

PROIECT BAZE DE DATE
TEMĂ: GESTIUNEA UNUI LANȚ DE
MAGAZINE DE DVD-URI

LECTOR DOCTOR MARIN LETIȚIA ANA

STUDENTĂ: NEAGU MARA TEODORA

SERIA 23, GRUPA 231

ANUL II, SEMESTRUL II

2023

CUPRINS

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
2. Prezentarea constrângerilor impuse asupra modelului.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
6. Realizarea diagramei entitate-relație.
7. Realizarea diagramei conceptuale.
8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale.
9. Realizarea normalizării până la forma normală 3.
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre aceste.
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe.
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.
14. Implementarea unei vizualări complexe.
15. Cerere ce utilizează outer – join pe 4 tabele, cerere ce utilizează operația division, cerere ce implementează analiză top – n.
16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale.
17. a) Realizarea normalizării BCNF, FN4, FN5.
b) Aplicarea denormalizării, justificând necesitatea acesteia.

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

În acest proiect vom analiza gestiunea unui lanț de magazine de DVD-uri. Acesta va ajuta la o mai bună funcționare, la posibilitatea de a ține evidența atât a comenzilor plasate în magazine, cât și a închirierilor de filme. Regulile de funcționare pentru acest model sunt următoarele:

- ⊗ Fiind o franciză, magazinele vor avea numele comun: „*The Velvet Stage*“. Totuși, acestea pot avea variații în funcție de județ sau oraș; astfel, numele este consemnat în baza de date.
- ⊗ Angajații magazinul pot ocupa un singur post, de vânzător.
- ⊗ Angajații se pot ocupa atât de închirieri, cât și de vânzarea DVD-urilor. Ambele sunt realizate exclusiv în incinta magazinului, neexistând opțiunea plasării comenzilor la domiciliu.
- ⊗ Comenzile și închirierile nu sunt singulare pentru un DVD.
- ⊗ Prețul închirierii unui DVD este diferit față de cel de cumpărare.
- ⊗ DVD-urile care sunt închiriate/vândute în magazine prezintă informații despre limba vorbită în film; aceste DVD-uri pot avea și traduceri în alte limbi, dar nu este o opțiune obligatorie.
- ⊗ Termenul de retur pentru un film este de o săptămână, fiind calculat după data închirierii.
- ⊗ Categoriile de filme posibile sunt: *ACȚIUNE, AVENTURĂ, COMEDIE, COPII, DRAMATIC, GROAZĂ, MUZICALE, POLIȚISTE, ROMANTICE, SCIENCE FICTION*.

2. Prezentarea constrângerilor impuse asupra modelului.

Constrângerile acestui model sunt următoarele:

- ⊗ Un magazin trebuie să aibă minim un film în inventarul lor.
- ⊗ Un magazin trebuie să aibă minim un angajat.
- ⊗ Un angajat poate lucra într-un singur magazin.
- ⊗ Un angajat poate procesa atât comenzi, cât și închirieri de filme.
- ⊗ Un client poate face mai multe comenzi sau închirieri în aceeași zi; acesta poate cumpăra din nou același film, dar nu poate închiria un film deja închiriat de acesta, decât dacă perioadă de închiriere a expirat.
- ⊗ Numărul de telefon și email-ul clienților sunt opționale.
- ⊗ Filmele pot fi încadrate într-o singură categorie.
- ⊗ Filmele pot fi regizate de un singur regizor.
- ⊗ Regizorul trebuie să fii regizat minim un film din inventarul magazinului.
- ⊗ Un actor trebuie să fii jucat în minim un film din inventarul magazinului.
- ⊗ Filmele pot avea sau nu descriere sau rating.
- ⊗ Rating-ul unui film este între 1 și 10.
- ⊗ Traducerea unui film este opțională.

3. Descrierea entităților, incluzând precizarea cheii primare.

Pentru modelul de date referitor la lanțul de magazine de DVD-uri, structurile *store_mng*, *employee_mng*, *film_mng*, *category_mng*, *actor_mng*, *director_mng*, *language_mng*, *customer_mng*, *rental_mng*, *order_mng* reprezintă entități.

✿ *STORE_MNG*

- magazin care se ocupă cu vânzarea și închirierea fizică a DVD-urilor. Entitatea este necesară deoarece modelăm un lanț de magazine. Cheia primară a entității este *#store_id*.

✿ *EMPLOYEE_MNG*

- persoană fizică, angajată a unui magazin de DVD-uri. Cheia primară a entității este *#emp_id*.

✿ *FILM_MNG*

- identifică informații legate de un film al magazinului de DVD-uri. Cheia primară este *#film_id*.

✿ *CATEGORY_MNG*

- identifică categoria în care se încadrează un film al magazinului de DVD-uri. Cheia primară este *#category_id*.

✿ *ACTOR_MNG*

- identifică informații legate de un actor, care joacă în unul sau mai multe filme ale magazinului de DVD-uri. Cheia primară este *#actor_id*.

✿ *DIRECTOR_MNG*

- identifică informații legate de un regizor, care a regizat unul sau mai multe filme ale magazinului de DVD-uri. Cheia primară este *#director_id*.

✿ *LANGUAGE_MNG*

- identifică informații legate de o limbă, în care a fost tradus unul sau mai multe filme ale magazinului de DVD-uri. Cheia primară este *#language_id*.

✿ *CUSTOMER_MNG*

- persoană fizică, care poate închiria sau cumpăra unul sau mai multe filme ale magazinului de DVD-uri. Cheia primară este *#cust_id*.

✿ *RENTAL_MNG*

- identifică informații legate de închirierea unui film al magazinului de DVD-uri. Cheia primară este *#rental_id*.

✿ *ORDER_MNG*

- identifică informații legate de o comandă plasată în magazinul de DVD-uri. Cheia primară este *#order_id*.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Relațiile acestui model sunt următoarele:

⊗ *employee_mng* ***WORKS_IN*** *store_mng*

- relație de tip many – to – many ce leagă entitățile *store_mng* și *employee_mng*, reflectând legătura dintre acestea (un angajat lucrează într-un magazin);
- *cardinalitate minimă*: unu – unu (1:1)
 - În câte magazine trebuie să lucreze un angajat? Cel puțin unul.
 - Câți angajați trebuie să aibă un magazin? Cel puțin unul.
- *cardinalitate maximă*: many – many (m:n)
 - În câte magazine poate lucra un angajat? Multe.
 - Câți angajați poate avea un magazin? Mulți.

⊗ *store_mng* ***STORE_INVENTORY*** *film_mng*

- relație de tip many – to – many ce leagă entitățile *store_mng* și *film_mng*, reflectând legătura dintre acestea (un magazin are un inventar de filme);
- *cardinalitate minimă*: unu – unu (1:1)
 - Câte filme trebuie să aibă un magazin în inventar? Cel puțin unul.
 - În inventarul câtor magazine trebuie să se afle un film? Cel puțin unul.
- *cardinalitate maximă*: many – many (m:n)
 - Câte filme poate avea un magazin în inventar? Multe.
 - În inventarul câtor magazine se poate afla un film? Multe.

⊗ *film_mng* ***BELONGS_TO*** *category_mng*

- relație ce leagă entitățile *film_mng* și *category_mng*, reflectând legătura dintre acestea (un film aparține unei categorii);
- *cardinalitate minimă*: zero – unu (0:1)
 - În câte categorii trebuie să se încadreze un film? Cel puțin una.
 - Câtor filme trebuie să le fie atribuită o categorie? Zero.
- *cardinalitate maximă*: many – zero (m:1)
 - În câte categorii poate fi încadrat un film? Cel mult una.
 - Câtor filme pot să le fie atribuită o categorie? Multe.

⊗ *actor_mng* ***ACTS_IN*** *film_mng*

- relație many – to – many ce leagă entitățile *actor_mng* și *film_mng*, reflectând legătura dintre acestea (un actor joacă într-un film);
- *cardinalitate minimă*: unu – unu (1:1)
 - În câte filme trebuie să joace un actor? Cel puțin unul.
 - Câți actori trebuie să aibă un film în distribuția lor? Cel puțin unul.
- *cardinalitate maximă*: many – many (m:n)
 - În câte filme poate juca un actor? Multe.
 - Câți actori pot avea un film în distribuția lor? Mulți.

⊗ *film_mng* **DIRECTED_BY** *director_mng*

- relație ce leagă entitățile *film_mng* și *director_mng*, reflectând legătura dintre acestea (un film este regizat de un regizor);
- *cardinalitate minimă*: unu – unu (1:1)
 - Câți regizori trebuie să regizeze un film? Cel puțin unul.
 - Câte filme trebuie să regizeze un regizor? Cel puțin unul.
- *cardinalitate maximă*: many – unu (m:1)
 - Câți regizori pot regiza un film? Cel mult unul.
 - Câte filme pot fi regizate de un regizor? Multe.

⊗ *film_mng* **TRANSLATED_TO** *language_mng*

- relație many – to – many ce leagă entitățile *film_mng* și *language_mng*, reflectând legătura dintre acestea (un film este tradus într-o limbă).
- *cardinalitate minimă*: unu – zero (1:0)
 - Câte filme trebuie să fie traduse într-o anumită limbă? Cel puțin unul.
 - În câte limbi trebuie să fie tradus un film? Zero.
- *cardinalitate maximă*: many – many (m:n)
 - Câte filme pot fi traduse într-o anumită limbă? Multe.
 - În câte limbi poate fi tradus un film? Multe.

⊗ *employee_mng* **PROCESSES** *rental_mng*

- relație ce leagă entitățile *employee_mng* și *rental_mng*, reflectând legătura dintre acestea (un angajat procesează o închiriere);
- *cardinalitate minimă*: unu – zero (1:0)
 - Câte închirieri trebuie procesate de un angajat? Zero.
 - Câți angajați trebuie să proceseze o închiriere? Cel puțin unul.
- *cardinalitate maximă*: unu – many (1:n)
 - Câte închirieri pot fi procesate de un angajat? Multe.

- Câți anjați pot să proceseze o închiriere? Cel mult unul.

⊗ *employee_mng* **PROCESSES** *order_mng*

- relație ce leagă entitățile *employee_mng* și *order_mng*, reflectând legătura dintre acestea (un angajat procesează o comandă);
- *cardinalitate minimă*: unu – zero (1:0)
 - Câte comenzi trebuie să proceseze un angajat? Zero.
 - Câți angajați trebuie să proceseze o comandă? Cel puțin unul.
- *cardinalitate maximă*: unu – many (1:n)
 - Câte comenzi poate procesa un angajat? Multe.
 - Câți angajați pot procesa o comandă? Cel mult unul.

⊗ *customer_mng* **MAKES** *rental_mng*

- relație ce leagă entitățile *customer_mng* și *rental_mng*, reflectând legătura dintre acestea (un client face o închiriere);
- *cardinalitate minimă*: unu – unu (1:1)
 - Câte închirieri trebuie să facă un client? Cel puțin una.
 - Câți clienți trebuie să facă o închiriere? Cel puțin unul.
- *cardinalitate maximă*: unu – many (1:n)
 - Câte închirieri poate face un client? Multe.
 - Câți clienți pot face o închiriere? Cel mult unul.

⊗ *customer_mng* **MAKES** *order_mng*

- relație ce leagă entitățile *customer_mng* și *order_mng*, reflectând legătura dintre acestea (un client face o comandă);
- *cardinalitate minimă*: unu – unu (1:1)
 - Câte comenzi trebuie să facă un client? Cel puțin una.
 - Câți clienți trebuie să facă o comandă? Cel puțin unul.
- *cardinalitate maximă*: unu – many (1:n)
 - Câte comenzi poate face un client? Multe.
 - Câți clienți pot face o comandă? Cel mult unul.

⊗ *rental_mng* **RENTAL CONSISTS OF** *film_mng*

- relație de tip many – to – many ce leagă entitățile *rental_mng* și *film_mng*, reflectând legătura dintre acestea (o închiriere este formată din filme);
- *cardinalitate minimă*: zero – unu (0:1)
 - Din câte filme trebuie să fie formată o închiriere? Cel puțin unul.

- Din câte închirieri trebuie să facă parte un film? Zero.
- *cardinalitate maximă*: many – many (m:n)
 - Din câte filme poate să fie formată o închiriere? Multe.
 - Din câte închirieri poate să facă parte un film? Multe.

⊗ *order_mng_***ORDER_CONSISTS_OF***_film_mng*

- relație de tip many – to – many ce leagă entitățile *order_mng* și *film_mng*, reflectând legătura dintre acestea (o comandă este formată din filme).
- *cardinalitate minimă*: zero – unu (0:1)
 - Din câte filme trebuie să fie formată o comandă? Cel puțin unul.
 - Din câte comenzi trebuie să facă parte un film? Zero.
- *cardinalitate maximă*: many – many (m:n)
 - Din câte filme poate să fie formată o comandă? Multe.
 - Din câte comenzi poate să facă parte un film? Multe.

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

Entitatea independentă *store_mng* are ca atribute:

- ✿ *store_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui magazin de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *store_name* – variabilă de tip caracter, de lungime maximă 50, ce reprezintă numele magazinului de DVD-uri. Constrângere de tip CHECK: NOT NULL.
- ✿ *manager_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul manager-ului magazinului de DVD-uri. Constrângere de tip CHECK: NOT NULL și UNIQUE.
- ✿ *adresa* – variabilă de tip text, de lungime maximă 40, ce reprezintă adresa magazinului de DVD-uri. Constrângere de tip CHECK: NOT NULL.
- ✿ *city* – variabilă de tip caracter, de lungime maximă 15, ce reprezintă orașul în care este localizat magazinul de DVD-uri. Constrângere de tip CHECK: NOT NULL.
- ✿ *state* – variabilă de tip caracter, de lungime maximă 15, ce reprezintă județul în care este localizat magazinul de DVD-uri. Poate fi NULL.
- ✿ *zip* – variabilă de tip caracter, de lungime maximă 6, ce reprezintă codul poștal al zonei în care este localizat magazinul de DVD-uri. Constrângere de tip CHECK: NOT NULL.
- ✿ *phone* – variabilă de tip caracter, de lungime maximă 10, ce reprezintă numărul de telefon prin care poate fi contactat magazinul de DVD-uri. Constrângere de tip CHECK: NOT NULL și UNIQUE.
- ✿ *email* – variabilă de tip caracter, de lungime maximă 20, ce reprezintă email-ul prin care poate fi contactat magazinul de DVD-uri. Constrângere de tip CHECK: NOT NULL și UNIQUE.

Entitatea independentă *employee_mng* are ca atribute:

- ✿ *emp_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui angajat al magazinului de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *emp_last* – variabilă de tip caracter, de lungime maximă 25, ce reprezintă numele angajatului. Constrângere de tip CHECK: NOT NULL.
- ✿ *emp_first* – variabilă de tip caracter, de lungime maximă 25, ce reprezintă prenumele angajatului. Constrângere de tip CHECK: NOT NULL.
- ✿ *emp_phone* – variabilă de tip caracter, de lungime maximă 10, ce reprezintă numărul de telefon prin care poate fi contactat angajatul. Constrângere de tip CHECK: NOT NULL și UNIQUE.

- ✿ *emp_email* – variabilă de tip caracter, de lungime maximă 30, ce reprezintă email-ul prin care poate fi contactat angajatul. Constrângere de tip CHECK: NOT NULL și UNIQUE.
- ✿ *emp_hire_date* – variabilă de tip dată calendaristică, ce reprezintă data angajării salariatului.
- ✿ *emp_salary* – variabilă de tip întreg, de lungime maximă 4, ce reprezintă salariul angajatului. Constrângere de tip CHECK: NOT NULL.

Entitatea independentă *film_mng* are ca atribute:

- ✿ *film_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui film care este pus la vânzare/închiriere în magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *film_title* – variabilă de tip caracter, de lungime maximă 40, ce reprezintă titlul filmului. Constrângere de tip CHECK: NOT NULL.
- ✿ *film_description* – variabilă de tip text, de lungime maximă 500, ce include o scurtă descriere a filmului, plus informații adiționale (numele actorilor principali, numele companiei de producție, etc.). Poate fi NULL.
- ✿ *film_release* – variabilă de tip dată calendaristică, ce reprezintă data apariției filmului.
- ✿ *film_original_language* – variabilă de tip caracter, de lungime maximă 15, ce reprezintă limba vorbită în cadrul filmului. Constrângere de tip CHECK: NOT NULL.
- ✿ *film_length* – variabilă de tip întreg, de lungime maximă 3, ce reprezintă durata filmului, în minute. Constrângere de tip CHECK: NOT NULL.
- ✿ *film_rental_price* – variabilă de tip întreg, de lungime maximă 2, ce reprezintă prețul închirierii filmului. Constrângere de tip CHECK: NOT NULL.
- ✿ *film_order_price* – variabilă de tip întreg, de lungime maximă 2, ce reprezintă prețul cumpărării unui DVD. Constrângere de tip CHECK: NOT NULL.
- ✿ *film_rating* – variabilă de tip numeric, de lungime maximă 1, ce reprezintă rating-ul filmului, având maximul 10.
- ✿ *director_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui regizor al unui/mai multor filme din magazinul de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *director_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *category_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unei categorii în care se încadrează unul/mai multe filme din magazinul de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *category_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.

Entitatea independentă *category_mng* are ca atribute:

- ⊗ *category_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unei categorii în care se încadrează unul/mai multe filme care sunt puse la vânzare/închiriere în magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ⊗ *category_name* = variabilă de tip text, de lungime maximă 20, ce reprezintă numele unei categorii. Constrângere de tip CHECK: NOT NULL și UNIQUE.

Entitatea independentă *actor_mng* are ca attribute:

- ⊗ *actor_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui actor care a jucat în unul/mai multe filme care sunt puse la vânzare/închiriere în magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ⊗ *actor_last* – variabilă de tip caracter, de lungime maximă 20, ce reprezintă numele unui actor. Constrângere de tip CHECK: NOT NULL.
- ⊗ *actor_first* – variabilă de tip caracter, de lungime maximă 20, ce reprezintă prenumele unui actor. Constrângere de tip CHECK: NOT NULL.

Entitatea independentă *director_mng* are ca attribute:

- ⊗ *director_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui regizor care a regizat unul/mai multe filme care sunt puse la vânzare/închiriere în magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL ȘI UNIQUE, nu are valori DEFAULT.
- ⊗ *director_last* – variabilă de tip caracter, de lungime maximă 20, ce reprezintă numele unui regizor. Constrângere de tip CHECK: NOT NULL.
- ⊗ *director_first* – variabilă de tip caracter, de lungime maximă 20, ce reprezintă prenumele unui regizor. Constrângere de tip CHECK: NOT NULL.

Entitatea independentă *language_mng* are ca attribute:

- ⊗ *language_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unei limbi în care a fost tradus unul/mai multe filme care sunt puse la vânzare/închiriere în magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ⊗ *language_name* – variabilă de tip caracter, de lungime maximă 15, ce reprezintă numele limbii. Constrângere de tip CHECK: NOT NULL și UNIQUE.

Entitatea independentă *customer_mng* are ca attribute:

- ⊗ *cust_id* = variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui client, care plasează o comandă/face o închiriere a unuia/mai multor filme din magazinul de

DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.

- ✿ *cust_last* = variabilă de tip caracter, de lungime maximă 20, ce reprezintă numele clientului. Constrângere de tip CHECK: NOT NULL.
- ✿ *cust_first* = variabilă de tip caracter, de lungime maximă 20, ce reprezintă prenumele clientului. Constrângere de tip CHECK: NOT NULL.
- ✿ *cust_phone* = variabilă de tip caracter, de lungime maximă 10, ce reprezintă numărul de telefon prin care poate fi contactat clientul. Constrângere de tip CHECK: UNIQUE.
- ✿ *cust_email* = variabilă de tip caracter, de lungime maximă 20, ce reprezintă email-ul prin care poate fi contactat clientul. Constrângere de tip CHECK: UNIQUE.

Entitatea independentă *rental_mng* are ca atribute:

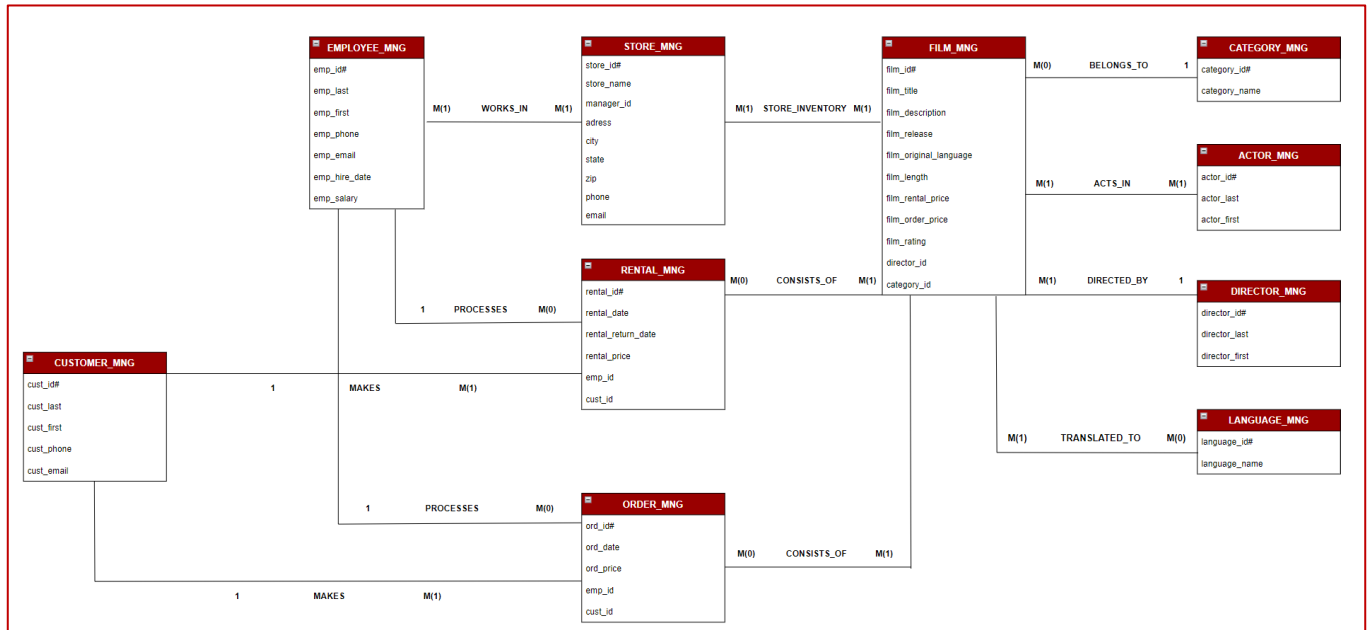
- ✿ *rental_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unei închirieri de film din magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *rental_date* – variabilă de tip dată calendaristică, ce reprezintă data în care a fost făcută închirierea. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul RENTAL_RETURN_MNG. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valori DEFAULT.
- ✿ *rental_return_date* – variabilă de tip dată calendaristică, ce reprezintă data în care filmul a fost returnat la magazinul de DVD-uri.
- ✿ *rental_price* – variabilă de tip întreg, de lungime maximă 6, ce reprezintă prețul unei închirieri. Constrângere de tip CHECK: NOT NULL.
- ✿ *emp_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui angajat al magazinului de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *employee_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valoare DEFAULT.
- ✿ *cust_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui client, care plasează o comandă/face o închiriere a unui/mai multor filme din magazinul de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *customer_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valoare DEFAULT.

