptie\_id) ,  
 constraint fk\_serial\_asocf foreign key (film\_id) references FILM(film\_id) ,  
 CONSTRAINT uk\_film\_subscriptie UNIQUE (film\_id, subscriptie\_id)  
);



# Inserări

## SUBSCRIPȚIE

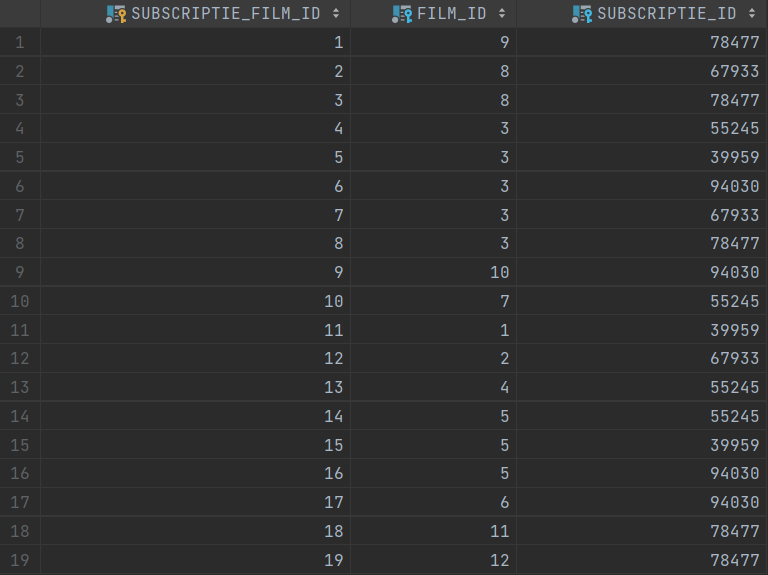


## FILM



## SUBSCRIPȚIE\_FILM

Inserările actuale nu respectă modelul, în sensul în care un film basic trebuie să apară in toate celelalte subcripții. De ce? Pentru că am făcut query-ul înainte să fac aceste schimbări la model, și nu ar mai funcționa altfel.



# Query

Cerintă: Există o subscripție care să conțină toate filmele cu nota 10? Să se afișeze DA sau NU corespunzător.

-- exista o subscriptie care sa contina toate filmele cu nota 10? Sa se afiseze DA sau NU corespunzator.  
select case when not *exists*(  
 (select FILM\_ID  
 from FILM  
 where NOTA = 10)  
MINUS  
 (select f.FILM\_ID  
 from SUBSCRIPTIE\_FILM s2, film f  
 where f.FILM\_ID = s2.FILM\_ID))  
then 'DA'  
else 'NU'  
end Exista  
from dual;