Entitatea independentă *order_mng* are ca atribute:

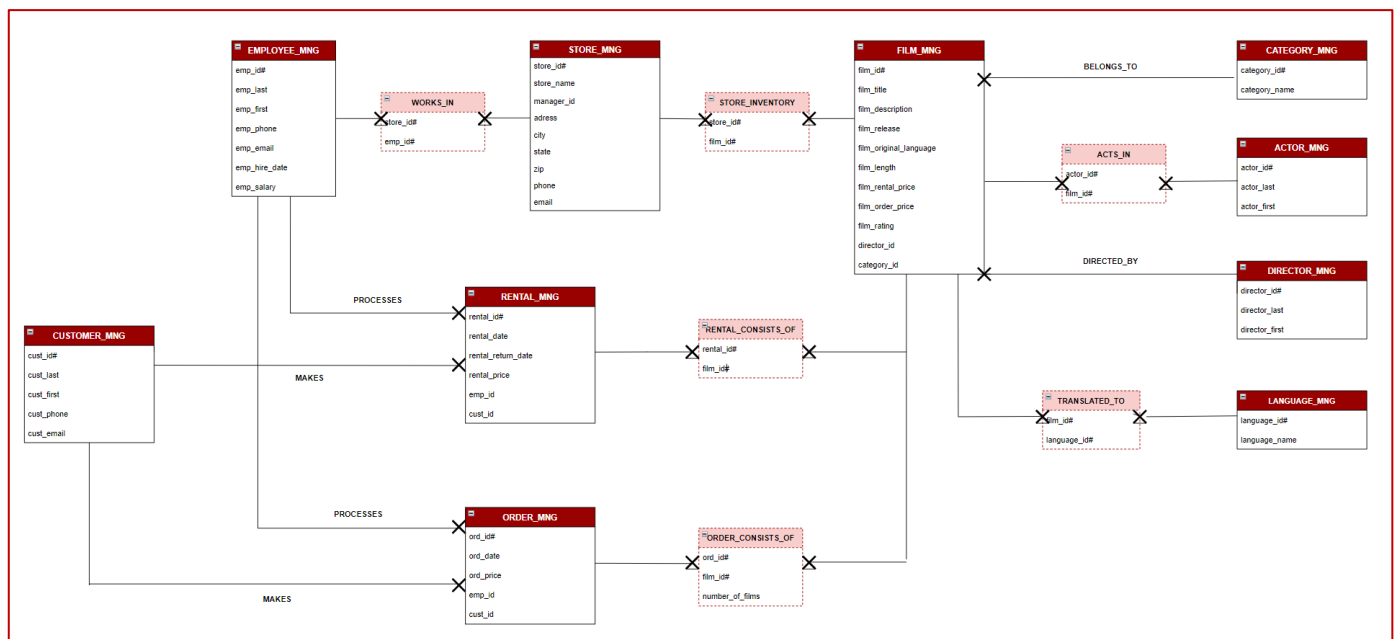
- ✿ *ord_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unei comenzi plasată la magazinul de DVD-uri. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valoare DEFAULT.
- ✿ *ord_date* – variabilă de tip dată calendaristică, ce reprezintă data la care a fost făcută comanda. Constrângere de tip CHECK: NOT NULL.

- ⊗ *ord_price* – variabilă de tip întreg, de lungime maximă 6, ce reprezintă prețul unei comenzi. Constrângere de tip CHECK: NOT NULL.
- ⊗ *emp_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui angajat al magazinului de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *employee_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valoare DEFAULT.
- ⊗ *cust_id* – variabilă de tip întreg, de lungime maximă 5, ce reprezintă codul unui client, care plasează o comandă/face o închiriere a unuia/mai multor filme din magazinul de DVD-uri. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul *customer_mng*. Constrângere de tip PRIMARY KEY: NOT NULL și UNIQUE, nu are valoare DEFAULT.

6. Realizarea diagramei entitate – relație.



7. Realizarea diagramei conceptuale.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale.

Schemele relaționale corespunzătoare diagramei conceptuale sunt următoarele:

- ⊗ *store_mng* (#store_id, store_name, manager_id, adress, city, state, zip, phone, email);
- ⊗ *employee_mng* (#emp_id, emp_last, emp_first, emp_phone, emp_email, emp_hire_date, emp_salary);
- ⊗ *works_in* (#store_id, #emp_id);
- ⊗ *film_mng* (#film_id, film_title, film_description, film_release, film_original_language, film_length, film_rental_price, film_order_price, film_rating, director_id, category_id);
- ⊗ *store_inventory* (#store_id, #film_id);
- ⊗ *category_mng* (#category_id, category_name);
- ⊗ *actor_mng* (#actor_id, actor_last, actor_first);
- ⊗ *acts_in* (#actor_id, #film_id);
- ⊗ *director_mng* (#director_id, director_last, director_first);
- ⊗ *language_mng* (#language_id, language_name);
- ⊗ *translated_to* (#language_id, #film_id);
- ⊗ *customer_mng* (#cust_id, cust_last, cust_first, cust_phone, cust_email);
- ⊗ *rental_mng* (#rental_id, rental_date, rental_return_date, rental_price, emp_id, cust_id);
- ⊗ *rental_consists_of* (#rental_id, #film_id);
- ⊗ *order_mng* (#ord_id, ord_date, ord_price, emp_id, cust_id);
- ⊗ *order_consists_of* (#ord_id, #film_id, number_of_films);

9. Realizarea normalizării până la forma normală 3.

⊗ FN1

O relație este în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă.

store_id#	film_id
S1	F1, F2
S2	F3
S3	F4, F5

După cum putem observa, acest tabel nu se află în FN1 deoarece fiecărui magazin îi corespund mai multe filme în aceeași înregistrare. Vom rezolva această problemă cu tabelul asociativ dintre entitatea *store_mng* și *film_mng*, *store_inventory*;

store_id#	film_id#
S1	F1
S1	F2
S2	F3
S3	F4
S3	F5

⊗ FN2

O relație este în FN2 dacă și numai dacă este în FN1 și fiecare atribut care nu participă la cheia primară este dependent de întreaga cheie primară.

ord_id#	ord_date	ord_price	emp_id	cust_id	number_of_films
O1	16-MAY-22	50	E1	C4	2
O2	17-MAY-22	60	E1	C2	3
O3	18-MAY-22	70	E2	C3	4

Acest tabel nu este în FN2 pentru că *number_of_films* depinde atât de *ord_id*, cât și de *film_id*, numărul de DVD-uri dintr-o comandă fiind individual pentru fiecare film din comandă. Vom transforma în FN2 următoarele tabele *order_mng* și *order_consists_of*:

ord_id#	ord_date	ord_price	emp_id	cust_id
O1	16-MAY-22	50	E1	C4
O2	17-MAY-22	60	E1	C2
O3	18-MAY-22	70	E2	C3

ord_id#	film_id#	number_of_films
O1	F1	2
O2	F2	3
O3	F3	4

✿ FN3

O relație este în FN3 dacă și numai dacă este în FN2 și fiecare atribut care nu participă la o cheie depinde direct de cheia primară.

film_id#	film_title	language_id	language_name
F1	„Titanic”	L2	franceză
F2	„Pride & Prejudice”	L1	engleză
F3	„Jurassic Park”	L3	română

Tabelul nu este în FN3 deoarece *language_name* depinde tranzitiv de cheia primară *film_id#*: *film_id#* → *language_id* → *language_name*. Vom transforma în FN3 este următoarele tabele *film_mng* și *language_mng*:

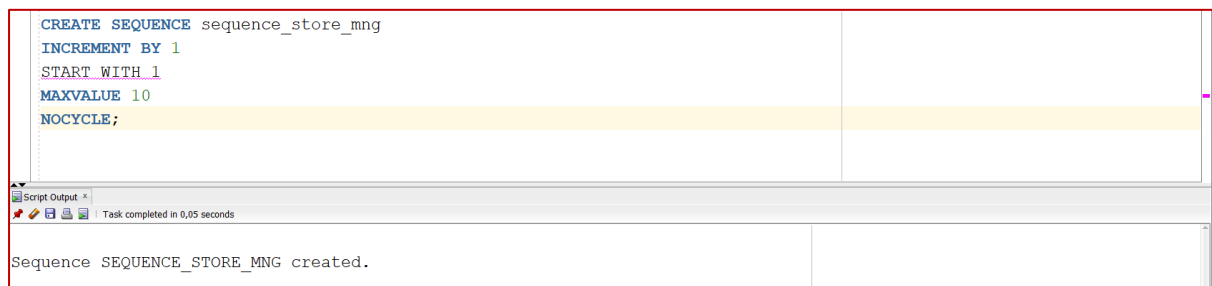
film_id#	film_title	language_id
F1	„Titanic”	L2
F2	„Pride & Prejudice”	L1
F3	„Jurassic Park”	L3

language_id#	language_name
L1	engleză
L2	franceză
L3	română

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

Secvența *sequence_store_mng* determină id-ul unui magazin.

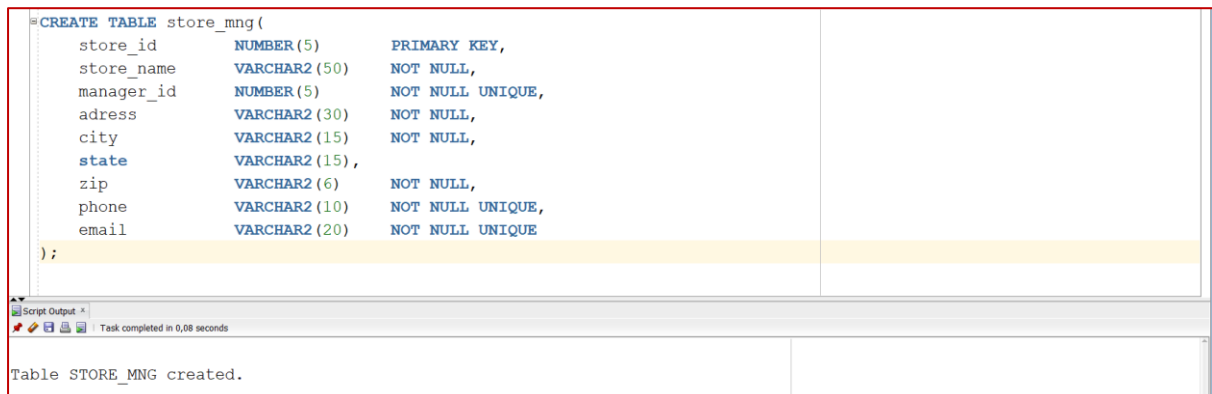
```
CREATE SEQUENCE sequence_store_mng  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 10  
NOCYCLE;
```



The screenshot shows a SQL script editor with the following text: `CREATE SEQUENCE sequence_store_mng`, `INCREMENT BY 1`, `START WITH 1`, `MAXVALUE 10`, and `NOCYCLE;`. Below the editor is a 'Script Output' window that displays the message 'Sequence SEQUENCE_STORE_MNG created.' and a status bar indicating 'Task completed in 0,05 seconds'.

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre aceste.

```
CREATE TABLE store_mng(
  store_id    NUMBER(5)    PRIMARY KEY,
  store_name  VARCHAR2(50) NOT NULL,
  manager_id  NUMBER(5)    NOT NULL UNIQUE,
  address     VARCHAR2(30) NOT NULL,
  city        VARCHAR2(15) NOT NULL,
  state       VARCHAR2(15),
  zip         VARCHAR2(6)  NOT NULL,
  phone       VARCHAR2(10) NOT NULL UNIQUE,
  email       VARCHAR2(20) NOT NULL UNIQUE
);
```



```
CREATE TABLE employee_mng(
  emp_id      NUMBER(5)    PRIMARY KEY,
  emp_last    VARCHAR2(25) NOT NULL,
  emp_first   VARCHAR2(25) NOT NULL,
  emp_phone   VARCHAR2(10) NOT NULL UNIQUE,
  emp_email   VARCHAR2(10) NOT NULL UNIQUE,
  emp_hire_date DATE,
  emp_salary  NUMBER(4,2)  NOT NULL
);
```

```

CREATE TABLE employee_mng(
  emp_id      NUMBER(5)      PRIMARY KEY,
  emp_last    VARCHAR2(25)    NOT NULL,
  emp_first   VARCHAR2(25)    NOT NULL,
  emp_phone   VARCHAR2(10)    NOT NULL UNIQUE,
  emp_email   VARCHAR2(10)    NOT NULL UNIQUE,
  emp_hire_date DATE,
  emp_salary  NUMBER(4,2)     NOT NULL
);

```

Script Output: Task completed in 0,07 seconds

Table EMPLOYEE_MNG created.

```

CREATE TABLE works_in(
  store_id    NUMBER(5)      REFERENCES store_mng(store_id) ON DELETE
CASCADE,
  emp_id      NUMBER(5)      REFERENCES employee_mng(emp_id) ON DELETE
CASCADE,
  CONSTRAINT pk_works_in PRIMARY KEY(store_id, emp_id));

```

```

CREATE TABLE works_in(
  store_id    NUMBER(5)      REFERENCES store_mng(store_id) ON DELETE CASCADE,
  emp_id      NUMBER(5)      REFERENCES employee_mng(emp_id) ON DELETE CASCADE,
  CONSTRAINT pk_works_in PRIMARY KEY(store_id, emp_id)
);

```

Script Output: Task completed in 0,048 seconds

Table WORKS_IN created.

```

CREATE TABLE category_mng(
  category_id  NUMBER(5)      PRIMARY KEY,
  category_name VARCHAR2(20)  NOT NULL UNIQUE
);

```

```

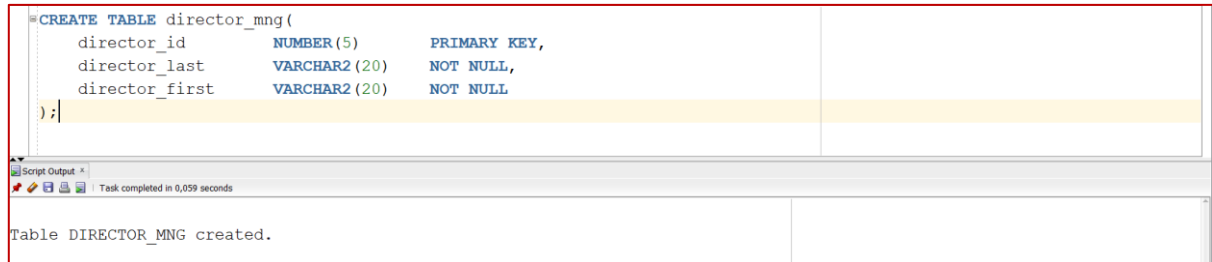
CREATE TABLE category_mng(
  category_id  NUMBER(5)      PRIMARY KEY,
  category_name VARCHAR2(20)  NOT NULL UNIQUE
);

```

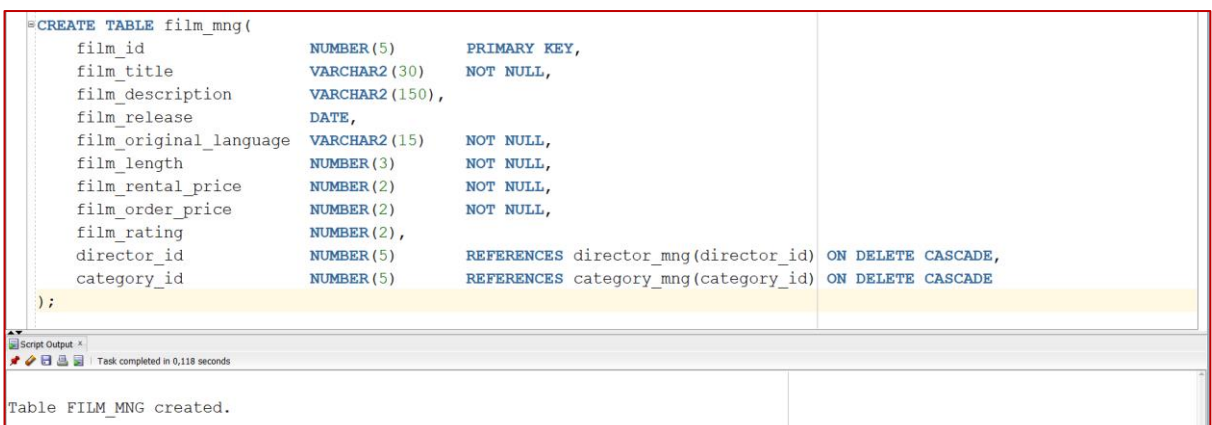
Script Output: Task completed in 0,059 seconds

Table CATEGORY_MNG created.

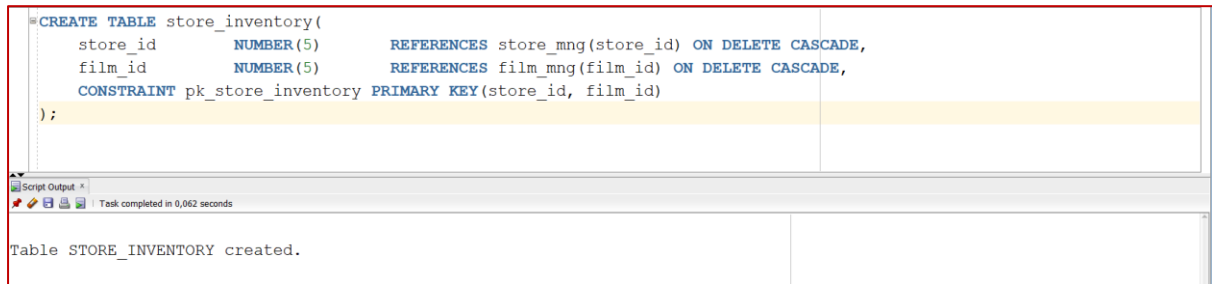
```
CREATE TABLE director_mng(
    director_id    NUMBER(5)    PRIMARY KEY,
    director_last   VARCHAR2(20) NOT NULL,
    director_first  VARCHAR2(20) NOT NULL
);
```



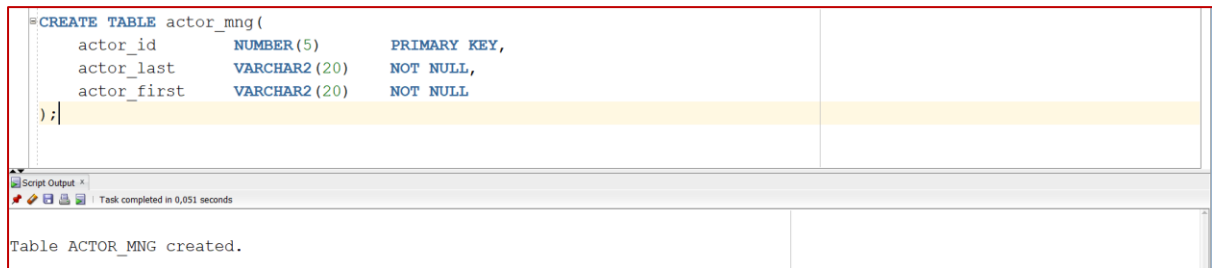
```
CREATE TABLE film_mng(
    film_id        NUMBER(5)    PRIMARY KEY,
    film_title      VARCHAR2(30) NOT NULL,
    film_description VARCHAR2(150),
    film_release    DATE,
    film_original_language VARCHAR2(15) NOT NULL,
    film_length     NUMBER(3)    NOT NULL,
    film_rental_price NUMBER(2)   NOT NULL,
    film_order_price NUMBER(2)   NOT NULL,
    film_rating     NUMBER(2),
    director_id     NUMBER(5)    REFERENCES director_mng(director_id) ON
DELETE CASCADE,
    category_id     NUMBER(5)    REFERENCES category_mng(category_id) ON
DELETE CASCADE
);
```



```
CREATE TABLE store_inventory(
    store_id    NUMBER(5)    REFERENCES store_mng(store_id) ON DELETE
CASCADE,
    film_id     NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE
CASCADE,
    CONSTRAINT pk_store_inventory PRIMARY KEY(store_id, film_id)
);
```



```
CREATE TABLE actor_mng(
    actor_id    NUMBER(5)    PRIMARY KEY,
    actor_last  VARCHAR2(20) NOT NULL,
    actor_first VARCHAR2(20) NOT NULL
);
```



```
CREATE TABLE acts_in(
    film_id    NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE CASCADE,
    actor_id   NUMBER(5)    REFERENCES actor_mng(actor_id) ON DELETE
CASCADE,
    CONSTRAINT pk_acts_in PRIMARY KEY(film_id, actor_id)
);
```

```

CREATE TABLE acts_in(
  film_id    NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE CASCADE,
  actor_id   NUMBER(5)    REFERENCES actor_mng(actor_id) ON DELETE CASCADE,
  CONSTRAINT pk_acts_in PRIMARY KEY(film_id, actor_id)
);

```

Script Output: Task completed in 0,044 seconds

Table ACTS_IN created.

```

CREATE TABLE language_mng(
  language_id  NUMBER(5)    PRIMARY KEY,
  language_name VARCHAR2(15) NOT NULL UNIQUE
);

```

```

CREATE TABLE language_mng(
  language_id  NUMBER(5)    PRIMARY KEY,
  language_name VARCHAR2(15) NOT NULL UNIQUE
);

```

Script Output: Task completed in 0,067 seconds

Table LANGUAGE_MNG created.

```

CREATE TABLE translated_to(
  film_id    NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE CASCADE,
  language_id NUMBER(5)    REFERENCES language_mng(language_id) ON DELETE CASCADE,
  CONSTRAINT pk_translated_to PRIMARY KEY(film_id, language_id)
);

```

```

CREATE TABLE translated_to(
  film_id    NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE CASCADE,
  language_id NUMBER(5)    REFERENCES language_mng(language_id) ON DELETE CASCADE,
  CONSTRAINT pk_translated_to PRIMARY KEY(film_id, language_id)
);

```

Script Output: Task completed in 0,055 seconds

Table TRANSLATED_TO created.

```

CREATE TABLE customer_mng(
  cust_id    NUMBER(5)    PRIMARY KEY,

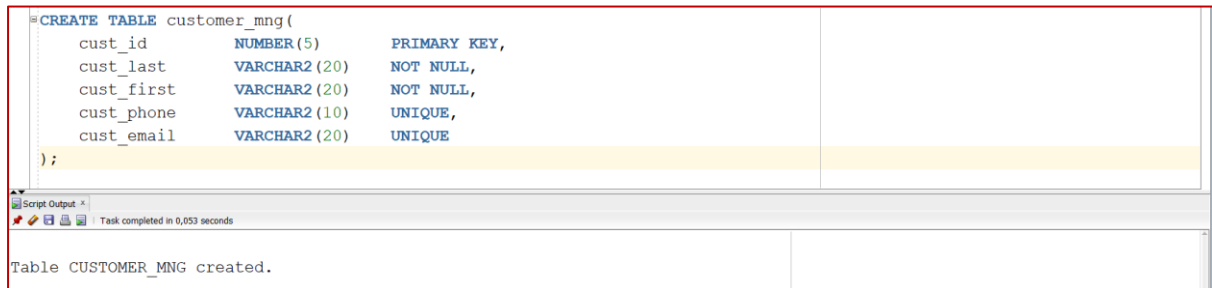
```



```

cust_last    VARCHAR2(20)  NOT NULL,
cust_first   VARCHAR2(20)  NOT NULL,
cust_phone   VARCHAR2(10)  UNIQUE,
cust_email   VARCHAR2(20)  UNIQUE
);

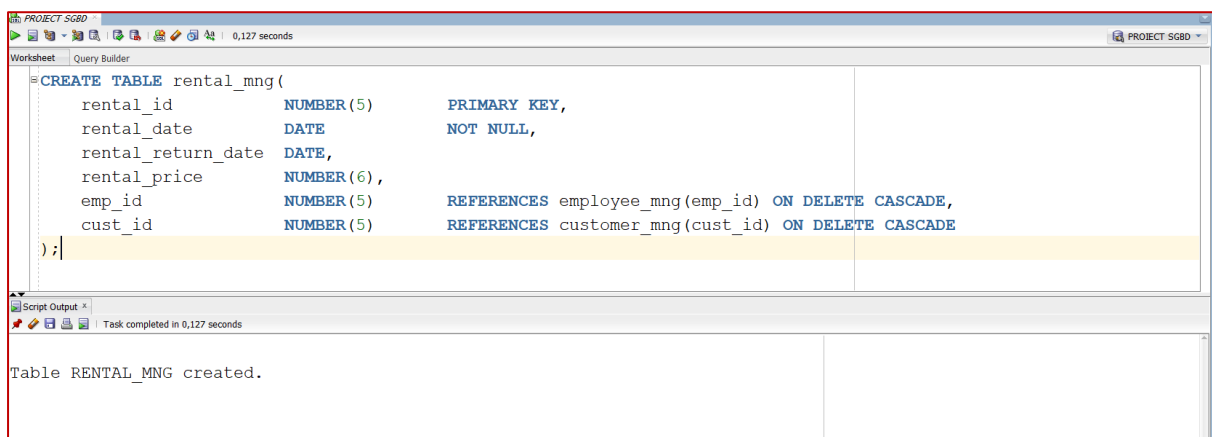
```



```

CREATE TABLE rental_mng(
  rental_id     NUMBER(5)      PRIMARY KEY,
  rental_date    DATE          NOT NULL,
  rental_return_date DATE,
  rental_price   NUMBER(6),
  emp_id        NUMBER(5)      REFERENCES employee_mng(emp_id) ON DELETE
CASCADE,
  cust_id       NUMBER(5)      REFERENCES customer_mng(cust_id) ON DELETE
CASCADE
);

```



```

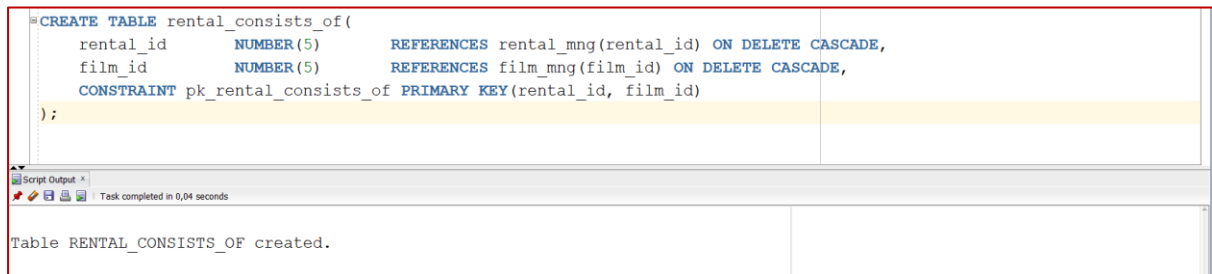
CREATE TABLE rental_consists_of(

```

```

    rental_id    NUMBER(5)    REFERENCES rental_mng(rental_id) ON DELETE
CASCADE,
    film_id      NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE
CASCADE,
    CONSTRAINT pk_rental_consists_of PRIMARY KEY(rental_id, film_id)
);

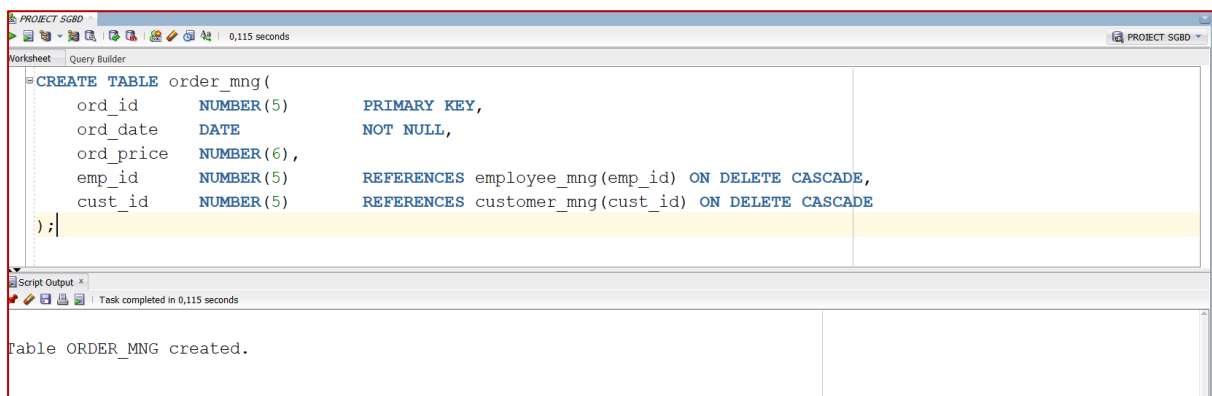
```



```

CREATE TABLE order_mng(
    ord_id    NUMBER(5)    PRIMARY KEY,
    ord_date  DATE          NOT NULL,
    ord_price NUMBER(6),
    emp_id    NUMBER(5)    REFERENCES employee_mng(emp_id) ON DELETE
CASCADE,
    cust_id   NUMBER(5)    REFERENCES customer_mng(cust_id) ON DELETE
CASCADE
);

```



```

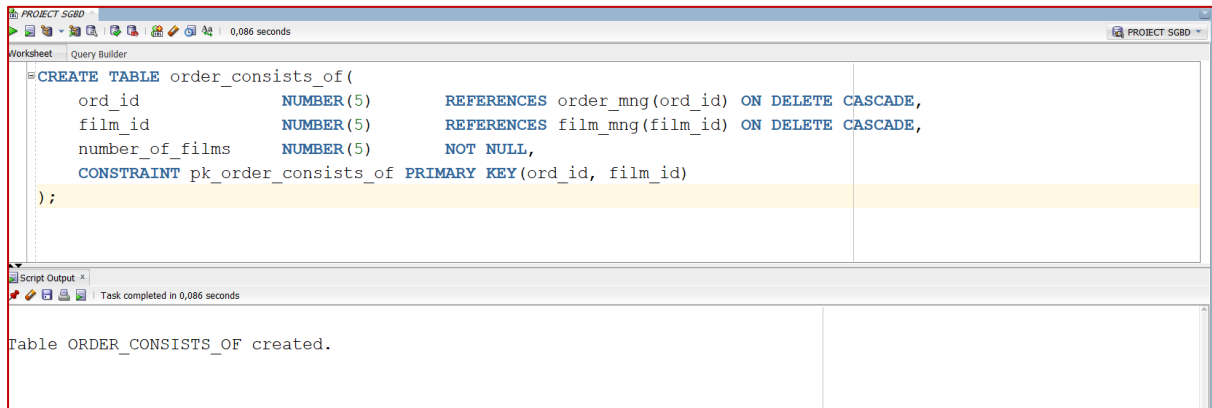
CREATE TABLE order_consists_of(
    ord_id    NUMBER(5)    REFERENCES order_mng(ord_id) ON DELETE CASCADE,

```

```

film_id    NUMBER(5)    REFERENCES film_mng(film_id) ON DELETE CASCADE,
number_of_films    NUMBER(5)    NOT NULL,
CONSTRAINT pk_order_consists_of PRIMARY KEY(ord_id, film_id)
);

```



Inserarea de date coerente:

insert into STORE_MNG

values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage', 102, 'Calea Victoriei nr. 30', 'Bucuresti', null, '030028', '0723000001', 'v_stage@gmail.com');

insert into STORE_MNG

values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage Brasov', 204, 'Strada Republicii nr. 1', 'Brasov', 'Brasov', '500030', '0723000002', 'v_stagebv@gmail.com');

insert into STORE_MNG

values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage Constanta', 300, 'Bulevardul Tomis nr. 27', 'Constanta', 'Constanta', '900178', '0723000003', 'v_stagect@gmail.com');

insert into STORE_MNG

values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage Sibiu', 402, 'Strada Nicolae Balcescu nr. 3', 'Sibiu', 'Sibiu', '550159', '0723000004', 'v_stagesb@gmail.com');

insert into STORE_MNG

values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage Cluj-Napoca', 504, 'Strada Regele Ferdinand nr. 6', 'Cluj-Napoca', 'Cluj', '400110', '0723000005', 'v_stagecj@gmail.com');

STORE_ID	STORE_NAME	MANAGER_ID	ADDRESS	CITY	STATE	ZIP	PHONE	EMAIL
1	2The Velvet Stage	Brasov	204Strada Republicii nr. 1	Brasov	Brasov	500030	0723000002	v stageb
2	3The Velvet Stage	Constanta	300Bulevardul Tomis nr. 27	Constanta	Constanta	900178	0723000003	v stagec
3	4The Velvet Stage	Sibiu	402Strada Nicolae Balcescu nr. 3	Sibiu	Sibiu	550159	0723000004	v stagee
4	5The Velvet Stage	Cluj-Napoca	504Strada Regele Ferdinand nr. 6	Cluj-Napoca	Cluj	400110	0723000005	v stagec
5	1The Velvet Stage	Bucuresti	102Calea Victoriei nr. 30	Bucuresti	(null)	030028	0723000001	v stagec

```
insert into EMPLOYEE_MNG
```

```
values (100,'Iuliu','Carmen','0723100000','iuliu.carmen@gmail.com','06-OCT-2015',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (101,'Radu','Nicolae','0723101000','radu.nicolae@gmail.com','23-DEC-2013',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (102,'Sorin','Daciana','0723102000','sorin.daciana@gmail.com','31-MAR-2020',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (103, 'Leonard', 'Floriana', '0723103000', 'leonard.floriana@gmail.com','17-MAR-2020',2300);
```

```
insert into EMPLOYEE_MNG
```

```
values (104, 'Raluca', 'Natalia', '0723104000', 'raluca.natalia@gmail.com','26-OCT-2021',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (105,'Ciprian','Amalia','0723105000','ciprian.amalia@gmail.com','12-MAR-2019',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (200,'Rozalia','Mariana','0723200000','rozalia.mariana@gmail.com','18-AUG-2017',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (201,'Tereza','Cosmin','0723201000','tereza.cosmin@gmail.com','02-IAN-2018',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (202,'Ghenadie','Remus','0723202000','ghenadie.remus@gmail.com','24-IUN-2020',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (203,'Adam','Cristina','0723203000','adam.cristina@gmail.com','16-SEP-2021',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (204,'Marin','Tudor','0723204000','marin.tudor@gmail.com','13-IUL-2019',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (205,'Apostol','Alexandra','0723205000','apostol.alexandra@gmail.com','12-APR-2017',2300);
```

```
insert into EMPLOYEE_MNG
```

```
values (300,'Dragomir','Ruxandra','0723300000','dragomir.ruxandra@gmail.com','08-MAI-2022',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (301,'Raduica','Miruna','0723301000','raduica.miruna@gmail.com','23-MAR-2018',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (302,'Filip','Cezar','0723302000','filip.cezar@gmail.com','17-OCT-2015',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (303,'Codrin','Elena','0723303000','codrin.elena@gmail.com','13-FEB-2017',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (304,'Isac','Emanuel','0723304000','isac.emanuel@gmail.com','08-MAR-2016',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (305,'Mihail','Ivan','0723305000','mihail.ivan@gmail.com','11-IUL-2020',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (400,'Sebastian','Andrei','0723400000','sebastian.andrei@gmail.com','04-DEC-2021',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (401,'Filimon','Cosmina','0723401000','filimon.cosmina@gmail.com','02-IAN-2018',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (402,'Mihai','Catalina','0723402000','mihai.catalina@gmail.com','01-MAR-2022',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (403,'Horea','Stefan','0723403000','horea.stefan@gmail.com','20-SEP-2020',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (404,'Theodor','Felicia','0723404000','theodor.felicia@gmail.com','13-IUN-2016',3000);
```

```
insert into EMPLOYEE_MNG
```

```
values (405,'Petru','Vlad','0723405000','petru.vlad@gmail.com','14-OCT-2021',2100);
```

```
insert into EMPLOYEE_MNG
```

```
values (500,'Ilie','Adrian','0723500000','ilie.adrian@gmail.com','03-MAR-2020',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (501,'Radu','Claudia','0723501000','radu.claudia@gmail.com','18-APR-2020',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (502,'Pompeiu','Alexandru','0723502000','pompeiu.alexandru@gmail.com','02-SEP-2021',3000);
```

```
insert into EMPLOYEE_MNG
```

```
values (503,'Soare','Eugen','0723503000','soare.eugen@gmail.com','07-MAR-2019',2500);
```

```
insert into EMPLOYEE_MNG
```

```
values (504,'Renata','Anca','0723504000','renata.anca@gmail.com','19-SEP-2019',2800);
```

```
insert into EMPLOYEE_MNG
```

```
values (505,'Neagu','Mara','0723505000','neagu.mara@gmail.com','05-FEB-2021',3000);
```

PROJECT_SGBD EMPLOYEE_MNG

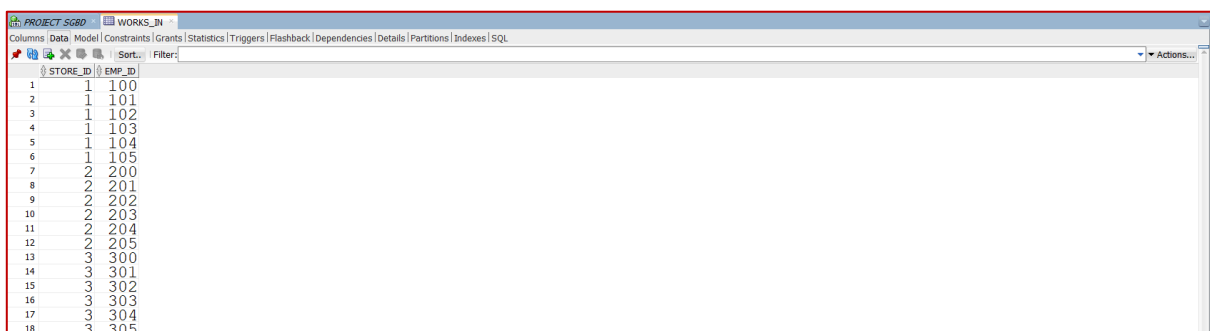
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

<

```
insert into WORKS_IN values (1,100);
```

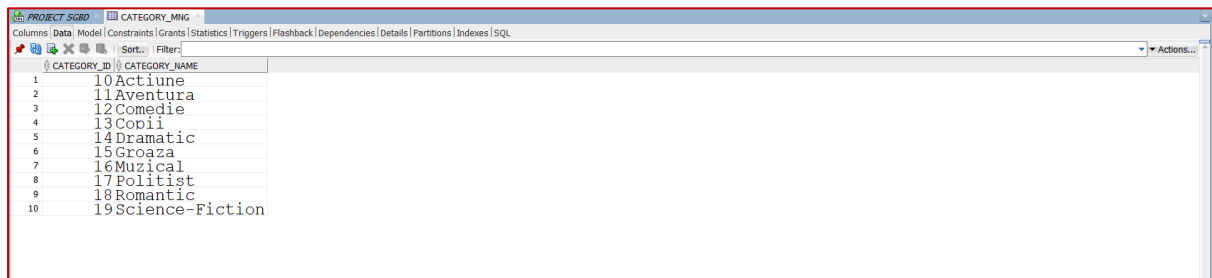
```
insert into WORKS_IN values (1,101);
```

```
insert into WORKS_IN values (1,102);
insert into WORKS_IN values (1,103);
insert into WORKS_IN values (1,104);
insert into WORKS_IN values (1,105);
insert into WORKS_IN values (2,200);
insert into WORKS_IN values (2,201);
insert into WORKS_IN values (2,202);
insert into WORKS_IN values (2,203);
insert into WORKS_IN values (2,204);
insert into WORKS_IN values (2,205);
insert into WORKS_IN values (3,300);
insert into WORKS_IN values (3,301);
insert into WORKS_IN values (3,302);
insert into WORKS_IN values (3,303);
insert into WORKS_IN values (3,304);
insert into WORKS_IN values (3,305);
insert into WORKS_IN values (4,400);
insert into WORKS_IN values (4,401);
insert into WORKS_IN values (4,402);
insert into WORKS_IN values (4,403);
insert into WORKS_IN values (4,404);
insert into WORKS_IN values (4,405);
insert into WORKS_IN values (5,500);
insert into WORKS_IN values (5,501);
insert into WORKS_IN values (5,502);
insert into WORKS_IN values (5,503);
insert into WORKS_IN values (5,504);
insert into WORKS_IN values (5,505);
```



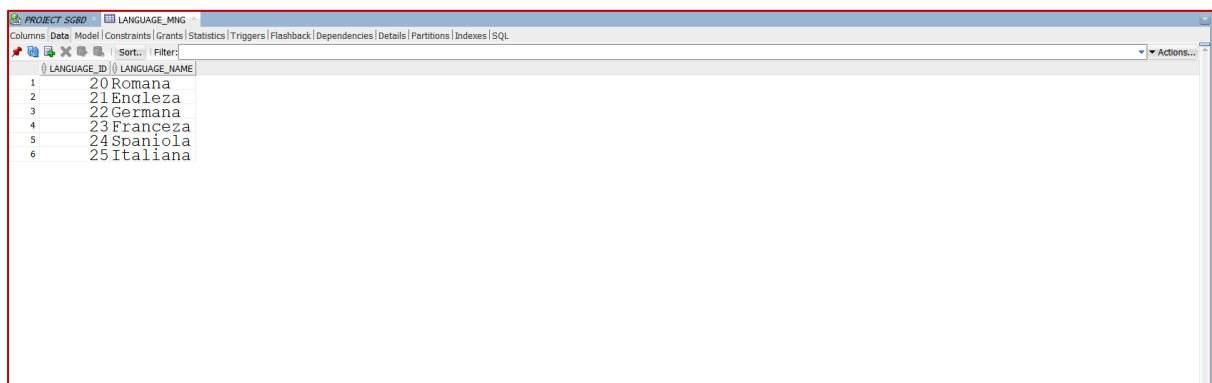
	STORE_ID	EMP_ID
1	1	100
2	1	101
3	1	102
4	1	103
5	1	104
6	1	105
7	2	200
8	2	201
9	2	202
10	2	203
11	2	204
12	2	205
13	3	300
14	3	301
15	3	302
16	3	303
17	3	304
18	3	305

```
insert into CATEGORY_MNG values (10,'Actiune');
insert into CATEGORY_MNG values (11,'Aventura');
insert into CATEGORY_MNG values (12,'Comedie');
insert into CATEGORY_MNG values (13,'Copii');
insert into CATEGORY_MNG values (14,'Dramatic');
insert into CATEGORY_MNG values (15,'Groaza');
insert into CATEGORY_MNG values (16,'Muzical');
insert into CATEGORY_MNG values (17,'Politist');
insert into CATEGORY_MNG values (18,'Romantic');
insert into CATEGORY_MNG values (19,'Science-Fiction');
```



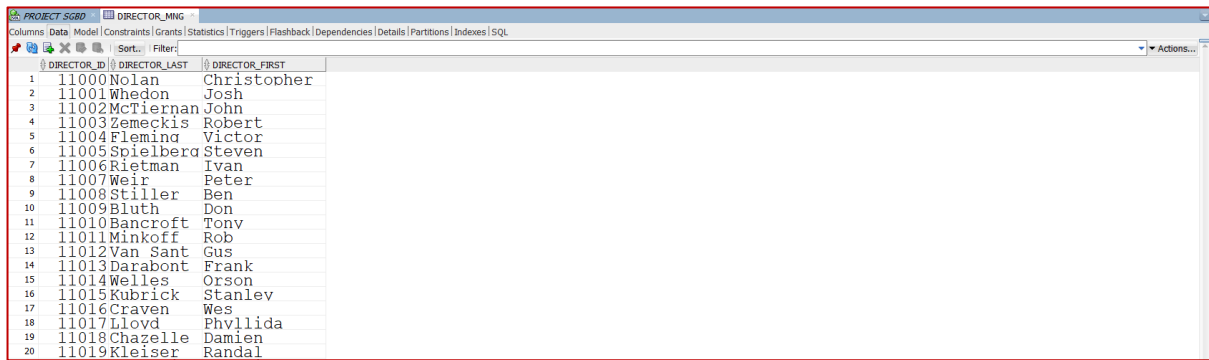
CATEGORY_ID	CATEGORY_NAME
10	Actiune
11	Aventura
12	Comedie
13	Copii
14	Dramatic
15	Groaza
16	Muzical
17	Politist
18	Romantic
19	Science-Fiction

```
insert into LANGUAGE_MNG values (20,'Romana');
insert into LANGUAGE_MNG values (21,'Engleza');
insert into LANGUAGE_MNG values (22,'Germana');
insert into LANGUAGE_MNG values (23,'Franceza');
insert into LANGUAGE_MNG values (24,'Spaniola');
insert into LANGUAGE_MNG values (25,'Italiana');
```



LANGUAGE_ID	LANGUAGE_NAME
20	Romana
21	Engleza
22	Germana
23	Franceza
24	Spaniola
25	Italiana


```
insert into DIRECTOR_MNG values (11000,'Nolan','Christopher');
insert into DIRECTOR_MNG values (11000,'Nolan','Christopher');
insert into DIRECTOR_MNG values (11001,'Whedon','Josh');
insert into DIRECTOR_MNG values (11002,'McTiernan','John');
insert into DIRECTOR_MNG values (11003,'Zemeckis','Robert');
insert into DIRECTOR_MNG values (11004,'Fleming','Victor');
insert into DIRECTOR_MNG values (11005,'Spielberg','Steven');
insert into DIRECTOR_MNG values (11006,'Rietman','Ivan');
insert into DIRECTOR_MNG values (11007,'Weir','Peter');
insert into DIRECTOR_MNG values (11008,'Stiller','Ben');
insert into DIRECTOR_MNG values (11009,'Bluth','Don');
insert into DIRECTOR_MNG values (11010,'Bancroft','Tony');
insert into DIRECTOR_MNG values (11011,'Minkoff','Rob');
insert into DIRECTOR_MNG values (11012,'Van Sant','Gus');
insert into DIRECTOR_MNG values (11013,'Darabont','Frank');
insert into DIRECTOR_MNG values (11014,'Welles','Orson');
insert into DIRECTOR_MNG values (11015,'Kubrick','Stanley');
insert into DIRECTOR_MNG values (11016,'Craven','Wes');
insert into DIRECTOR_MNG values (11017,'Lloyd','Phyllida');
insert into DIRECTOR_MNG values (11018,'Chazelle','Damien');
insert into DIRECTOR_MNG values (11019,'Kleiser','Randal');
insert into DIRECTOR_MNG values (11020,'Tarantino','Quentin');
insert into DIRECTOR_MNG values (11021,'Coppola','Francis Ford');
insert into DIRECTOR_MNG values (11022,'Scorsese','Martin');
insert into DIRECTOR_MNG values (11023,'Linklater','Richard');
insert into DIRECTOR_MNG values (11024,'Wright','Joe');
insert into DIRECTOR_MNG values (11025,'Reiner','Rob');
insert into DIRECTOR_MNG values (11026,'Lucas','Geroge');
insert into DIRECTOR_MNG values (11027,'Wachowski','Lana');
```



	DIRECTOR_ID	DIRECTOR_LAST	DIRECTOR_FIRST
1	11000	Nolan	Christopher
2	11001	Whedon	Josh
3	11002	McTiernan	John
4	11003	Zemeckis	Robert
5	11004	Fleming	Victor
6	11005	Spielberg	Steven
7	11006	Rietman	Ivan
8	11007	Weir	Peter
9	11008	Stiller	Ben
10	11009	Bluth	Don
11	11010	Bancroft	Tony
12	11011	Minkoff	Rob
13	11012	Van Sant	Gus
14	11013	Darabont	Frank
15	11014	Welles	Orson
16	11015	Kubrick	Stanley
17	11016	Craven	Wes
18	11017	Lloyd	Phyllida
19	11018	Chazelle	Damien
20	11019	Kleiser	Randal

insert into FILM_MNG

values (900,'Cavalerul intunecat',

"Cavalerul intunecat" este un film de actiune scris si regizat de Christopher Nolan. Bazat pe benzile desenate cu Batman, filmul este o continuare a filmului "Batman - Inceputuri" (2005). Christian Bale are rolul principal. Batman, personajul principal al filmului, are conflicte cu Joker (Heath Ledger) și o prietenie strânsă cu avocatul său Harvey Dent (Aaron Eckhart).',

'25-IUL-2008','Engleza',152,20,60,9,11000,10);

insert into FILM_MNG

values (901,'Razbunatorii',

'Filmul este primul din franciza "Razbunatorii" si al saselea din Marvel Cinematic Universe (MCU). Cum filmul aduce impreuna toti super-eroii din MCU de pana acum, el reprezinta sfarsitul Fazei 1 a MCU.

Filmul este scris si regizat de Joss Whedon, iar in rolurile principale interpreteaza actorii Robert Downey, Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth, Scarlett Johansson, Jeremy Renner, Tom Hiddleston și Samuel L. Jackson.',

'04-APR-2012','Engleza',143,20,60,8,11001,10);

insert into FILM_MNG

values (902,'Greu de ucis',

"Greu de ucis" este un film de actiune american din 1988 regizat de John McTiernan. Rolurile principale au fost interpretate de actorii Bruce Willis și Alan Rickman. Detectivul NYPD John McClane tocmai ajuns in Los Angeles pentru a petrece Craciunul alaturi de sotia sa este nevoit sa opreasca un grup de teroristi condus de Hans Gruber, care au luat ostatici pe toata lumea din cladirea Nakatomi Plaza.',

'04-IUN-1993','Engleza',132,15,50,8,2,11002,10);

insert into FILM_MNG

values (903,'Inapoi in viitor',

'Michael J. Fox joaca in rolul lui Marty McFly, adolescentul american obisnuit al anilor 1980 care este trimis in mod accidental in 1955 intr-o masina a timpului, al carei inventator este un om de stiinta nebun, interpretat de Christopher Lloyd. In timpul incredibilei calatorii in timp, Marty trebuie sa se asigure ca parintii sai, adolescenti, Crispin Glover si Lea Thompson, se intalnesc si se indragostesc pentru ca el sa poata reveni in viitor.',

'03-IUL-1985','Engleza',116,20,60,8.5,11003,11);

insert into FILM_MNG

values (904,'Vrajitorul din Oz',

'Ca sa-si salveze catelusul Toto de o vecina rea, Dorothy vrea sa plece de acasa, dar se intoarce tocmai cand o tornada loveste ferma din Kansas si o transporta in lumea lui Oz. Pe drumul pavat de caramizi galbene, fetita se imprieteneste cu trei personaje si pornesc impreuna catre orasul unde locuieste Marele Vrajitor, fiecare cu cate o dorinta. Numai ca si aici, ca in orice tara de basm, exista vrajitoare si zane, iar obstacolele nu sunt putine...'

'12-AUG-1939','Engleza',101,30,60,8.6,11004,11);

insert into FILM_MNG

values (905,'Indiana Jones si cautatorii arcei pierdute',

'Renumitul arheolog si expert in stiintele oculte, Dr. Indiana Jones (Harrison Ford) este angajat de catre Guvernul SUA sa gaseasca legendara Arca a legamantului care se presupune ca mai pastreaza ascunse tablelele cu cele 10 Porunci. Din nefericire, agenti ai Gestapo-ului, temuta politie secreta nazista sunt si ei pe urmele Arcai. Un film de exceptie semnat Steven Spielberg care a obtinut nu mai putin de 4 premii Oscar in 1982.',

'12-IUN-1981','Engleza',115,25,50,8.5,11005,11);

insert into FILM_MNG

values (906,'The Truman Show',

'The Truman Show' este un film american SF de comedie din 1998 regizat de Peter Weir. Rolurile principale au fost interpretate de actorii Jim Carrey, Laura Linney, Noah Emmerich. Filmul prezinta viata unui om care initial nu este constient ca intreaga sa viata are loc in centrul unei emisiuni de televiziune urmarita de miliarde de oameni din intreaga lume. Truman devine suspicios de realitatea perceputa si porneste intr-o cautare de descoperire a adevarului despre viata sa.',

'05-IUN-1998','Engleza',103,20,40,8.2,11006,12);

insert into FILM_MNG

values (907,'Vanatorii de fantome',

'Dr. Peter Venkman (Bill Murray), Dr. Raymond Stantz (Dan Aykroyd), Dr. Egon Spengler (Harold Ramis) pierd finantarea guvernamentala pentru proiectul la care lucreaza si se decid sa intre in afaceri pe cont propriu.

Astfel, inarmati cu arme speciale, cei trei pornesc in cautarea si prinderea jefuitorilor de morminte, a fantomelor si a multor alte creaturi ciudat.',

'07-IUN-1984','Engleza',107,20,60,7.6,11007,12);

insert into FILM_MNG

values (908,'Zoolander',

'Comedia regizată de Ben Stiller este o parodie la adresa superficialității din lumea modei și a oamenilor care o pun în mișcare. Nume celebre din lumea modei sunt ridiculizate și se fac trimiteri subtile la diverși creatori care par să fi pierdut contactul cu realitatea. Într-un contrast ironic, familia personajului principal, un model masculin de succes, este prezentată una modestă, de mineri, singurii care par să păstreze contactul cu lumea reală.',

'04-IAN-2001','Engleza',89,20,60,6.3,11008,12);

insert into FILM_MNG

values (909,'Anastasia',

'Candva, nu foarte demult, familia Romanov traia intr-o lume incantatoare a palatelor elegante si a petrecerilor fastuoase. Dar cumplitul Rasputin, intors la Sankt Petersburg din dorinta de a se razbuna, va aprinde focul revolutiei ce va cuprinde intreaga tara, in cele din urma distrugand familia Romanov.

In urma atacului asupra Palatului Regal, singurele care vor supravietui sunt imparateasa Marie si nepoata ei, Anastasia. Urmarii de Rasputin, singura lor speranta este sa prinda ultimul tren...'

'14-NOI-1997','Engleza',94,15,35,8.5,11009,13);

insert into FILM_MNG

values (910,'Mulan',

'Intr-o China plina de prejudecati, o tanara pregatita pentru intalnirea cu petitoarea se face de ras, neaducand onoare familiei. In acelasi timp, hunii ataca teritoriul chinez, iar imparatul decreteaza mobilizarea recrutilor. Trista ca nu si-a multumit tatal, Mulan pleaca in locul acestuia la incorporare, desi pedeapsa pentru fetele care indraznesc sa se incorporeze in

armata este moartea. Mulan este ajutata in drumul ei de un dragon si de un greieras norocos, precum si de un cal falnic.',

'05-IUN-1998','Engleza',88,15,30,8.5,11010,13);

insert into FILM_MNG

values (911,'Regele leu',

'Actiunea se petrece in locul fictiv Pride Lands din Africa, unde un leu este conducatorul tuturor animalelor ca rege. La inceputul filmului, Rafiki, un mandril shaman, il prezinta pe Simba, noul pui nascut al regelui Mufasa si al reginei Sarabi, unei adunari a animalelor la Pride Rock. Intre timp, fratele mai tanar al lui Mufasa, Scar, realizeaza ca nu mai este mostenitorul tronului si pune la cale sa il ucida pe Simba, si pe Mufasa.',

'24-IUN-1994','Engleza',89,15,30,8.9,11011,13);

insert into FILM_MNG

values (912,'Good Will Hunting',

"Good Will Hunting" este un film de drama american din 1997 regizat de Gus Van Sant, avandu-i in distributie pe Matt Damon, Robin Williams, Ben Affleck, Minnie Driver si Stellan Skarsgard. Cu scenariul scris de Affleck si Damon, filmul urmareste povestea muncitorului de douazeci de ani din South Boston Will Hunting, un geniu nerecunoscut, care devine pacientul psihologului Sean Maguire (Robin Williams) dupa ce ataca un politist.',

'05-DEC-1997','Engleza',126,15,40,8.3,11012,14);

insert into FILM_MNG

values (913,'Inchisoarea ingerilor',

"Inchisoarea ingerilor" este un film american din anul 1994, scris si regizat de Frank Darabont. Filmul este o ecranizare dupa nuvela lui Stephen King, Rita Hayworth "Shawshank Redemption", publicata initial in colectia "Anotimpuri diferite" din 1982. Actorii principali sunt Tim Robbins si Morgan Freeman.

Inchisoarea ingerilor este povestea vietii lui Andy Dufresne aflat in inchisoare dupa ce este condamnat pe nedrept pentru uciderea sotiei sale si amantului acesteia.',

'22-SEP-1994','Engleza',142,15,45,9.3,11013,14);

insert into FILM_MNG

values (914,'Cetateanul Kane',

'Un mogul media moare singur in resedinta sa super-luxoasa. Jurnalistii, fostii sai colegi de breasla, incearca sa afle chintesenta vietii acestuia pornind de la ultimul cuvânt rostit pe patul de moarte "rosebud" (boboc de trandafir). Cum va reusi dl. Thomson sa se achite de aceasta sarcina depinde de povestile din viata cetateanului Kane, spuse de cei care au fost o parte importanta sau neglijabila din magnifica sa viata.'

'01-APR-1941','Engleza',119,15,45,8.3,11014,14);

insert into FILM_MNG

values (915,'Stralucirea',

"Stralucirea" este un film horror psihologic din 1980, produs si regizat de Stanley Kubrick. Rolurile principale sunt interpretate de Jack Nicholson, Shelley Duvall și Danny Lloyd. Filmul spune povestea unui scriitor, Jack Torrance, care accepta o slujba de supraveghetor al unui hotel, care pe timp de iarna este absolut pustiu. Fiul sau posedă abilitati de medium si este capabil sa vada lucruri din trecut si din viitor, precum si fantomele care locuiesc in hotel.'

'23-APR-1980','Engleza',119,30,60,8.5,11015,15);

insert into FILM_MNG

values (916,'Cosmar pe strada Elm',

'Nancy are un cosmar teribil legat de o figura inspaimantatoare care ii va patrunde in viata. Figura este cea a lui Freddy Krugger având degete lungi, sub forma de cutite, la mana dreapta. Nancy descopera curand ca si prietenii ei au cosmaruri asemanatoare, cu acelasi personaj. Cand unii dintre cunoscuti incep sa moara, Nancy crede ca singura cale de supravietuire este sa stea treaza. Secretul identitatii ucigasului de copii de pe strada Ulmilor devine prioritatea lui Nancy.'

'16-NOI-1984','Engleza',91,20,50,8.0,11016,15);

insert into FILM_MNG

values (917,'Scream',

'Amestec de umor negru si mister. Un ucigas in serie hartuieste un grup de tineri. Sydney Prescott trece prin momente grele in viata sa: inca nu si-a revenit dupa ce mama ei a fost omorata cu un an in urma, iar acum una din prietenele ei a fost ucisa de un nebun imbracat intr-un costum de Halloween, care o hartuia cu telefoane in mijlocul noptii. Acum Sydney primește apeluri asemanatoare. Cat despre criminal, poate fi oricare din prietenii ei.'

'20-DEC-1996','Engleza',111,15,45,7.7,11016,15);

insert into FILM_MNG

values (918,'Mamma Mia',

'Sophie Sheridan are 20 de ani si locuieste cu mama sa, pe o insula din Grecia. Ea este pe cale sa se casatoreasca si are o singura dorinta, si anume, ca tatal sau sa o conduca la altar. Problema este ca nu stie cine este. Dupa ce gaseste jurnalul Donnei, invita la nunta trei barbati, fiecare dintre ei putand fi tatal sau. Sophie este convinsa ca il va recunoaste, insa, in momentul in care ii vede sosind, nu are nici cea mai mică idee. De aici, o intreaga incurcatura.'

'30-IUN-2008','Engleza',108,20,60,9.0,11017,16);

insert into FILM_MNG

values (919,'La La Land',

"La La Land" este un film american muzical de dragoste dramatic din 2016, scris si regizat de Damien Chazelle. Rolurile principale au fost interpretate de actorii Ryan Gosling, Emma Stone, John Legend și Rosemarie DeWitt. Filmul prezintă o actrita aspiranta si un muzician care se intalnesc si se indragostesc in Los Angeles.'

'09-DEC-2016','Engleza',128,15,50,8.6,11018,16);

insert into FILM_MNG

values (920,'Grease',

"Grease" este un film muzical din 1978, stabilit in anii 1950, regizat de Randal Kleiser. Musicalul spune povestea de dragoste a rebelului Danny Zuko (John Travolta) si a lui Sandy Olsson (Olivia Newton-John). Ambii se cunosc in timpul verii iar la despartire nici unul nu credea ca se vor revedea, pana cand ea decide sa se inscrie la aceeași școală ca prietena ei Frenchy (Didi Conn) ultimul lucru la care se astepta era sa devina colega lui Danny, co-lider al gastii T-Birds.'

'13-IUN-1978','Engleza',110,20,50,7.8,11019,16);

insert into FILM_MNG

values (921,'Pulp Fiction',

"Pulp Fiction" este un film american regizat de Quentin Tarantino, care a scris si scenariul impreuna cu Roger Avary. Filmul este cunoscut pentru dialogul sau spumos, eclectic, combinatia ironica de umor si violenta dar si pentru firul narativ nonlinear. Filmul a fost nominalizat la sapte premii Oscar, printre care pentru cel mai bun film; Tarantino și Avary au castigat premiul pentru cel mai bun scenariu original.'

'14-OCT-1994','Engleza',165,20,70,8.9,11020,17);

insert into FILM_MNG

values (922,'Nasul',

"Nasul" este un film american din 1972, în regia lui Francis Ford Coppola. Este o adaptare după romanul lui Mario Puzo, "Nasul". Acțiunea filmului se petrece în jurul anilor 1940-1950 și prezintă viața fictivă a familiei mafioate Corleone. Distribuția filmului este impresionantă: Marlon Brando, Al Pacino, James Caan, Robert Duvall, Diane Keaton. Filmul a primit foarte multe premii, printre care și 3 Oscaruri și 5 Globuri de Aur.',

'24-MAR-1972','Engleza',175,15,60,9.2,11021,17);

insert into FILM_MNG

values (923,'Baieti buni',

'O capodopera a genului semnată de Martin Scorsese. Henry Hill (Ray Liotta) este un gangster de mână a doua care ia parte la o spargere împreună cu Jimmy Conway (Robert de Niro) și Tommy De Vito (Joe Pesci). Ultimii doi ajung apoi să devină personaje importante ale Mafiei, în timp ce Henry este afectat de succesul lor. Filmul este inspirat de un caz real și povestea lui este conturată în jurul lui Henry, mafiotul care își vede prietenii urcând pe scara ierarhică a crimei.',

'12-SEP-1990','Engleza',146,20,60,9.0,11022,17);

insert into FILM_MNG

values (924,'Înainte de răsărit',

'O studentă franceză, aflată în ultimul an, Celine, interpretată de Julie Deply, întâlnește un băiat american pe nume Jesse (Ethan Hawke) în trenul pe ruta Budapesta - Viena. În Viena coboară amândoi și ies împreună. Idila lor de 14 ore este foarte impresionantă, doar că la sfârșitul călătoriei, vor trebui să se despărță. Sau... nu?',

'27-IAN-1995','Engleza',105,15,35,8.1,11023,18);

insert into FILM_MNG

values (925,'Mândrie și prejudecată',

'Povestea clasică de dragoste este plasată în Anglia sfârșitului secolului 18. Cele cinci surori Bennet - Elizabeth (Keira Knightley), Jane (Rosamund Pike), Lydia, Mary și Kitty – au fost crescute sub strictă îngrijire a mamei lor care se concentrează doar asupra găsirii unor soți fiicelor ei care să le asigure din punct de vedere financiar. Însa, inteligentă și curajoasă Elizabeth se străduiește să-și trăiască viața într-o perspectivă mai largă, încurajată fiind de tatăl bolnav.',

'25-IUL-2005','Engleza',127,25,50,8.6,11024,18);

insert into FILM_MNG

values (926,'Cand Harry o intalneste pe Sally',

'Apreciatul regizor Rob Reiner contureaza aceasta comedie clasica romantica, avandu-i în rolurile principale pe Billy Crystal si Meg Ryan. Una dintre cele mai indragite povesti de dragoste ale tuturor timpurilor, "Cand Harry o întalneste pe Sally" este un film care "scanteiaza de inteligenta si umor" (Boxoffice), prezentand "genul de adevar despre relatii care nu a mai aparut în filme pana acum" (New York)... si nici de la momentul respectiv incoace!'

'12-IUL-1989','Engleza',96,20,45,7.9,11025,18);

insert into FILM_MNG

values (927,'Interstellar: Calatorind prin univers',

"'Interstellar: Calatorind prin univers" este un film stiintifico-fantastic regizat de Christopher Nolan. In rolurile principale sunt Matthew McConaughey, Anne Hathaway, Jessica Chastain, Bill Irwin, Ellen Burstyn si Michael Caine. Filmul prezinta povestea unei echipe de exploratori spatiali care calatoresc printr-o gaura de vierme, in incercarea de a gasi o noua planeta locuabila, dupa ce, pe pamant, viata devenise imposibila.'

'07-NOI-2014','Engleza',169,15,50,8.6,11000,19);

insert into FILM_MNG

values (928,'Star Wars',

'Luke Skywalker (Mark Hamill) locuieste intr-o ferma de pe Planeta Tatooine si, intr-o zi, observa ca robotul de mana a doua primit de la parintii lui este purtatorul unui mesaj venit din spatiul intergalactic, de la o anumita printesa Leia (Carrie Fisher). Aceasta cere disperata ajutorul unui anumit Obi-Wan Kenobi, care il va pune in tema pe Luke cu privire la razboiul dintre gruparea Rebelilor și Imperiul Conducator si energia spirituala numita Forta.'

'25-APR-1977','Engleza',125,30,70,8.7,11026,19);

insert into FILM_MNG

values (929,'Matrix',

'Exista doua lumi: prima, cea de zi cu zi, și cea de-a doua, paralela. Una este vis. Cealalta este Matrix.

Neo (Keanu Reeves) incearca sa afle adevarul despre acea lume despre care a auzit doar in soapta. Dar ce este Matrix? Neo crede ca Morpheus (Laurence Fishburne), o persoana pe care o cunoaste doar din povesti, considerat in acelasi timp cel mai periculos barbat in viata, poate

sa-i dea acest raspuns. Dar adevarul pe care Neo il afla este peste asteptari. Iar viata lui nu va mai fi la fel.'

'31-MAR-1999','Engleza',136,20,50,8.7,11000,19);

FILM_ID	FILM_TITLE	FILM_DESCRIPTION
900	Cavalerul intunecat	"Cavalerul intunecat" este un film de actiune scris si regizat de Chr
901	Razbunatorii	Filmul este primul din franciza "Razbunatorii" si al saselea din Marv
902	Greu de ucis	"Greu de ucis" este un film de actiune american din 1988 regizat de J
903	Inapoi in viitor	Michael J. Fox joaca in rolul lui Marty McFly, adolescentul american
904	Vrajitorul din Oz	Ca sa-si salveze catelusul Toto de o vecina rea, Dorothy vrea sa plec
905	Indiana Jones si cautatorii arcei pierdute	Renumitul arheolog si expert in stiintele oculte, Dr. Indiana Jones (
906	The Truman Show	"The Truman Show" este un film american SF de comedie din 1998 regiza
907	Vanatorii de fantome	Dr. Peter Venkman (Bill Murray), Dr. Raymond Stantz (Dan Aykroyd), Dr
908	Zoolander	Comedia regizata de Ben Stiller este o parodie la adresa superficiali
909	Anastasia	Candva, nu foarte demult, familia Romanov traia intr-o lume incantato
910	Mulan	Intr-o China plina de prejudecati, o tanara pregatita pentru intalnir
911	Regele leu	Actiunea se petrece in locul fictiv Pride Lands din Africa, unde un l
912	Good Will Hunting	"Good Will Hunting" este un film de drama american din 1997 regizat d
913	Inchisoarea inderilor	"Inchisoarea inderilor" este un film american din anul 1994, scris si
914	Cetateanul Kane	Un mozul media moare singur in resedinta sa super-luxoasa. Jurnalisti
915	Stralucirea	"Stralucirea" este un film horror psihologic din 1980, produs si regi
916	Cosmar pe strada Elm	Nancy are un cosmar teribil legat de o figura inspaimantatoare care i
917	Scream	Amestec de umor negru si mister. Un ucigas in serie hartuieste un gru
918	Mamma Mia	Sophie Sheridan are 20 de ani si locuieste cu mama sa, pe o insula di
919	La La Land	"La La Land" este un film american muzical de dragoste dramatic din 2
920	Grease	"Grease" este un film muzical din 1978, stabilit in anii 1950, regiza
921	Pulp Fiction	"Pulp Fiction" este un film american regizat de Quentin Tarantino, ca
922	Nasul	"Nasul" este un film american din 1972, in regia lui Francis Ford Cop
923	Baleti buni	O cabodopera a genului semnata Martin Scorsese. Henry Hill (Ray Liott
924	Imainte de rasarit	O studenta franzuoaica, aflata in ultimul an, Celine, interpretata d
925	Mandrie si prejudecata	Povestea clasica de dragoste este plasata in Anglia sfarsitului secol
926	Cand Harry o intalneste pe Sally	Apresiasiul regizor Rob Reiner contureaza aceasta comedie clasica roma
927	Interstellar: Calatorind prin univers	"Interstellar: Calatorind prin univers" este un film stiintifico-fant
928	Star Wars	Luke Skywalker (Mark Hamill) locuieste intr-o ferma de pe Planeta Tat
929	Matrix	Exista doua lumi: prima, cea de zi cu zi, si cea de-a doua, paralela.

insert into STORE_INVENTORY values (1,900);

insert into STORE_INVENTORY values (1,901);

insert into STORE_INVENTORY values (1,902);

insert into STORE_INVENTORY values (1,903);

insert into STORE_INVENTORY values (1,904);

insert into STORE_INVENTORY values (1,905);

insert into STORE_INVENTORY values (1,906);

insert into STORE_INVENTORY values (1,907);

insert into STORE_INVENTORY values (1,908);

insert into STORE_INVENTORY values (1,909);

insert into STORE_INVENTORY values (1,910);

insert into STORE_INVENTORY values (1,911);

insert into STORE_INVENTORY values (1,912);

insert into STORE_INVENTORY values (1,913);

insert into STORE_INVENTORY values (1,914);

insert into STORE_INVENTORY values (1,915);

```
insert into STORE_INVENTORY values (1,916);
insert into STORE_INVENTORY values (1,917);
insert into STORE_INVENTORY values (1,918);
insert into STORE_INVENTORY values (1,919);
insert into STORE_INVENTORY values (1,920);
insert into STORE_INVENTORY values (1,921);
insert into STORE_INVENTORY values (1,922);
insert into STORE_INVENTORY values (1,923);
insert into STORE_INVENTORY values (1,924);
insert into STORE_INVENTORY values (1,925);
insert into STORE_INVENTORY values (1,926);
insert into STORE_INVENTORY values (1,927);
insert into STORE_INVENTORY values (1,928);
insert into STORE_INVENTORY values (1,929);
insert into STORE_INVENTORY values (2,900);
insert into STORE_INVENTORY values (2,901);
insert into STORE_INVENTORY values (2,902);
insert into STORE_INVENTORY values (2,903);
insert into STORE_INVENTORY values (2,904);
insert into STORE_INVENTORY values (2,905);
insert into STORE_INVENTORY values (2,907);
insert into STORE_INVENTORY values (2,908);
insert into STORE_INVENTORY values (2,909);
insert into STORE_INVENTORY values (2,911);
insert into STORE_INVENTORY values (2,912);
insert into STORE_INVENTORY values (2,913);
insert into STORE_INVENTORY values (2,915);
insert into STORE_INVENTORY values (2,916);
insert into STORE_INVENTORY values (2,917);
insert into STORE_INVENTORY values (2,918);
```

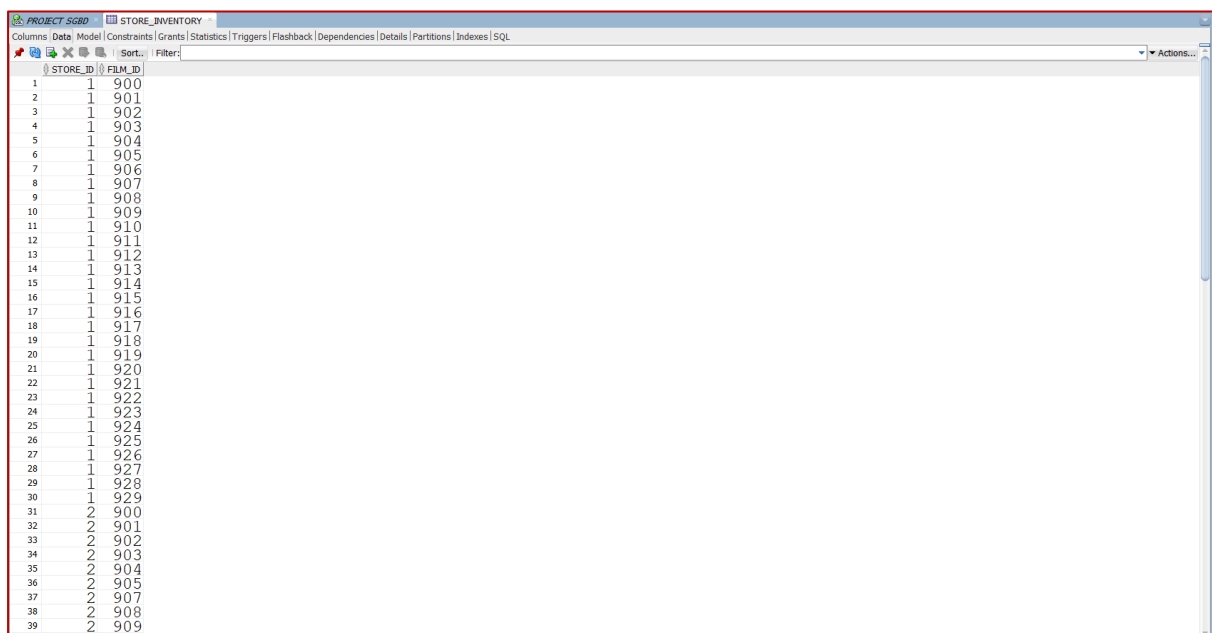
```
insert into STORE_INVENTORY values (2,920);
insert into STORE_INVENTORY values (2,921);
insert into STORE_INVENTORY values (2,922);
insert into STORE_INVENTORY values (2,924);
insert into STORE_INVENTORY values (2,925);
insert into STORE_INVENTORY values (2,926);
insert into STORE_INVENTORY values (2,927);
insert into STORE_INVENTORY values (2,928);
insert into STORE_INVENTORY values (2,929);
insert into STORE_INVENTORY values (3,901);
insert into STORE_INVENTORY values (3,902);
insert into STORE_INVENTORY values (3,903);
insert into STORE_INVENTORY values (3,904);
insert into STORE_INVENTORY values (3,906);
insert into STORE_INVENTORY values (3,907);
insert into STORE_INVENTORY values (3,908);
insert into STORE_INVENTORY values (3,909);
insert into STORE_INVENTORY values (3,910);
insert into STORE_INVENTORY values (3,913);
insert into STORE_INVENTORY values (3,914);
insert into STORE_INVENTORY values (3,915);
insert into STORE_INVENTORY values (3,916);
insert into STORE_INVENTORY values (3,918);
insert into STORE_INVENTORY values (3,919);
insert into STORE_INVENTORY values (3,921);
insert into STORE_INVENTORY values (3,922);
insert into STORE_INVENTORY values (3,925);
insert into STORE_INVENTORY values (3,926);
insert into STORE_INVENTORY values (3,927);
insert into STORE_INVENTORY values (3,928);
```

```
insert into STORE_INVENTORY values (3,929);
insert into STORE_INVENTORY values (4,900);
insert into STORE_INVENTORY values (4,901);
insert into STORE_INVENTORY values (4,903);
insert into STORE_INVENTORY values (4,904);
insert into STORE_INVENTORY values (4,906);
insert into STORE_INVENTORY values (4,907);
insert into STORE_INVENTORY values (4,908);
insert into STORE_INVENTORY values (4,910);
insert into STORE_INVENTORY values (4,911);
insert into STORE_INVENTORY values (4,912);
insert into STORE_INVENTORY values (4,914);
insert into STORE_INVENTORY values (4,915);
insert into STORE_INVENTORY values (4,916);
insert into STORE_INVENTORY values (4,917);
insert into STORE_INVENTORY values (4,918);
insert into STORE_INVENTORY values (4,919);
insert into STORE_INVENTORY values (4,921);
insert into STORE_INVENTORY values (4,923);
insert into STORE_INVENTORY values (4,924);
insert into STORE_INVENTORY values (4,926);
insert into STORE_INVENTORY values (4,927);
insert into STORE_INVENTORY values (4,928);
insert into STORE_INVENTORY values (4,929);
insert into STORE_INVENTORY values (5,900);
insert into STORE_INVENTORY values (5,901);
insert into STORE_INVENTORY values (5,902);
insert into STORE_INVENTORY values (5,903);
insert into STORE_INVENTORY values (5,905);
insert into STORE_INVENTORY values (5,906);
```

```

insert into STORE_INVENTORY values (5,907);
insert into STORE_INVENTORY values (5,908);
insert into STORE_INVENTORY values (5,910);
insert into STORE_INVENTORY values (5,912);
insert into STORE_INVENTORY values (5,913);
insert into STORE_INVENTORY values (5,914);
insert into STORE_INVENTORY values (5,915);
insert into STORE_INVENTORY values (5,916);
insert into STORE_INVENTORY values (5,918);
insert into STORE_INVENTORY values (5,919);
insert into STORE_INVENTORY values (5,920);
insert into STORE_INVENTORY values (5,921);
insert into STORE_INVENTORY values (5,922);
insert into STORE_INVENTORY values (5,925);
insert into STORE_INVENTORY values (5,926);
insert into STORE_INVENTORY values (5,927);
insert into STORE_INVENTORY values (5,928);
insert into STORE_INVENTORY values (5,929);

```



	STORE_ID	FILM_ID
1	1	900
2	1	901
3	1	902
4	1	903
5	1	904
6	1	905
7	1	906
8	1	907
9	1	908
10	1	909
11	1	910
12	1	911
13	1	912
14	1	913
15	1	914
16	1	915
17	1	916
18	1	917
19	1	918
20	1	919
21	1	920
22	1	921
23	1	922
24	1	923
25	1	924
26	1	925
27	1	926
28	1	927
29	1	928
30	1	929
31	2	900
32	2	901
33	2	902
34	2	903
35	2	904
36	2	905
37	2	907
38	2	908
39	2	909

```
insert into TRANSLATED_TO values (900,20);
insert into TRANSLATED_TO values (900,22);
insert into TRANSLATED_TO values (900,23);
insert into TRANSLATED_TO values (900,24);
insert into TRANSLATED_TO values (901,20);
insert into TRANSLATED_TO values (901,23);
insert into TRANSLATED_TO values (901,24);
insert into TRANSLATED_TO values (902,20);
insert into TRANSLATED_TO values (902,22);
insert into TRANSLATED_TO values (902,24);
insert into TRANSLATED_TO values (903,20);
insert into TRANSLATED_TO values (903,22);
insert into TRANSLATED_TO values (903,23);
insert into TRANSLATED_TO values (903,24);
insert into TRANSLATED_TO values (903,25);
insert into TRANSLATED_TO values (904,20);
insert into TRANSLATED_TO values (904,23);
insert into TRANSLATED_TO values (904,25);
insert into TRANSLATED_TO values (905,20);
insert into TRANSLATED_TO values (905,22);
insert into TRANSLATED_TO values (905,23);
insert into TRANSLATED_TO values (905,24);
insert into TRANSLATED_TO values (906,20);
insert into TRANSLATED_TO values (906,24);
insert into TRANSLATED_TO values (906,25);
insert into TRANSLATED_TO values (907,20);
insert into TRANSLATED_TO values (907,22);
insert into TRANSLATED_TO values (907,23);
insert into TRANSLATED_TO values (907,24);
insert into TRANSLATED_TO values (907,25);
```

```
insert into TRANSLATED_TO values (908,20);
insert into TRANSLATED_TO values (908,23);
insert into TRANSLATED_TO values (908,25);
insert into TRANSLATED_TO values (909,20);
insert into TRANSLATED_TO values (909,22);
insert into TRANSLATED_TO values (909,23);
insert into TRANSLATED_TO values (910,20);
insert into TRANSLATED_TO values (910,23);
insert into TRANSLATED_TO values (910,25);
insert into TRANSLATED_TO values (911,20);
insert into TRANSLATED_TO values (911,22);
insert into TRANSLATED_TO values (911,23);
insert into TRANSLATED_TO values (911,24);
insert into TRANSLATED_TO values (911,25);
insert into TRANSLATED_TO values (912,20);
insert into TRANSLATED_TO values (912,23);
insert into TRANSLATED_TO values (912,24);
insert into TRANSLATED_TO values (913,20);
insert into TRANSLATED_TO values (913,22);
insert into TRANSLATED_TO values (913,23);
insert into TRANSLATED_TO values (913,24);
insert into TRANSLATED_TO values (914,20);
insert into TRANSLATED_TO values (914,22);
insert into TRANSLATED_TO values (914,23);
insert into TRANSLATED_TO values (914,24);
insert into TRANSLATED_TO values (914,25);
insert into TRANSLATED_TO values (915,20);
insert into TRANSLATED_TO values (915,24);
insert into TRANSLATED_TO values (915,25);
insert into TRANSLATED_TO values (916,20);
```

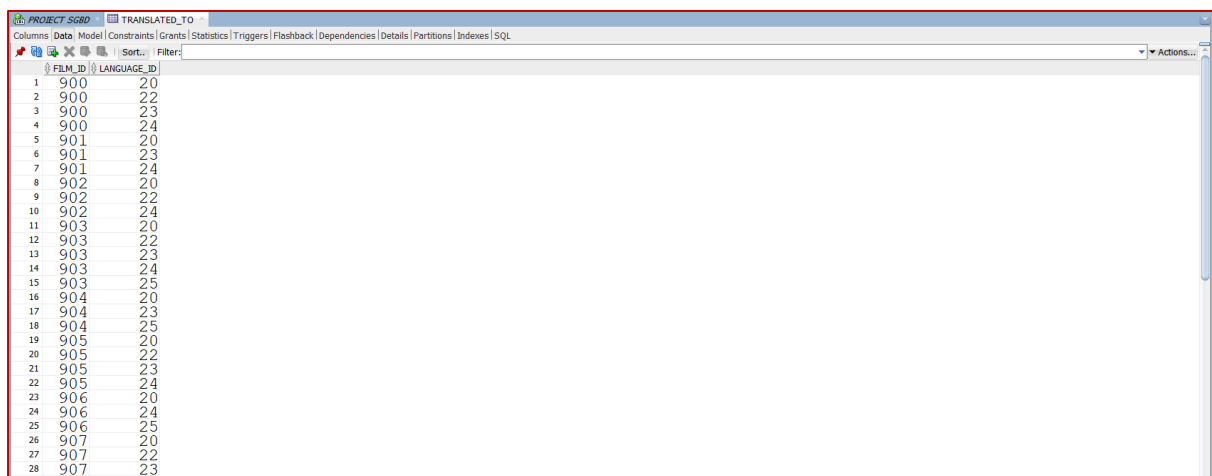


```
insert into TRANSLATED_TO values (916,22);
insert into TRANSLATED_TO values (916,24);
insert into TRANSLATED_TO values (917,20);
insert into TRANSLATED_TO values (917,23);
insert into TRANSLATED_TO values (917,24);
insert into TRANSLATED_TO values (917,25);
insert into TRANSLATED_TO values (918,20);
insert into TRANSLATED_TO values (918,22);
insert into TRANSLATED_TO values (918,23);
insert into TRANSLATED_TO values (918,24);
insert into TRANSLATED_TO values (918,25);
insert into TRANSLATED_TO values (919,20);
insert into TRANSLATED_TO values (919,24);
insert into TRANSLATED_TO values (919,25);
insert into TRANSLATED_TO values (920,20);
insert into TRANSLATED_TO values (920,22);
insert into TRANSLATED_TO values (920,24);
insert into TRANSLATED_TO values (921,20);
insert into TRANSLATED_TO values (921,23);
insert into TRANSLATED_TO values (921,25);
insert into TRANSLATED_TO values (922,20);
insert into TRANSLATED_TO values (922,22);
insert into TRANSLATED_TO values (922,23);
insert into TRANSLATED_TO values (922,24);
insert into TRANSLATED_TO values (922,25);
insert into TRANSLATED_TO values (923,20);
insert into TRANSLATED_TO values (923,24);
insert into TRANSLATED_TO values (923,23);
insert into TRANSLATED_TO values (924,20);
insert into TRANSLATED_TO values (924,23);
```

```

insert into TRANSLATED_TO values (924,24);
insert into TRANSLATED_TO values (925,20);
insert into TRANSLATED_TO values (925,22);
insert into TRANSLATED_TO values (925,24);
insert into TRANSLATED_TO values (926,20);
insert into TRANSLATED_TO values (926,22);
insert into TRANSLATED_TO values (926,23);
insert into TRANSLATED_TO values (926,25);
insert into TRANSLATED_TO values (927,20);
insert into TRANSLATED_TO values (927,22);
insert into TRANSLATED_TO values (927,23);
insert into TRANSLATED_TO values (927,24);
insert into TRANSLATED_TO values (927,25);
insert into TRANSLATED_TO values (928,20);
insert into TRANSLATED_TO values (928,23);
insert into TRANSLATED_TO values (928,24);
insert into TRANSLATED_TO values (928,25);
insert into TRANSLATED_TO values (929,20);
insert into TRANSLATED_TO values (929,22);
insert into TRANSLATED_TO values (929,23);
insert into TRANSLATED_TO values (929,25);

```



	FILM_ID	LANGUAGE_ID
1	900	20
2	900	22
3	900	23
4	900	24
5	901	20
6	901	23
7	901	24
8	902	20
9	902	22
10	902	24
11	903	20
12	903	22
13	903	23
14	903	24
15	903	25
16	904	20
17	904	23
18	904	25
19	905	20
20	905	22
21	905	23
22	905	24
23	906	20
24	906	24
25	906	25
26	907	20
27	907	22
28	907	23

```
insert into ACTOR_MNG values (10000,'Bale','Christian');
insert into ACTOR_MNG values (10001,'Ledger','Heath');
insert into ACTOR_MNG values (10002,'Oldman','Garry');
insert into ACTOR_MNG values (10003,'Caine','Michael');
insert into ACTOR_MNG values (10004,'Freeman','Morgan');
insert into ACTOR_MNG values (10005,'Murphy','Cilian');
insert into ACTOR_MNG values (10006,'Eckhart','Aaron');
insert into ACTOR_MNG values (10007,'Downey Jr.','Robert');
insert into ACTOR_MNG values (10008,'Johansson','Scarlett');
insert into ACTOR_MNG values (10009,'Evans','Chris');
insert into ACTOR_MNG values (10010,'Hemsworth','Chris');
insert into ACTOR_MNG values (10011,'Rufallo','Mark');
insert into ACTOR_MNG values (10012,'Renner','Jeremy');
insert into ACTOR_MNG values (10013,'Hiddleston','Tom');
insert into ACTOR_MNG values (10014,'Jackson','Samuel L. ');
insert into ACTOR_MNG values (10015,'Willis','Bruce');
insert into ACTOR_MNG values (10016,'Rickman','Alan');
insert into ACTOR_MNG values (10017,'Bedelia','Bonnie');
insert into ACTOR_MNG values (10018,'VelJohnson','Reginald');
insert into ACTOR_MNG values (10019,'Fox','Michael G. ');
insert into ACTOR_MNG values (10020,'Lloyd','Christopher');
insert into ACTOR_MNG values (10021,'Thompson','Lea');
insert into ACTOR_MNG values (10022,'Glover','Crispin');
insert into ACTOR_MNG values (10023,'Wilson','Thomas F. ');
insert into ACTOR_MNG values (10024,'Garland','Judy');
insert into ACTOR_MNG values (10025,'Lahr','Bert');
insert into ACTOR_MNG values (10026,'Haley','Jack');
insert into ACTOR_MNG values (10027,'Bolger','Ray');
insert into ACTOR_MNG values (10028,'Hamilton','Margaret');
insert into ACTOR_MNG values (10029,'Morgan','Frank');
```

```
insert into ACTOR_MNG values (10030,'Burke','Billie');
insert into ACTOR_MNG values (10031,'Ford','Harrison');
insert into ACTOR_MNG values (10032,'Allen','Karen');
insert into ACTOR_MNG values (10033,'Molina','Alfred');
insert into ACTOR_MNG values (10034,'Freeman','Paul');
insert into ACTOR_MNG values (10035,'Rhys-Davies','John');
insert into ACTOR_MNG values (10036,'Carrey','Jim');
insert into ACTOR_MNG values (10037,'Harris','Ed');
insert into ACTOR_MNG values (10038,'Linney','Laura');
insert into ACTOR_MNG values (10039,'McElhone','Natascha');
insert into ACTOR_MNG values (10040,'Murray','Bill');
insert into ACTOR_MNG values (10041,'Aykroyd','Dan');
insert into ACTOR_MNG values (10042,'Ramis','Harold');
insert into ACTOR_MNG values (10043,'Weaver','Sigourney');
insert into ACTOR_MNG values (10044,'Hudson','Ernie');
insert into ACTOR_MNG values (10045,'Stiller','Ben');
insert into ACTOR_MNG values (10046,'Wilson','Owen');
insert into ACTOR_MNG values (10047,'Ferrell','Will');
insert into ACTOR_MNG values (10048,'Taylor','Christine');
insert into ACTOR_MNG values (10049,'Jovovitch','Milia');
insert into ACTOR_MNG values (10050,'Vaughn','Vince');
insert into ACTOR_MNG values (10051,'Ryan','Meg');
insert into ACTOR_MNG values (10052,'Cusack','John');
insert into ACTOR_MNG values (10053,'Dunst','Kirsten');
insert into ACTOR_MNG values (10054,'Lansbury','Angela');
insert into ACTOR_MNG values (10055,'Wen','Ming-Na');
insert into ACTOR_MNG values (10056,'Murphy','Eddie');
insert into ACTOR_MNG values (10146,'Hong','James');
insert into ACTOR_MNG values (10147,'Wong','B. D. ');
insert into ACTOR_MNG values (10148,'Fierstein','Harvey');
```

```

insert into ACTOR_MNG values (10057,'Irons','Jeremy');
insert into ACTOR_MNG values (10058,'Jones','James Earl');
insert into ACTOR_MNG values (10059,'Broderick','Matthew');
insert into ACTOR_MNG values (10060,'Atkinson','Rowan');
insert into ACTOR_MNG values (10061,'Lane','Nathan');
insert into ACTOR_MNG values (10062,'Sabella','Ernie');
insert into ACTOR_MNG values (10063,'Moira','Kelly');
insert into ACTOR_MNG values (10064,'Damon','Matt');
insert into ACTOR_MNG values (10065,'Affleck','Ben');
insert into ACTOR_MNG values (10066,'Williams','Robin');
insert into ACTOR_MNG values (10067,'Driver','Minnie');
insert into ACTOR_MNG values (10068,'Robins','Tim');
insert into ACTOR_MNG values (10069,'Gunton','Bob');
insert into ACTOR_MNG values (10070,'Brown','Clancy');
insert into ACTOR_MNG values (10071,'Sadler','William');
insert into ACTOR_MNG values (10072,'Welles','Orson');
insert into ACTOR_MNG values (10073,'Cotten','Joseph');
insert into ACTOR_MNG values (10074,'Comingore','Dorothy');
insert into ACTOR_MNG values (10075,'Alland','William');
insert into ACTOR_MNG values (10076,'Sloane','Everett');
insert into ACTOR_MNG values (10077,'Nicholson','Jack');
insert into ACTOR_MNG values (10078,'Duvall','Shelley');
insert into ACTOR_MNG values (10079,'Lloyd','Danny');
insert into ACTOR_MNG values (10080,'Crothers','Scatman');
insert into ACTOR_MNG values (10081,'Englund','Robert');
insert into ACTOR_MNG values (10082,'Langenkamp','Heather');
insert into ACTOR_MNG values (10083,'Depp','Johnny');
insert into ACTOR_MNG values (10084,'Saxon','John');
insert into ACTOR_MNG values (10085,'Wyss','Amanda');
insert into ACTOR_MNG values (10086,'Campbell','Neve');

```

```

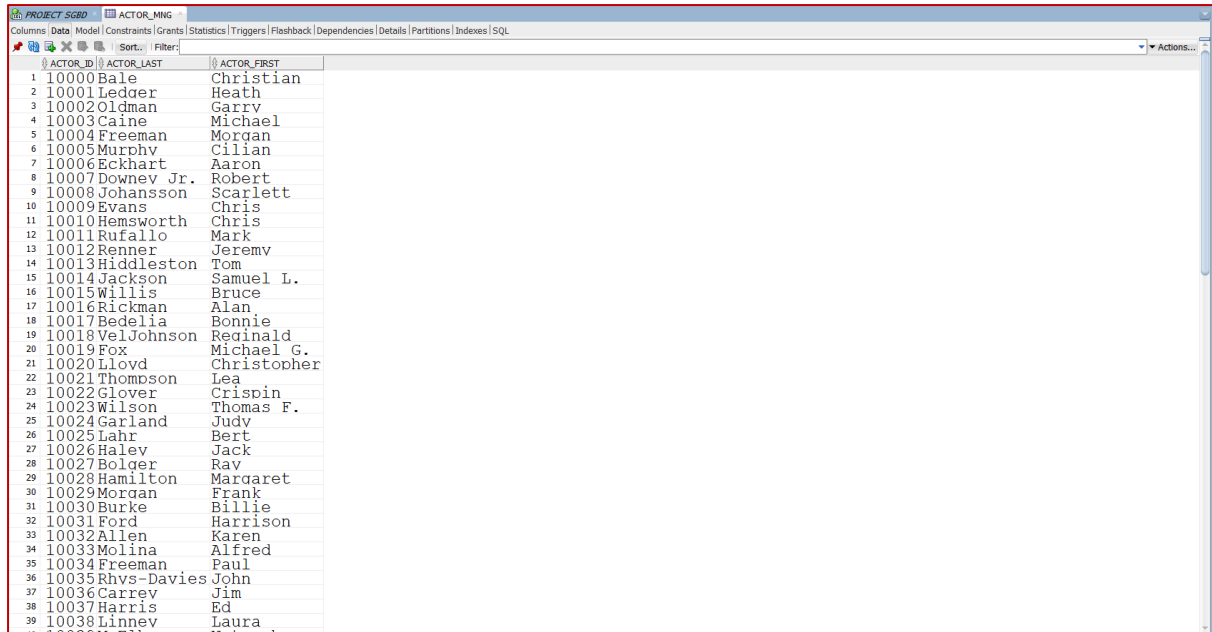
insert into ACTOR_MNG values (10087,'Arquette','David');
insert into ACTOR_MNG values (10088,'Cox','Courtney');
insert into ACTOR_MNG values (10089,'Ulrich','Skeet');
insert into ACTOR_MNG values (10090,'Lillard','Matthew');
insert into ACTOR_MNG values (10091,'Barrymore','Drew');
insert into ACTOR_MNG values (10092,'Streep','Meryl');
insert into ACTOR_MNG values (10093,'Seyfried','Amanda');
insert into ACTOR_MNG values (10094,'Brosnan','Pierce');
insert into ACTOR_MNG values (10095,'Firth','Colin');
insert into ACTOR_MNG values (10096,'Baranski','Christine');
insert into ACTOR_MNG values (10097,'Skarsgard','Stellan');
insert into ACTOR_MNG values (10098,'Walters','Julie');
insert into ACTOR_MNG values (10099,'Stone','Emma');
insert into ACTOR_MNG values (10100,'Gosling','Ryan');
insert into ACTOR_MNG values (10101,'Simmons','J. K. ');
insert into ACTOR_MNG values (10102,'Travolta','John');
insert into ACTOR_MNG values (10103,'Newton-John','Olivia');
insert into ACTOR_MNG values (10104,'Pfeiffer','Michelle');
insert into ACTOR_MNG values (10105,'Biehn','Michael');
insert into ACTOR_MNG values (10106,'Conaway','Jeff');
insert into ACTOR_MNG values (10107,'Thurman','Uma');
insert into ACTOR_MNG values (10149,'Roth','Tim');
insert into ACTOR_MNG values (10108,'Keitel','Harvey');
insert into ACTOR_MNG values (10109,'Pacino','Al');
insert into ACTOR_MNG values (10110,'Brando','Marlon');
insert into ACTOR_MNG values (10111,'De Niro','Robert');
insert into ACTOR_MNG values (10112,'Keaton','Diane');
insert into ACTOR_MNG values (10113,'Caan','James');
insert into ACTOR_MNG values (10114,'Biehn','Michael');
insert into ACTOR_MNG values (10115,'Liotta','Ray');

```

```

insert into ACTOR_MNG values (10116,'Pesci','Joe');
insert into ACTOR_MNG values (10117,'Bracco','Lorraine');
insert into ACTOR_MNG values (10118,'Sorvino','Paul');
insert into ACTOR_MNG values (10119,'Hawke','Ethan');
insert into ACTOR_MNG values (10120,'Delpy','Julie');
insert into ACTOR_MNG values (10121,'Knightley','Keira');
insert into ACTOR_MNG values (10122,'Macfadyen','Matthew');
insert into ACTOR_MNG values (10123,'Pike','Rosemund');
insert into ACTOR_MNG values (10124,'Woods','Simon');
insert into ACTOR_MNG values (10125,'Mulligan','Carey');
insert into ACTOR_MNG values (10126,'Riley','Talulah');
insert into ACTOR_MNG values (10127,'Malone','Jena');
insert into ACTOR_MNG values (10128,'Sutherland','Donald');
insert into ACTOR_MNG values (10129,'Crystal','Billie');
insert into ACTOR_MNG values (10130,'Kirby','Bruno');
insert into ACTOR_MNG values (10131,'Fisher','Carrie');
insert into ACTOR_MNG values (10132,'McConaughey','Matthew');
insert into ACTOR_MNG values (10133,'Hathaway','Anne');
insert into ACTOR_MNG values (10134,'Chastain','Jessica');
insert into ACTOR_MNG values (10135,'Foy','Mackenzie');
insert into ACTOR_MNG values (10136,'Chalamet','Timothée');
insert into ACTOR_MNG values (10137,'Hamill','Mark');
insert into ACTOR_MNG values (10138,'Guinness','Alec');
insert into ACTOR_MNG values (10139,'Mayhew','Peter');
insert into ACTOR_MNG values (10140,'Prowse','David');
insert into ACTOR_MNG values (10141,'Reeves','Keanu');
insert into ACTOR_MNG values (10142,'Moss','Carrie-Anne');
insert into ACTOR_MNG values (10143,'Fishburne','Lawrence');
insert into ACTOR_MNG values (10144,'Weaving','Hugo');
insert into ACTOR_MNG values (10145,'Pantoliano','Joe');

```



	ACTOR_ID	ACTOR_LAST	ACTOR_FIRST
1	10000	Bale	Christian
2	10001	Ledger	Heath
3	10002	Oldman	Garry
4	10003	Caine	Michael
5	10004	Freeman	Morgan
6	10005	Murphy	Cillian
7	10006	Eckhart	Aaron
8	10007	Downey Jr.	Robert
9	10008	Johansson	Scarlett
10	10009	Evans	Chris
11	10010	Hemsworth	Chris
12	10011	Rufallo	Mark
13	10012	Renner	Jeremy
14	10013	Hiddleston	Tom
15	10014	Jackson	Samuel L.
16	10015	Willis	Bruce
17	10016	Rickman	Alan
18	10017	Bedelia	Bonnie
19	10018	VelJohnson	Reginald
20	10019	Fox	Michael G.
21	10020	Lloyd	Christopher
22	10021	Thompson	Lea
23	10022	Glover	Crispin
24	10023	Wilson	Thomas F.
25	10024	Garland	Judy
26	10025	Lahr	Bert
27	10026	Haley	Jack
28	10027	Bolger	Rav
29	10028	Hamilton	Margaret
30	10029	Morgan	Frank
31	10030	Burke	Billie
32	10031	Ford	Harrison
33	10032	Allen	Karen
34	10033	Molina	Alfred
35	10034	Freeman	Paul
36	10035	Rhys-Davies	John
37	10036	Carrey	Jim
38	10037	Harris	Ed
39	10038	Linney	Laura

insert into ACTS_IN values (900,10000);

insert into ACTS_IN values (900,10001);

insert into ACTS_IN values (900,10002);

insert into ACTS_IN values (900,10003);

insert into ACTS_IN values (900,10004);

insert into ACTS_IN values (900,10005);

insert into ACTS_IN values (900,10006);

insert into ACTS_IN values (901,10007);

insert into ACTS_IN values (901,10008);

insert into ACTS_IN values (901,10009);

insert into ACTS_IN values (901,10010);

insert into ACTS_IN values (901,10011);

insert into ACTS_IN values (901,10012);

insert into ACTS_IN values (901,10013);

insert into ACTS_IN values (901,10014);

insert into ACTS_IN values (902,10015);

insert into ACTS_IN values (902,10016);

insert into ACTS_IN values (902,10017);


```
insert into ACTS_IN values (902,10018);
insert into ACTS_IN values (902,10057);
insert into ACTS_IN values (903,10019);
insert into ACTS_IN values (903,10020);
insert into ACTS_IN values (903,10021);
insert into ACTS_IN values (903,10022);
insert into ACTS_IN values (903,10023);
insert into ACTS_IN values (904,10024);
insert into ACTS_IN values (904,10025);
insert into ACTS_IN values (904,10026);
insert into ACTS_IN values (904,10027);
insert into ACTS_IN values (904,10028);
insert into ACTS_IN values (904,10029);
insert into ACTS_IN values (904,10030);
insert into ACTS_IN values (905,10031);
insert into ACTS_IN values (905,10032);
insert into ACTS_IN values (905,10033);
insert into ACTS_IN values (905,10034);
insert into ACTS_IN values (905,10035);
insert into ACTS_IN values (906,10036);
insert into ACTS_IN values (906,10037);
insert into ACTS_IN values (906,10038);
insert into ACTS_IN values (906,10039);
insert into ACTS_IN values (907,10040);
insert into ACTS_IN values (907,10041);
insert into ACTS_IN values (907,10042);
insert into ACTS_IN values (907,10043);
insert into ACTS_IN values (907,10044);
insert into ACTS_IN values (908,10045);
insert into ACTS_IN values (908,10046);
```

```
insert into ACTS_IN values (908,10047);
insert into ACTS_IN values (908,10048);
insert into ACTS_IN values (908,10049);
insert into ACTS_IN values (908,10050);
insert into ACTS_IN values (909,10051);
insert into ACTS_IN values (909,10052);
insert into ACTS_IN values (909,10053);
insert into ACTS_IN values (909,10054);
insert into ACTS_IN values (910,10055);
insert into ACTS_IN values (910,10056);
insert into ACTS_IN values (910,10146);
insert into ACTS_IN values (910,10147);
insert into ACTS_IN values (910,10148);
insert into ACTS_IN values (911,10057);
insert into ACTS_IN values (911,10058);
insert into ACTS_IN values (911,10059);
insert into ACTS_IN values (911,10060);
insert into ACTS_IN values (911,10061);
insert into ACTS_IN values (911,10062);
insert into ACTS_IN values (911,10063);
insert into ACTS_IN values (911,10064);
insert into ACTS_IN values (912,10065);
insert into ACTS_IN values (912,10066);
insert into ACTS_IN values (912,10067);
insert into ACTS_IN values (913,10068);
insert into ACTS_IN values (913,10004);
insert into ACTS_IN values (913,10069);
insert into ACTS_IN values (913,10070);
insert into ACTS_IN values (913,10071);
insert into ACTS_IN values (914,10072);
```

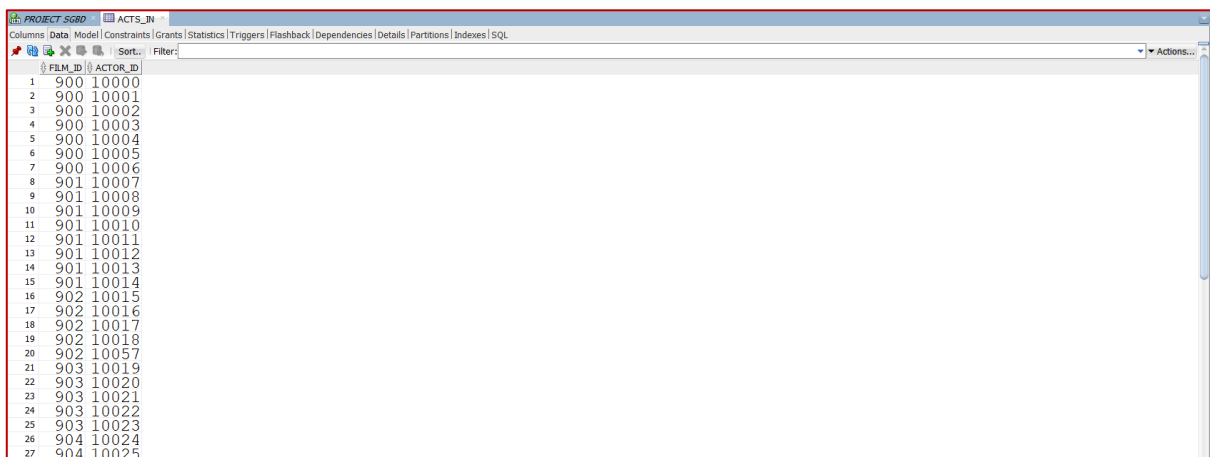
```
insert into ACTS_IN values (914,10073);
insert into ACTS_IN values (914,10074);
insert into ACTS_IN values (914,10075);
insert into ACTS_IN values (914,10076);
insert into ACTS_IN values (914,10077);
insert into ACTS_IN values (915,10078);
insert into ACTS_IN values (915,10079);
insert into ACTS_IN values (915,10080);
insert into ACTS_IN values (916,10081);
insert into ACTS_IN values (916,10082);
insert into ACTS_IN values (916,10083);
insert into ACTS_IN values (916,10084);
insert into ACTS_IN values (916,10085);
insert into ACTS_IN values (917,10086);
insert into ACTS_IN values (917,10087);
insert into ACTS_IN values (917,10088);
insert into ACTS_IN values (917,10089);
insert into ACTS_IN values (917,10090);
insert into ACTS_IN values (917,10091);
insert into ACTS_IN values (918,10092);
insert into ACTS_IN values (918,10093);
insert into ACTS_IN values (918,10094);
insert into ACTS_IN values (918,10095);
insert into ACTS_IN values (918,10096);
insert into ACTS_IN values (918,10097);
insert into ACTS_IN values (918,10098);
insert into ACTS_IN values (919,10099);
insert into ACTS_IN values (919,10100);
insert into ACTS_IN values (919,10101);
insert into ACTS_IN values (920,10102);
```

```
insert into ACTS_IN values (920,10103);
insert into ACTS_IN values (920,10104);
insert into ACTS_IN values (920,10105);
insert into ACTS_IN values (920,10106);
insert into ACTS_IN values (921,10102);
insert into ACTS_IN values (921,10014);
insert into ACTS_IN values (921,10107);
insert into ACTS_IN values (921,10015);
insert into ACTS_IN values (921,10149);
insert into ACTS_IN values (921,10108);
insert into ACTS_IN values (922,10109);
insert into ACTS_IN values (922,10110);
insert into ACTS_IN values (922,10111);
insert into ACTS_IN values (922,10112);
insert into ACTS_IN values (922,10113);
insert into ACTS_IN values (922,10114);
insert into ACTS_IN values (923,10115);
insert into ACTS_IN values (923,10111);
insert into ACTS_IN values (923,10116);
insert into ACTS_IN values (923,10117);
insert into ACTS_IN values (923,10118);
insert into ACTS_IN values (924,10119);
insert into ACTS_IN values (924,10120);
insert into ACTS_IN values (925,10121);
insert into ACTS_IN values (925,10122);
insert into ACTS_IN values (925,10123);
insert into ACTS_IN values (925,10124);
insert into ACTS_IN values (925,10125);
insert into ACTS_IN values (925,10126);
insert into ACTS_IN values (925,10127);
```

```

insert into ACTS_IN values (925,10128);
insert into ACTS_IN values (926,10129);
insert into ACTS_IN values (926,10051);
insert into ACTS_IN values (926,10130);
insert into ACTS_IN values (926,10131);
insert into ACTS_IN values (927,10132);
insert into ACTS_IN values (927,10133);
insert into ACTS_IN values (927,10134);
insert into ACTS_IN values (927,10135);
insert into ACTS_IN values (927,10136);
insert into ACTS_IN values (927,10003);
insert into ACTS_IN values (928,10137);
insert into ACTS_IN values (928,10031);
insert into ACTS_IN values (928,10131);
insert into ACTS_IN values (928,10138);
insert into ACTS_IN values (928,10139);
insert into ACTS_IN values (928,10140);
insert into ACTS_IN values (929,10141);
insert into ACTS_IN values (929,10142);
insert into ACTS_IN values (929,10143);
insert into ACTS_IN values (929,10144);
insert into ACTS_IN values (929,10145);

```



	FILM_ID	ACTOR_ID
1	900	10000
2	900	10001
3	900	10002
4	900	10003
5	900	10004
6	900	10005
7	900	10006
8	901	10007
9	901	10008
10	901	10009
11	901	10010
12	901	10011
13	901	10012
14	901	10013
15	901	10014
16	902	10015
17	902	10016
18	902	10017
19	902	10018
20	902	10019
21	903	10020
22	903	10021
23	903	10022
24	903	10023
25	903	10024
26	904	10025
27	904	10025

```
insert into CUSTOMER_MNG
values (1000,'Petre','Veronica','0732100000',null);

insert into CUSTOMER_MNG
values (1001,'Emil','Alexandru','0732100100','emil.alexandru@gmail.com');

insert into CUSTOMER_MNG
values (1002,'Anton','Iacob',null,null);

insert into CUSTOMER_MNG
values (1003,'Antoniou','Isabela','0732100300','antoniou.isabela@gmail.com');

insert into CUSTOMER_MNG
values (1004,'Toma','Alin',null,'toma.alin@gmail.com');

insert into CUSTOMER_MNG
values (1005,'Eduard','Carmen','0732100500',null);

insert into CUSTOMER_MNG
values (1006,'Augustin','Iuliana','0732100600','augustin.iuliana@gmail.com');

insert into CUSTOMER_MNG
values (1007,'Simion','Constantin',null,null);

insert into CUSTOMER_MNG
values (1008,'Emanuel','Dragos','0732100800','emanuel.dragos@gmail.com');

insert into CUSTOMER_MNG
values (1009,'Gheorghe','Liviu','0732100900',null);

insert into CUSTOMER_MNG
values (1010,'Leonard','Emil','0732101000','leonard.emil@gmail.com');

insert into CUSTOMER_MNG
values (1011,'Florian','Cezar','0732101100',null);

insert into CUSTOMER_MNG
values (1012,'Dinu','Valentin',null,null);

insert into CUSTOMER_MNG
values (1013,'Iosif','Stefan','0732101300','iosif.stefan@gmail.com');

insert into CUSTOMER_MNG
values (1014,'Iancu','Serghei',null,'iancu.serghei@gmail.com');
```

```
insert into CUSTOMER_MNG
values (1015,'Ioan','Paul','0732101500',null);

insert into CUSTOMER_MNG
values (1016,'Adrian','Petru','0732101600','adrian.petru@gmail.com');

insert into CUSTOMER_MNG
values (1017,'Marius','Albert',null,null);

insert into CUSTOMER_MNG
values (1018,'Gavril','Vlad',null,'gavril.vlad@gmail.com');

insert into CUSTOMER_MNG
values (1019,'Amalia','Camelia',null,null);

insert into CUSTOMER_MNG
values (1020,'Enache','Andreea','0732102000',null);

insert into CUSTOMER_MNG
values (1021,'Renata','Maria','0732102100','renata.maria@gmail.com');

insert into CUSTOMER_MNG
values (1022,'Magda','Mihaela','0732102200',null);

insert into CUSTOMER_MNG
values (1023,'Solomon','Constantina','0732102300',null);

insert into CUSTOMER_MNG
values (1024,'Lungu','Gabriela',null,null);

insert into CUSTOMER_MNG
values (1025,'Stan','Larisa',null,'stan.larisa@gmail.com');

insert into CUSTOMER_MNG
values (1026,'Antonescu','Monica','0732102600','antonescu.monica@gmail.com');

insert into CUSTOMER_MNG
values (1027,'David','Amalia','0732102700',null);

insert into CUSTOMER_MNG
values (1028,'Vladimirescu','Antonia',null,null);

insert into CUSTOMER_MNG
values (1029,'Ungur','Bianca','0732102900','ungur.bianca@gmail.com');
```

```

insert into CUSTOMER_MNG
values (1030,'Petran','Elena','0732103000',null);

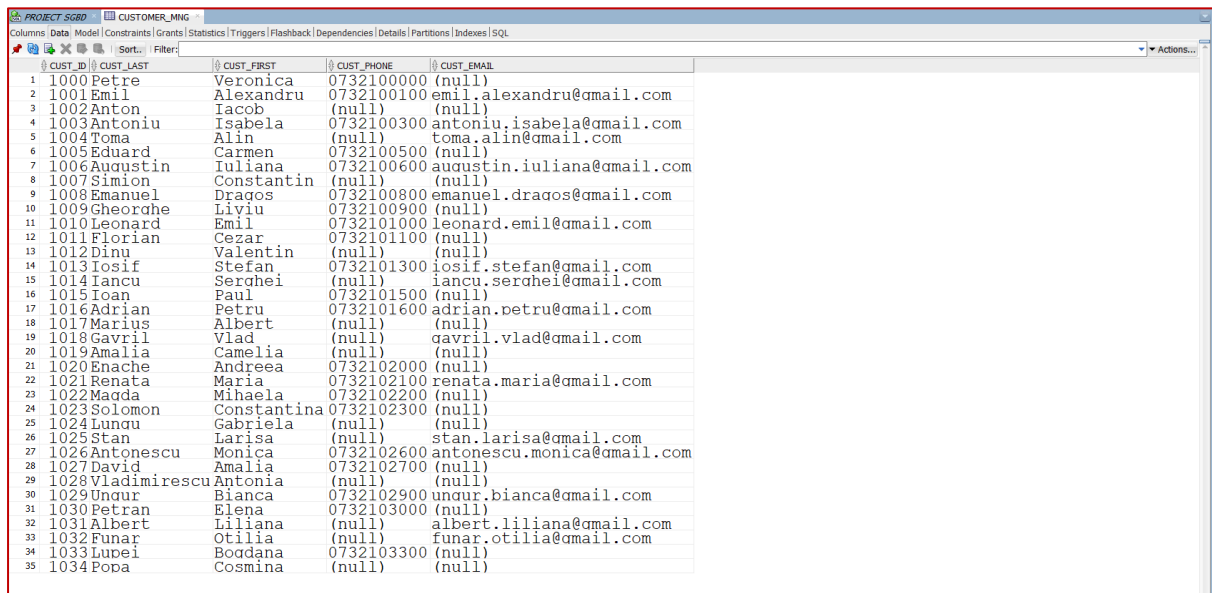
insert into CUSTOMER_MNG
values (1031,'Albert','Liliana',null,'albert.liliana@gmail.com');

insert into CUSTOMER_MNG
values (1032,'Funar','Otilia',null,'funar.otilia@gmail.com');

insert into CUSTOMER_MNG
values (1033,'Lupei','Bogdana','0732103300',null);

insert into CUSTOMER_MNG
values (1034,'Popa','Cosmina',null,null);

```



The screenshot shows a database management tool interface with a table named 'CUSTOMER_MNG'. The table has five columns: CUST_ID, CUST_LAST, CUST_FIRST, CUST_PHONE, and CUST_EMAIL. The data is displayed in a grid with 35 rows, numbered 1 to 35. The last four rows (31 to 34) correspond to the data inserted in the previous SQL blocks.

CUST_ID	CUST_LAST	CUST_FIRST	CUST_PHONE	CUST_EMAIL
1000	Petre	Veronica	0732100000	(null)
1001	Emil	Alexandru	0732100100	emil.alexandru@gmail.com
1002	Anton	Iacob	(null)	(null)
1003	Antoni	Isabela	0732100300	antoni.isabela@gmail.com
1004	Toma	Alin	(null)	toma.alin@gmail.com
1005	Eduard	Carmen	0732100500	(null)
1006	Augustin	Iuliana	0732100600	augustin.iuliana@gmail.com
1007	Simion	Constantin	(null)	(null)
1008	Emanuel	Dragos	0732100800	emanuel.dragos@gmail.com
1009	Gheorghe	Liviu	0732100900	(null)
1010	Leonard	Emil	0732101000	leonard.emil@gmail.com
1011	Florin	Cezar	0732101100	(null)
1012	Dinu	Valentin	(null)	(null)
1013	Iosif	Stefan	0732101300	iosif.stefan@gmail.com
1014	Iancu	Serghei	(null)	iancu.serghei@gmail.com
1015	Ioan	Paul	0732101500	(null)
1016	Adrian	Petru	0732101600	adrian.petru@gmail.com
1017	Marius	Albert	(null)	(null)
1018	Gavril	Vlad	(null)	gavril.vlad@gmail.com
1019	Amalia	Camelia	(null)	(null)
1020	Enache	Andreea	0732102000	(null)
1021	Renata	Maria	0732102100	renata.maria@gmail.com
1022	Magda	Mihaela	0732102200	(null)
1023	Solomon	Constantina	0732102300	(null)
1024	Lunqu	Gabriela	(null)	(null)
1025	Stan	Larisa	(null)	stan.larisa@gmail.com
1026	Antonescu	Monica	0732102600	antonescu.monica@gmail.com
1027	David	Amalia	0732102700	(null)
1028	Vladimirescu	Antonia	(null)	(null)
1029	Unsur	Bianca	0732102900	unsur.bianca@gmail.com
1030	Petran	Elena	0732103000	(null)
1031	Albert	Liliana	(null)	albert.liliana@gmail.com
1032	Funar	Otilia	(null)	funar.otilia@gmail.com
1033	Lupei	Bogdana	0732103300	(null)
1034	Popa	Cosmina	(null)	(null)

```

insert into ORDER_MNG values (2000,'07-MAI-2016',null,100,1003);

insert into ORDER_MNG values (2001,'13-AUG-2017',null,100,1020);

insert into ORDER_MNG values (2002,'23-SEP-2018',null,101,1010);

insert into ORDER_MNG values (2003,'14-FEB-2019',null,101,1004);

insert into ORDER_MNG values (2004,'29-OCT-2020',null,102,1017);

insert into ORDER_MNG values (2005,'18-IAN-2021',null,102,1011);

insert into ORDER_MNG values (2006,'25-NOI-2020',null,103,1016);

insert into ORDER_MNG values (2007,'13-APR-2021',null,103,1001);

```



```

insert into ORDER_MNG values (2008,'16-DEC-2021',null,104,1002);
insert into ORDER_MNG values (2009,'08-MAR-2022',null,104,1014);
insert into ORDER_MNG values (2010,'24-MAR-2020',null,105,1023);
insert into ORDER_MNG values (2011,'07-MAI-2020',null,105,1017);
insert into ORDER_MNG values (2012,'21-AUG-2018',null,200,1013);
insert into ORDER_MNG values (2013,'19-DEC-2017',null,200,1005);
insert into ORDER_MNG values (2014,'23-FEB-2019',null,201,1025);
insert into ORDER_MNG values (2015,'28-MAI-2020',null,201,1003);
insert into ORDER_MNG values (2016,'12-SEP-2021',null,202,1020);
insert into ORDER_MNG values (2017,'27-MAR-2022',null,202,1030);
insert into ORDER_MNG values (2018,'01-NOI-2021',null,203,1034);
insert into ORDER_MNG values (2019,'01-MAR-2022',null,203,1009);
insert into ORDER_MNG values (2020,'30-OCT-2019',null,204,1021);
insert into ORDER_MNG values (2021,'19-IAN-2020',null,204,1016);
insert into ORDER_MNG values (2022,'30-MAR-2014',null,205,1003);
insert into ORDER_MNG values (2023,'10-MAI-2016',null,205,1010);
insert into ORDER_MNG values (2024,'10-MAI-2022',null,300,1022);
insert into ORDER_MNG values (2025,'17-APR-2018',null,301,1017);
insert into ORDER_MNG values (2026,'02-IUN-2019',null,301,1008);
insert into ORDER_MNG values (2027,'23-SEP-2016',null,302,1013);
insert into ORDER_MNG values (2028,'11-OCT-2017',null,302,1028);
insert into ORDER_MNG values (2029,'23-OCT-2016',null,303,1021);
insert into ORDER_MNG values (2030,'18-DEC-2016',null,303,1029);
insert into ORDER_MNG values (2031,'20-FEB-2014',null,304,1034);
insert into ORDER_MNG values (2032,'20-MAI-2015',null,304,1032);
insert into ORDER_MNG values (2033,'14-IUL-2021',null,305,1007);
insert into ORDER_MNG values (2034,'16-AUG-2021',null,305,1021);
insert into ORDER_MNG values (2035,'03-IAN-2022',null,400,1010);
insert into ORDER_MNG values (2036,'28-APR-2022',null,400,1015);
insert into ORDER_MNG values (2037,'05-AUG-2019',null,401,1019);

```

```

insert into ORDER_MNG values (2038,'24-NOI-2019',null,401,1023);
insert into ORDER_MNG values (2039,'15-APR-2022',null,402,1033);
insert into ORDER_MNG values (2040,'01-MAI-2022',null,402,1022);
insert into ORDER_MNG values (2041,'27-OCT-2019',null,403,1016);
insert into ORDER_MNG values (2042,'20-DEC-2019',null,403,1014);
insert into ORDER_MNG values (2043,'10-IAN-2017',null,404,1001);
insert into ORDER_MNG values (2044,'26-SEP-2018',null,404,1018);
insert into ORDER_MNG values (2045,'13-NOI-2021',null,405,1007);
insert into ORDER_MNG values (2046,'26-FEB-2022',null,405,1021);
insert into ORDER_MNG values (2047,'14-IUL-2019',null,500,1027);
insert into ORDER_MNG values (2048,'13-AUG-2019',null,500,1031);
insert into ORDER_MNG values (2049,'03-DEC-2020',null,501,1011);
insert into ORDER_MNG values (2050,'19-FEB-2021',null,501,1002);
insert into ORDER_MNG values (2051,'25-SEP-2017',null,502,1012);
insert into ORDER_MNG values (2052,'16-APR-2018',null,502,1024);
insert into ORDER_MNG values (2053,'17-FEB-2020',null,503,1000);
insert into ORDER_MNG values (2054,'29-IUL-2021',null,503,1016);
insert into ORDER_MNG values (2055,'22-SEP-2017',null,504,1023);
insert into ORDER_MNG values (2056,'10-DEC-2018',null,504,1009);
insert into ORDER_MNG values (2057,'08-NOI-2019',null,505,1014);
insert into ORDER_MNG values (2058,'15-AUG-2020',null,505,1033);

```

	ORD_ID	ORD_DATE	ORD_PRICE	EMP_ID	CUST_ID
1	2000	07-05-2016	360	100	1003
2	2001	13-08-2017	95	100	1020
3	2002	23-09-2018	90	101	1010
4	2003	14-02-2019	250	101	1004
5	2004	29-10-2020	235	102	1017
6	2005	18-01-2021	60	102	1011
7	2006	25-11-2020	90	103	1016
8	2007	13-04-2021	100	103	1001
9	2008	16-12-2021	60	104	1002
10	2009	08-03-2022	100	104	1014
11	2010	24-03-2020	60	105	1023
12	2011	07-05-2020	145	105	1017
13	2012	21-08-2018	150	200	1013
14	2013	19-12-2017	90	200	1005
15	2014	23-02-2019	120	201	1025
16	2015	28-05-2020	60	201	1003
17	2016	12-09-2021	240	202	1020
18	2017	27-03-2022	140	202	1030
19	2018	01-11-2021	185	203	1034
20	2019	01-03-2022	120	203	1009
21	2020	30-10-2019	160	204	1021
22	2021	19-01-2020	60	204	1016
23	2022	30-03-2014	40	205	1003
24	2023	10-05-2016	245	205	1010
25	2024	10-05-2022	120	300	1022
26	2025	17-04-2018	95	301	1017
27	2026	02-06-2019	90	301	1008
28	2027	23-09-2016	120	302	1013
29	2028	11-10-2017	60	302	1028

```

insert into ORDER_CONSISTS_OF values (2000,901,1);
insert into ORDER_CONSISTS_OF values (2000,915,2);
insert into ORDER_CONSISTS_OF values (2000,904,3);
insert into ORDER_CONSISTS_OF values (2001,927,1);
insert into ORDER_CONSISTS_OF values (2001,917,1);
insert into ORDER_CONSISTS_OF values (2002,913,2);
insert into ORDER_CONSISTS_OF values (2003,917,4);
insert into ORDER_CONSISTS_OF values (2003,909,2);
insert into ORDER_CONSISTS_OF values (2004,903,1);
insert into ORDER_CONSISTS_OF values (2004,914,1);
insert into ORDER_CONSISTS_OF values (2004,911,1);
insert into ORDER_CONSISTS_OF values (2004,902,2);
insert into ORDER_CONSISTS_OF values (2005,910,2);
insert into ORDER_CONSISTS_OF values (2006,911,3);
insert into ORDER_CONSISTS_OF values (2007,912,1);
insert into ORDER_CONSISTS_OF values (2007,900,1);
insert into ORDER_CONSISTS_OF values (2008,907,1);
insert into ORDER_CONSISTS_OF values (2009,902,2);
insert into ORDER_CONSISTS_OF values (2010,915,1);
insert into ORDER_CONSISTS_OF values (2011,923,1);
insert into ORDER_CONSISTS_OF values (2011,924,1);
insert into ORDER_CONSISTS_OF values (2011,925,1);
insert into ORDER_CONSISTS_OF values (2012,920,2);
insert into ORDER_CONSISTS_OF values (2012,902,1);
insert into ORDER_CONSISTS_OF values (2013,914,2);
insert into ORDER_CONSISTS_OF values (2014,904,2);
insert into ORDER_CONSISTS_OF values (2015,908,1);
insert into ORDER_CONSISTS_OF values (2016,913,3);
insert into ORDER_CONSISTS_OF values (2016,917,1);
insert into ORDER_CONSISTS_OF values (2016,918,1);

```

```

insert into ORDER_CONSISTS_OF values (2017,921,2);
insert into ORDER_CONSISTS_OF values (2018,917,3);
insert into ORDER_CONSISTS_OF values (2018,920,1);
insert into ORDER_CONSISTS_OF values (2019,901,1);
insert into ORDER_CONSISTS_OF values (2019,910,2);
insert into ORDER_CONSISTS_OF values (2020,912,4);
insert into ORDER_CONSISTS_OF values (2021,903,1);
insert into ORDER_CONSISTS_OF values (2022,906,1);
insert into ORDER_CONSISTS_OF values (2023,908,2);
insert into ORDER_CONSISTS_OF values (2023,909,1);
insert into ORDER_CONSISTS_OF values (2023,910,3);
insert into ORDER_CONSISTS_OF values (2024,915,2);
insert into ORDER_CONSISTS_OF values (2025,917,1);
insert into ORDER_CONSISTS_OF values (2025,905,1);
insert into ORDER_CONSISTS_OF values (2026,913,2);
insert into ORDER_CONSISTS_OF values (2027,900,2);
insert into ORDER_CONSISTS_OF values (2028,903,1);
insert into ORDER_CONSISTS_OF values (2029,908,2);
insert into ORDER_CONSISTS_OF values (2030,911,1);
insert into ORDER_CONSISTS_OF values (2030,908,1);
insert into ORDER_CONSISTS_OF values (2030,903,2);
insert into ORDER_CONSISTS_OF values (2030,901,1);
insert into ORDER_CONSISTS_OF values (2030,909,1);
insert into ORDER_CONSISTS_OF values (2031,920,3);
insert into ORDER_CONSISTS_OF values (2032,925,3);
insert into ORDER_CONSISTS_OF values (2033,914,2);
insert into ORDER_CONSISTS_OF values (2033,911,1);
insert into ORDER_CONSISTS_OF values (2034,901,2);
insert into ORDER_CONSISTS_OF values (2035,919,1);
insert into ORDER_CONSISTS_OF values (2036,929,1);

```

```
insert into ORDER_CONSISTS_OF values (2037,928,2);
insert into ORDER_CONSISTS_OF values (2038,916,1);
insert into ORDER_CONSISTS_OF values (2039,910,1);
insert into ORDER_CONSISTS_OF values (2040,917,2);
insert into ORDER_CONSISTS_OF values (2041,920,2);
insert into ORDER_CONSISTS_OF values (2042,913,1);
insert into ORDER_CONSISTS_OF values (2042,900,1);
insert into ORDER_CONSISTS_OF values (2042,901,2);
insert into ORDER_CONSISTS_OF values (2043,906,1);
insert into ORDER_CONSISTS_OF values (2044,909,2);
insert into ORDER_CONSISTS_OF values (2045,910,3);
insert into ORDER_CONSISTS_OF values (2046,902,2);
insert into ORDER_CONSISTS_OF values (2046,912,1);
insert into ORDER_CONSISTS_OF values (2046,922,1);
insert into ORDER_CONSISTS_OF values (2047,915,2);
insert into ORDER_CONSISTS_OF values (2048,913,1);
insert into ORDER_CONSISTS_OF values (2049,909,1);
insert into ORDER_CONSISTS_OF values (2050,912,1);
insert into ORDER_CONSISTS_OF values (2050,910,1);
insert into ORDER_CONSISTS_OF values (2050,907,2);
insert into ORDER_CONSISTS_OF values (2051,918,3);
insert into ORDER_CONSISTS_OF values (2052,914,1);
insert into ORDER_CONSISTS_OF values (2053,902,3);
insert into ORDER_CONSISTS_OF values (2054,917,2);
insert into ORDER_CONSISTS_OF values (2055,916,1);
insert into ORDER_CONSISTS_OF values (2056,913,1);
insert into ORDER_CONSISTS_OF values (2057,914,2);
insert into ORDER_CONSISTS_OF values (2058,928,1);
```

	ORD_ID	FILM_ID	NUMBER_OF FILMS
1	2000	901	1
2	2000	915	2
3	2000	904	3
4	2001	927	1
5	2001	917	1
6	2002	913	2
7	2003	917	4
8	2003	909	2
9	2004	903	1
10	2004	914	1
11	2004	911	1
12	2004	902	2
13	2005	910	2
14	2006	911	3
15	2007	912	1
16	2007	900	1
17	2008	907	1
18	2009	902	2
19	2010	915	1
20	2011	923	1
21	2011	924	1
22	2011	925	1
23	2012	920	2

```

insert into RENTAL_MNG values (3000,'13-OCT-2017','20-10-2017',null,100,1000);
insert into RENTAL_MNG values (3001,'20-MAR-2019','27-03-2019',null,100,1001);
insert into RENTAL_MNG values (3002,'05-IAN-2020','12-01-2020',null,100,1003);
insert into RENTAL_MNG values (3003,'30-OCT-2019','06-11-2019',null,100,1004);
insert into RENTAL_MNG values (3004,'21-APR-2014','28-04-2014',null,101,1002);
insert into RENTAL_MNG values (3005,'30-SEP-2014','07-10-2014',null,101,1005);
insert into RENTAL_MNG values (3006,'10-IUN-2015','17-06-2015',null,101,1006);
insert into RENTAL_MNG values (3007,'13-FEB-2016','20-02-2016',null,101,1007);
insert into RENTAL_MNG values (3008,'19-OCT-2020','26-10-2020',null,102,1008);
insert into RENTAL_MNG values (3009,'20-DEC-2020','27-12-2020',null,102,1009);
insert into RENTAL_MNG values (3010,'02-MAI-2021','09-05-2021',null,102,1012);
insert into RENTAL_MNG values (3011,'28-FEB-2022','07-03-2022',null,102,1020);
insert into RENTAL_MNG values (3012,'09-SEP-2021','16-09-2021',null,102,1016);
insert into RENTAL_MNG values (3013,'17-IUL-2020','24-07-2020',null,103,1009);
insert into RENTAL_MNG values (3014,'03-OCT-2020','10-10-2020',null,103,1002);
insert into RENTAL_MNG values (3015,'05-IAN-2021','12-01-2021',null,103,1017);
insert into RENTAL_MNG values (3016,'21-MAI-2021','28-05-2021',null,103,1014);
insert into RENTAL_MNG values (3017,'23-APR-2022','30-04-2022',null,103,1008);
insert into RENTAL_MNG values (3018,'23-AUG-2021','30-08-2021',null,104,1013);
insert into RENTAL_MNG values (3019,'24-OCT-2021','31-10-2021',null,104,1024);
insert into RENTAL_MNG values (3020,'26-FEB-2022','05-03-2022',null,104,1030);

```

```

insert into RENTAL_MNG values (3021,'17-MAR-2022','24-03-2022',null,104,1010);
insert into RENTAL_MNG values (3022,'02-APR-2022','09-04-2022',null,104,1028);
insert into RENTAL_MNG values (3023,'19-IUL-2019','26-07-2019',null,105,1017);
insert into RENTAL_MNG values (3024,'30-OCT-2019','06-11-2019',null,105,1003);
insert into RENTAL_MNG values (3025,'02-MAR-2020','09-03-2020',null,105,1006);
insert into RENTAL_MNG values (3026,'15-MAR-2020','22-03-2020',null,105,1012);
insert into RENTAL_MNG values (3027,'20-MAI-2021','27-05-2021',null,105,1032);
insert into RENTAL_MNG values (3028,'12-DEC-2021','19-12-2021',null,105,1001);
insert into RENTAL_MNG values (3029,'21-OCT-2017','28-10-2017',null,200,1034);
insert into RENTAL_MNG values (3030,'20-IAN-2018','27-01-2018',null,200,1003);
insert into RENTAL_MNG values (3031,'08-IUL-2018','15-07-2018',null,200,1023);
insert into RENTAL_MNG values (3032,'13-DEC-2018','20-12-2018',null,200,1031);
insert into RENTAL_MNG values (3033,'03-MAR-2019','10-03-2019',null,200,1011);
insert into RENTAL_MNG values (3034,'28-OCT-2018','04-11-2018',null,201,1008);
insert into RENTAL_MNG values (3035,'19-IAN-2019','26-01-2019',null,201,1013);
insert into RENTAL_MNG values (3036,'11-MAI-2019','18-05-2019',null,201,1005);
insert into RENTAL_MNG values (3037,'20-OCT-2020','27-10-2020',null,201,1017);
insert into RENTAL_MNG values (3038,'13-IUN-2021','20-06-2021',null,201,1033);
insert into RENTAL_MNG values (3039,'10-OCT-2021','17-10-2021',null,201,1002);
insert into RENTAL_MNG values (3040,'02-OCT-2020','09-10-2020',null,202,1006);
insert into RENTAL_MNG values (3041,'12-APR-2021','19-04-2021',null,202,1013);
insert into RENTAL_MNG values (3042,'21-OCT-2021','28-10-2021',null,202,1004);
insert into RENTAL_MNG values (3043,'01-FEB-2022','08-02-2022',null,202,1019);
insert into RENTAL_MNG values (3044,'29-APR-2022','06-05-2022',null,202,1007);
insert into RENTAL_MNG values (3045,'23-DEC-2021','30-12-2021',null,203,1016);
insert into RENTAL_MNG values (3046,'03-MAR-2022','10-03-2022',null,203,1009);
insert into RENTAL_MNG values (3047,'20-APR-2022','27-04-2022',null,203,1012);
insert into RENTAL_MNG values (3048,'13-IAN-2022','20-01-2022',null,203,1004);
insert into RENTAL_MNG values (3049,'18-FEB-2022','25-02-2022',null,203,1019);
insert into RENTAL_MNG values (3050,'01-NOI-2019','08-11-2019',null,204,1034);

```



```

insert into RENTAL_MNG values (3051,'10-IAN-2020','17-01-2020',null,204,1030);
insert into RENTAL_MNG values (3052,'05-IUN-2020','12-06-2020',null,204,1015);
insert into RENTAL_MNG values (3053,'19-OCT-2020','26-10-2020',null,204,1002);
insert into RENTAL_MNG values (3054,'21-MAR-2021','28-03-2021',null,204,1010);
insert into RENTAL_MNG values (3055,'20-AUG-2014','27-08-2014',null,205,1017);
insert into RENTAL_MNG values (3056,'03-DEC-2014','10-12-2014',null,205,1021);
insert into RENTAL_MNG values (3057,'15-OCT-2015','22-10-2015',null,205,1004);
insert into RENTAL_MNG values (3058,'16-MAR-2016','23-03-2016',null,205,1002);
insert into RENTAL_MNG values (3059,'10-OCT-2016','17-10-2016',null,205,1011);
insert into RENTAL_MNG values (3060,'10-MAI-2022','17-05-2022',null,300,1000);
insert into RENTAL_MNG values (3061,'19-MAI-2022','26-05-2022',null,300,1034);
insert into RENTAL_MNG values (3062,'20-APR-2019','27-04-2019',null,301,1009);
insert into RENTAL_MNG values (3063,'10-IUN-2019','17-06-2019',null,301,1020);
insert into RENTAL_MNG values (3064,'13-MAR-2020','20-03-2020',null,301,1017);
insert into RENTAL_MNG values (3065,'28-SEP-2021','05-10-2021',null,301,1005);
insert into RENTAL_MNG values (3066,'14-MAR-2016','21-03-2016',null,302,1012);
insert into RENTAL_MNG values (3067,'23-SEP-2017','30-09-2017',null,302,1007);
insert into RENTAL_MNG values (3068,'15-FEB-2018','22-02-2018',null,302,1028);
insert into RENTAL_MNG values (3069,'13-OCT-2019','20-10-2019',null,302,1033);
insert into RENTAL_MNG values (3070,'15-OCT-2016','22-10-2016',null,303,1004);
insert into RENTAL_MNG values (3071,'06-DEC-2016','13-12-2016',null,303,1010);
insert into RENTAL_MNG values (3072,'07-IAN-2017','14-01-2017',null,303,1029);
insert into RENTAL_MNG values (3073,'01-FEB-2014','08-02-2014',null,304,1006);
insert into RENTAL_MNG values (3074,'29-OCT-2015','05-11-2015',null,304,1019);
insert into RENTAL_MNG values (3075,'27-IAN-2016','03-02-2016',null,304,1024);
insert into RENTAL_MNG values (3076,'10-DEC-2020','17-12-2020',null,305,1016);
insert into RENTAL_MNG values (3077,'03-DEC-2021','10-12-2021',null,305,1021);
insert into RENTAL_MNG values (3078,'16-IAN-2022','23-01-2022',null,305,1034);
insert into RENTAL_MNG values (3079,'23-APR-2022','30-04-2022',null,305,1005);
insert into RENTAL_MNG values (3080,'27-IAN-2022','03-02-2022',null,400,1007);

```



```

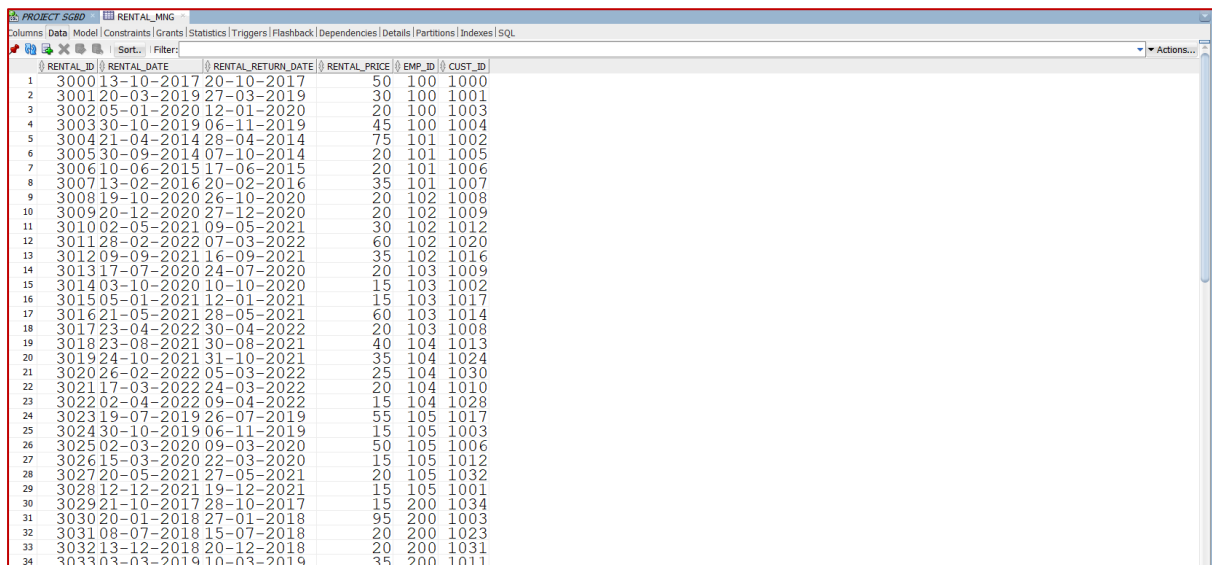
insert into RENTAL_MNG values (3081,'19-FEB-2022','26-02-2022',null,400,1018);
insert into RENTAL_MNG values (3082,'05-MAR-2022','12-03-2022',null,400,1023);
insert into RENTAL_MNG values (3083,'20-APR-2018','27-04-2018',null,401,1002);
insert into RENTAL_MNG values (3084,'27-OCT-2018','03-11-2018',null,401,1016);
insert into RENTAL_MNG values (3085,'23-MAI-2019','30-05-2019',null,401,1028);
insert into RENTAL_MNG values (3086,'02-NOI-2019','09-11-2019',null,401,1033);
insert into RENTAL_MNG values (3087,'16-MAR-2020','23-03-2020',null,401,1004);
insert into RENTAL_MNG values (3088,'20-MAI-2022','27-05-2022',null,402,1017);
insert into RENTAL_MNG values (3089,'26-APR-2017','03-05-2017',null,402,1034);
insert into RENTAL_MNG values (3090,'16-MAI-2019','23-05-2019',null,403,1002);
insert into RENTAL_MNG values (3091,'20-DEC-2019','27-12-2019',null,403,1028);
insert into RENTAL_MNG values (3092,'03-APR-2020','10-04-2020',null,403,1003);
insert into RENTAL_MNG values (3093,'07-IUN-2020','14-06-2020',null,403,1011);
insert into RENTAL_MNG values (3094,'23-OCT-2017','30-10-2017',null,404,1027);
insert into RENTAL_MNG values (3095,'24-FEB-2018','03-03-2018',null,404,1011);
insert into RENTAL_MNG values (3096,'10-DEC-2018','17-12-2018',null,404,1026);
insert into RENTAL_MNG values (3097,'22-FEB-2019','01-03-2019',null,404,1015);
insert into RENTAL_MNG values (3098,'29-IUN-2019','06-07-2019',null,404,1006);
insert into RENTAL_MNG values (3099,'04-NOI-2021','11-11-2021',null,405,1013);
insert into RENTAL_MNG values (3100,'12-MAR-2022','19-03-2022',null,405,1004);
insert into RENTAL_MNG values (3101,'06-APR-2022','13-04-2022',null,405,1009);
insert into RENTAL_MNG values (3102,'24-SEP-2018','01-10-2018',null,500,1004);
insert into RENTAL_MNG values (3103,'18-FEB-2019','25-02-2019',null,500,1012);
insert into RENTAL_MNG values (3104,'27-AUG-2019','03-09-2019',null,500,1031);
insert into RENTAL_MNG values (3105,'29-IAN-2020','05-02-2020',null,500,1027);
insert into RENTAL_MNG values (3106,'22-SEP-2020','29-09-2020',null,501,1007);
insert into RENTAL_MNG values (3107,'06-OCT-2021','13-10-2021',null,501,1020);
insert into RENTAL_MNG values (3108,'24-MAI-2016','31-05-2016',null,502,1013);
insert into RENTAL_MNG values (3109,'01-SEP-2017','08-09-2017',null,502,1001);
insert into RENTAL_MNG values (3110,'16-MAR-2018','23-03-2018',null,502,1017);

```

```

insert into RENTAL_MNG values (3111,'08-NOI-2021','15-11-2021',null,503,1030);
insert into RENTAL_MNG values (3112,'02-IUL-2020','09-07-2020',null,503,1004);
insert into RENTAL_MNG values (3113,'21-APR-2015','28-04-2015',null,504,1016);
insert into RENTAL_MNG values (3114,'02-DEC-2016','09-12-2016',null,504,1000);
insert into RENTAL_MNG values (3115,'21-AUG-2017','28-08-2017',null,504,1019);
insert into RENTAL_MNG values (3116,'13-IUN-2019','20-06-2019',null,505,1002);
insert into RENTAL_MNG values (3117,'20-NOI-2020','27-11-2020',null,505,1015);

```



The screenshot shows a database window titled 'PROJECT_SGBD' with the 'RENTAL_MNG' table selected. The table contains 34 rows of data. The columns are: RENTAL_ID, RENTAL_DATE, RENTAL_RETURN_DATE, RENTAL_PRICE, EMP_ID, and CUST_ID. The data is as follows:

RENTAL_ID	RENTAL_DATE	RENTAL_RETURN_DATE	RENTAL_PRICE	EMP_ID	CUST_ID
3000	13-10-2017	20-10-2017	50	100	1000
3001	20-03-2019	27-03-2019	30	100	1001
3002	05-01-2020	12-01-2020	20	100	1003
3003	30-10-2019	06-11-2019	45	100	1004
3004	21-04-2014	28-04-2014	75	101	1002
3005	30-09-2014	07-10-2014	20	101	1005
3006	10-06-2015	17-06-2015	20	101	1006
3007	13-02-2016	20-02-2016	35	101	1007
3008	19-10-2020	26-10-2020	20	102	1008
3009	20-12-2020	27-12-2020	20	102	1009
3010	02-05-2021	09-05-2021	30	102	1012
3011	28-02-2022	07-03-2022	60	102	1020
3012	09-09-2021	16-09-2021	35	102	1016
3013	17-07-2020	24-07-2020	20	103	1009
3014	03-10-2020	10-10-2020	15	103	1002
3015	05-01-2021	12-01-2021	15	103	1017
3016	21-05-2021	28-05-2021	60	103	1014
3017	23-04-2022	30-04-2022	20	103	1008
3018	23-08-2021	30-08-2021	40	104	1013
3019	24-10-2021	31-10-2021	35	104	1024
3020	26-02-2022	05-03-2022	25	104	1030
3021	17-03-2022	24-03-2022	20	104	1010
3022	02-04-2022	09-04-2022	15	104	1028
3023	19-07-2019	26-07-2019	55	105	1017
3024	30-10-2019	06-11-2019	15	105	1003
3025	02-03-2020	09-03-2020	50	105	1006
3026	15-03-2020	22-03-2020	15	105	1012
3027	20-05-2021	27-05-2021	20	105	1032
3028	12-12-2021	19-12-2021	15	105	1001
3029	21-10-2017	28-10-2017	15	200	1034
3030	20-01-2018	27-01-2018	95	200	1003
3031	08-07-2018	15-07-2018	20	200	1023
3032	13-12-2018	20-12-2018	20	200	1031
3033	03-03-2019	10-03-2019	35	200	1011

```

insert into RENTAL_CONSISTS_OF values (3000,912);
insert into RENTAL_CONSISTS_OF values (3000,907);
insert into RENTAL_CONSISTS_OF values (3000,902);
insert into RENTAL_CONSISTS_OF values (3001,927);
insert into RENTAL_CONSISTS_OF values (3001,917);
insert into RENTAL_CONSISTS_OF values (3002,900);
insert into RENTAL_CONSISTS_OF values (3003,901);
insert into RENTAL_CONSISTS_OF values (3003,905);
insert into RENTAL_CONSISTS_OF values (3004,910);
insert into RENTAL_CONSISTS_OF values (3004,911);
insert into RENTAL_CONSISTS_OF values (3004,912);

```

insert into RENTAL_CONSISTS_OF values (3004,904);
insert into RENTAL_CONSISTS_OF values (3005,907);
insert into RENTAL_CONSISTS_OF values (3006,926);
insert into RENTAL_CONSISTS_OF values (3007,923);
insert into RENTAL_CONSISTS_OF values (3007,919);
insert into RENTAL_CONSISTS_OF values (3008,920);
insert into RENTAL_CONSISTS_OF values (3009,901);
insert into RENTAL_CONSISTS_OF values (3010,915);
insert into RENTAL_CONSISTS_OF values (3011,925);
insert into RENTAL_CONSISTS_OF values (3011,924);
insert into RENTAL_CONSISTS_OF values (3011,921);
insert into RENTAL_CONSISTS_OF values (3012,920);
insert into RENTAL_CONSISTS_OF values (3012,902);
insert into RENTAL_CONSISTS_OF values (3013,907);
insert into RENTAL_CONSISTS_OF values (3014,911);
insert into RENTAL_CONSISTS_OF values (3015,910);
insert into RENTAL_CONSISTS_OF values (3016,903);
insert into RENTAL_CONSISTS_OF values (3016,921);
insert into RENTAL_CONSISTS_OF values (3016,906);
insert into RENTAL_CONSISTS_OF values (3017,901);
insert into RENTAL_CONSISTS_OF values (3018,903);
insert into RENTAL_CONSISTS_OF values (3018,908);
insert into RENTAL_CONSISTS_OF values (3019,910);
insert into RENTAL_CONSISTS_OF values (3019,921);
insert into RENTAL_CONSISTS_OF values (3020,905);
insert into RENTAL_CONSISTS_OF values (3021,923);
insert into RENTAL_CONSISTS_OF values (3022,914);
insert into RENTAL_CONSISTS_OF values (3023,902);
insert into RENTAL_CONSISTS_OF values (3023,919);
insert into RENTAL_CONSISTS_OF values (3023,905);

insert into RENTAL_CONSISTS_OF values (3024,917);
insert into RENTAL_CONSISTS_OF values (3025,920);
insert into RENTAL_CONSISTS_OF values (3025,915);
insert into RENTAL_CONSISTS_OF values (3026,902);
insert into RENTAL_CONSISTS_OF values (3027,901);
insert into RENTAL_CONSISTS_OF values (3028,913);
insert into RENTAL_CONSISTS_OF values (3029,924);
insert into RENTAL_CONSISTS_OF values (3030,901);
insert into RENTAL_CONSISTS_OF values (3030,918);
insert into RENTAL_CONSISTS_OF values (3030,905);
insert into RENTAL_CONSISTS_OF values (3030,911);
insert into RENTAL_CONSISTS_OF values (3030,919);
insert into RENTAL_CONSISTS_OF values (3031,908);
insert into RENTAL_CONSISTS_OF values (3032,926);
insert into RENTAL_CONSISTS_OF values (3033,903);
insert into RENTAL_CONSISTS_OF values (3033,922);
insert into RENTAL_CONSISTS_OF values (3034,910);
insert into RENTAL_CONSISTS_OF values (3035,926);
insert into RENTAL_CONSISTS_OF values (3036,929);
insert into RENTAL_CONSISTS_OF values (3037,916);
insert into RENTAL_CONSISTS_OF values (3038,903);
insert into RENTAL_CONSISTS_OF values (3039,910);
insert into RENTAL_CONSISTS_OF values (3040,902);
insert into RENTAL_CONSISTS_OF values (3041,910);
insert into RENTAL_CONSISTS_OF values (3042,923);
insert into RENTAL_CONSISTS_OF values (3042,907);
insert into RENTAL_CONSISTS_OF values (3042,911);
insert into RENTAL_CONSISTS_OF values (3043,919);
insert into RENTAL_CONSISTS_OF values (3044,902);
insert into RENTAL_CONSISTS_OF values (3045,905);

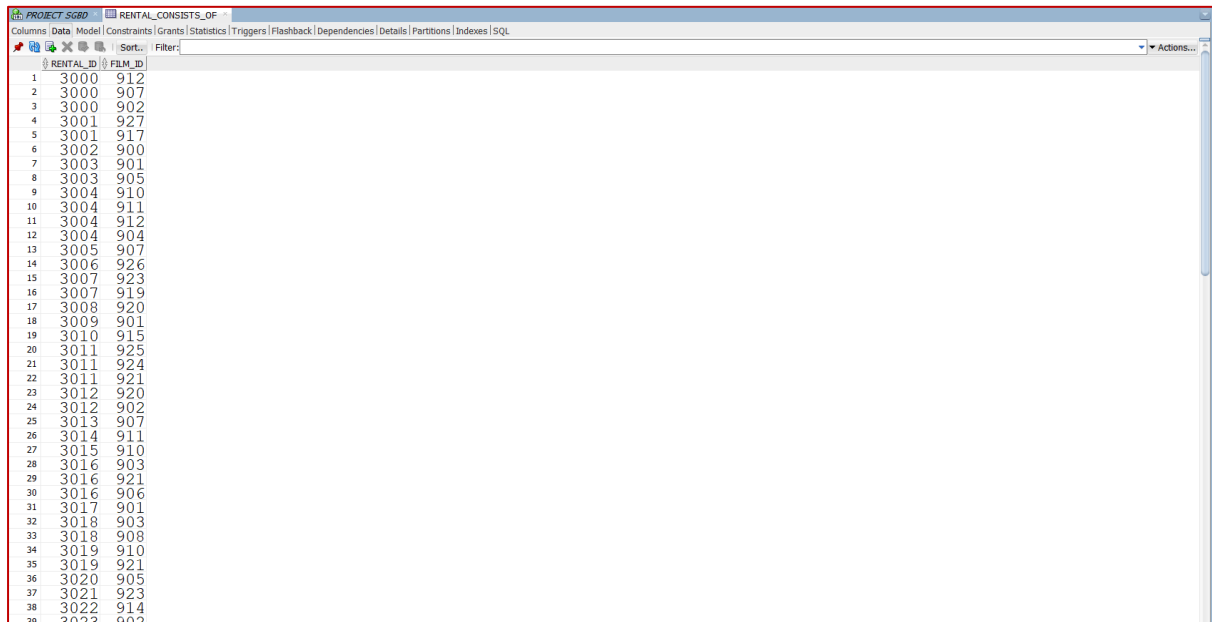
insert into RENTAL_CONSISTS_OF values (3046,921);
insert into RENTAL_CONSISTS_OF values (3046,908);
insert into RENTAL_CONSISTS_OF values (3046,916);
insert into RENTAL_CONSISTS_OF values (3047,903);
insert into RENTAL_CONSISTS_OF values (3048,901);
insert into RENTAL_CONSISTS_OF values (3049,910);
insert into RENTAL_CONSISTS_OF values (3050,928);
insert into RENTAL_CONSISTS_OF values (3050,901);
insert into RENTAL_CONSISTS_OF values (3050,919);
insert into RENTAL_CONSISTS_OF values (3051,921);
insert into RENTAL_CONSISTS_OF values (3052,905);
insert into RENTAL_CONSISTS_OF values (3053,921);
insert into RENTAL_CONSISTS_OF values (3054,908);
insert into RENTAL_CONSISTS_OF values (3055,914);
insert into RENTAL_CONSISTS_OF values (3056,904);
insert into RENTAL_CONSISTS_OF values (3057,927);
insert into RENTAL_CONSISTS_OF values (3058,902);
insert into RENTAL_CONSISTS_OF values (3059,900);
insert into RENTAL_CONSISTS_OF values (3060,929);
insert into RENTAL_CONSISTS_OF values (3061,917);
insert into RENTAL_CONSISTS_OF values (3062,916);
insert into RENTAL_CONSISTS_OF values (3063,918);
insert into RENTAL_CONSISTS_OF values (3064,910);
insert into RENTAL_CONSISTS_OF values (3065,911);
insert into RENTAL_CONSISTS_OF values (3066,924);
insert into RENTAL_CONSISTS_OF values (3067,927);
insert into RENTAL_CONSISTS_OF values (3068,913);
insert into RENTAL_CONSISTS_OF values (3069,909);
insert into RENTAL_CONSISTS_OF values (3070,916);
insert into RENTAL_CONSISTS_OF values (3071,910);

insert into RENTAL_CONSISTS_OF values (3072,912);
insert into RENTAL_CONSISTS_OF values (3073,926);
insert into RENTAL_CONSISTS_OF values (3074,920);
insert into RENTAL_CONSISTS_OF values (3075,901);
insert into RENTAL_CONSISTS_OF values (3076,907);
insert into RENTAL_CONSISTS_OF values (3077,921);
insert into RENTAL_CONSISTS_OF values (3078,929);
insert into RENTAL_CONSISTS_OF values (3079,910);
insert into RENTAL_CONSISTS_OF values (3080,924);
insert into RENTAL_CONSISTS_OF values (3081,914);
insert into RENTAL_CONSISTS_OF values (3082,905);
insert into RENTAL_CONSISTS_OF values (3083,924);
insert into RENTAL_CONSISTS_OF values (3084,923);
insert into RENTAL_CONSISTS_OF values (3085,911);
insert into RENTAL_CONSISTS_OF values (3086,900);
insert into RENTAL_CONSISTS_OF values (3087,918);
insert into RENTAL_CONSISTS_OF values (3088,906);
insert into RENTAL_CONSISTS_OF values (3089,923);
insert into RENTAL_CONSISTS_OF values (3090,911);
insert into RENTAL_CONSISTS_OF values (3091,902);
insert into RENTAL_CONSISTS_OF values (3092,928);
insert into RENTAL_CONSISTS_OF values (3093,929);
insert into RENTAL_CONSISTS_OF values (3094,910);
insert into RENTAL_CONSISTS_OF values (3095,927);
insert into RENTAL_CONSISTS_OF values (3096,912);
insert into RENTAL_CONSISTS_OF values (3097,911);
insert into RENTAL_CONSISTS_OF values (3098,922);
insert into RENTAL_CONSISTS_OF values (3099,928);
insert into RENTAL_CONSISTS_OF values (3100,901);
insert into RENTAL_CONSISTS_OF values (3101,911);

```

insert into RENTAL_CONSISTS_OF values (3102,926);
insert into RENTAL_CONSISTS_OF values (3103,906);
insert into RENTAL_CONSISTS_OF values (3104,908);
insert into RENTAL_CONSISTS_OF values (3105,919);
insert into RENTAL_CONSISTS_OF values (3106,926);
insert into RENTAL_CONSISTS_OF values (3107,920);
insert into RENTAL_CONSISTS_OF values (3108,910);
insert into RENTAL_CONSISTS_OF values (3109,925);
insert into RENTAL_CONSISTS_OF values (3110,912);
insert into RENTAL_CONSISTS_OF values (3111,923);
insert into RENTAL_CONSISTS_OF values (3112,904);
insert into RENTAL_CONSISTS_OF values (3113,907);
insert into RENTAL_CONSISTS_OF values (3114,909);
insert into RENTAL_CONSISTS_OF values (3115,921);
insert into RENTAL_CONSISTS_OF values (3116,917);
insert into RENTAL_CONSISTS_OF values (3117,920);

```



	RENTAL_ID	FILM_ID
1	3000	912
2	3000	907
3	3000	902
4	3001	927
5	3001	917
6	3002	900
7	3003	901
8	3003	905
9	3004	910
10	3004	911
11	3004	912
12	3004	904
13	3005	907
14	3006	926
15	3007	923
16	3007	919
17	3008	920
18	3009	901
19	3010	915
20	3011	925
21	3011	924
22	3011	921
23	3012	920
24	3012	902
25	3013	907
26	3014	911
27	3015	910
28	3016	903
29	3016	921
30	3016	906
31	3017	901
32	3018	903
33	3018	908
34	3019	910
35	3019	921
36	3020	905
37	3021	923
38	3022	914
39	3023	902

12. Implementarea a 5 cereri SQL complexe.

Aceste cereri vor utiliza în asamblul lor următoarele componente:

- ☞ subcereri sincronizate în care intervin cel puțin 3 tabele;
- ☞ subcereri nesincronizate în clauza FROM;
- ☞ grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri);
- ☞ ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri);
- ☞ utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE;
- ☞ utilizarea a cel puțin 1 bloc de cerere (clauza WITH);

~ CEREREA 1 ~

Cerință: Să se listeze pentru „The Velvet Stage” filmul care aduce cel mai mare profit: numărul de vânzări și închirieri cumulate.

WITH

```
O AS (SELECT film_title TITLU, SUM(o_c.number_of_films) PROFIT
      FROM film_mng f
      JOIN order_consists_of o_c ON (f.film_id = o_c.film_id)
      JOIN order_mng o ON (o_c.ord_id = o.ord_id)
      JOIN works_in w_i ON (o.emp_id = w_i.emp_id)
      JOIN store_mng s ON (w_i.store_id = s.store_id)
      WHERE UPPER(s.store_name) = 'THE VELVET STAGE'
      GROUP BY film_title),
```

```
R AS (SELECT film_title TITLU, COUNT(r_c.film_id) PROFIT
      FROM film_mng f
      JOIN rental_consists_of r_c ON (f.film_id = r_c.film_id)
      JOIN rental_mng r ON (r_c.rental_id = r.rental_id)
      JOIN works_in w_i ON (r.emp_id = w_i.emp_id)
```



```

WHERE UPPER(s.store_name) = 'THE VELVET STAGE'

GROUP BY film_title)

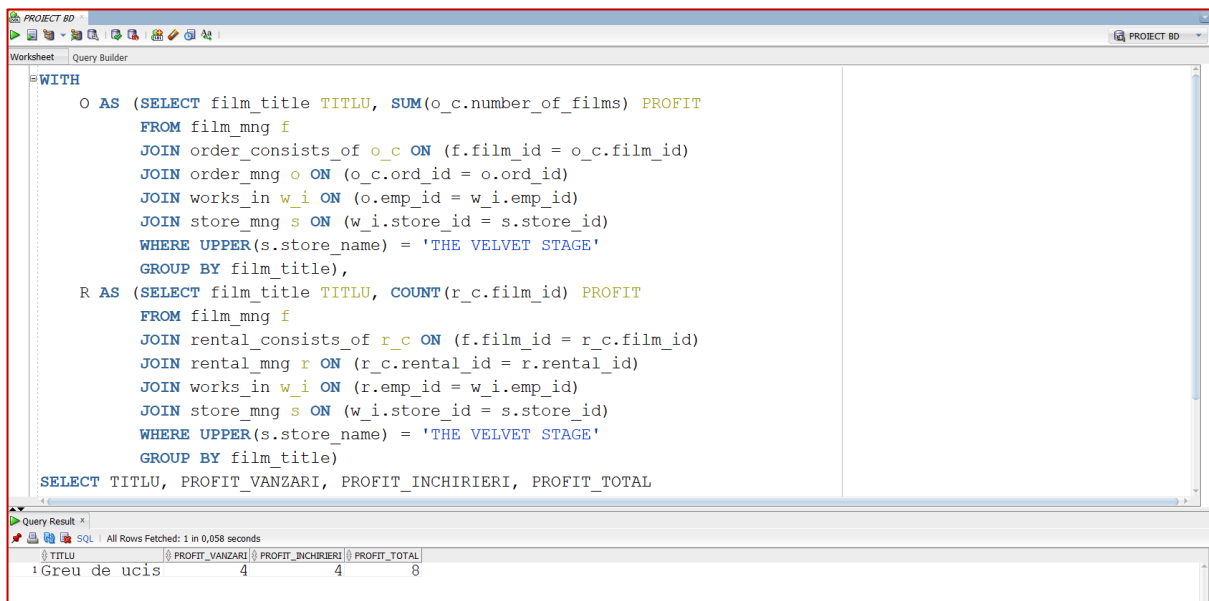
SELECT TITLU, PROFIT_VANZARI, PROFIT_INCHIRIERI, PROFIT_TOTAL
FROM (SELECT O.TITLU TITLU, O.PROFIT PROFIT_VANZARI, R.PROFIT
PROFIT_INCHIRIERI, (O.PROFIT + R.PROFIT) PROFIT_TOTAL

FROM O JOIN R ON (O.TITLU = R.TITLU)

ORDER BY PROFIT_TOTAL DESC)

WHERE ROWNUM = 1;

```



☞ subcerere nesincronizată în clauza FROM, utilizarea a cel puțin un bloc de cerere (clauza WITH), utilizarea unei funcții pe șiruri de caractere.

~ CEREREA 2 ~

Cerință: Să se trimită prin email, în cazul în care a fost completat, un anunț despre maratonul de filme de groază, clienților magazinului „The Velvet Stage Constanta”, care au închiriat cel puțin un film din acest gen.

```
SELECT cust_last NUME, cust_first PRENUME,
```

```

DECODE(NVL(cust_email, NULL), NULL, 'EMAIL NECOMPLETAT', 'ANUNT!:
PE DATA DE 30.10.2022, "THE VELVET STAGE" ORGANIZEAZA UN MARATON DE
FILME HORROR!, EMAIL: ' || cust_email) ANUNT_OFERTA,

```

```

    NUMAR_INCHIRIERI

```

```

FROM customer_mng,

```

```

    (SELECT c.cust_id ID, COUNT(*) NUMAR_INCHIRIERI

```

```

    FROM customer_mng c

```

```

    JOIN rental_mng r ON (c.cust_id = r.cust_id)

```

```

    JOIN employee_mng e ON (r.emp_id = e.emp_id)

```

```

    JOIN works_in w_i ON (e.emp_id = w_i.emp_id)

```

```

    JOIN store_mng s ON (w_i.store_id = s.store_id)

```

```

    JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)

```

```

    JOIN film_mng f ON (r_c.film_id = f.film_id)

```

```

    JOIN category_mng cy ON (f.category_id = cy.category_id)

```

```

    WHERE LOWER(category_name) = 'groaza'

```

```

    AND UPPER(store_name) = 'THE VELVET STAGE CONSTANTA'

```

```

    GROUP BY c.cust_id) R

```

```

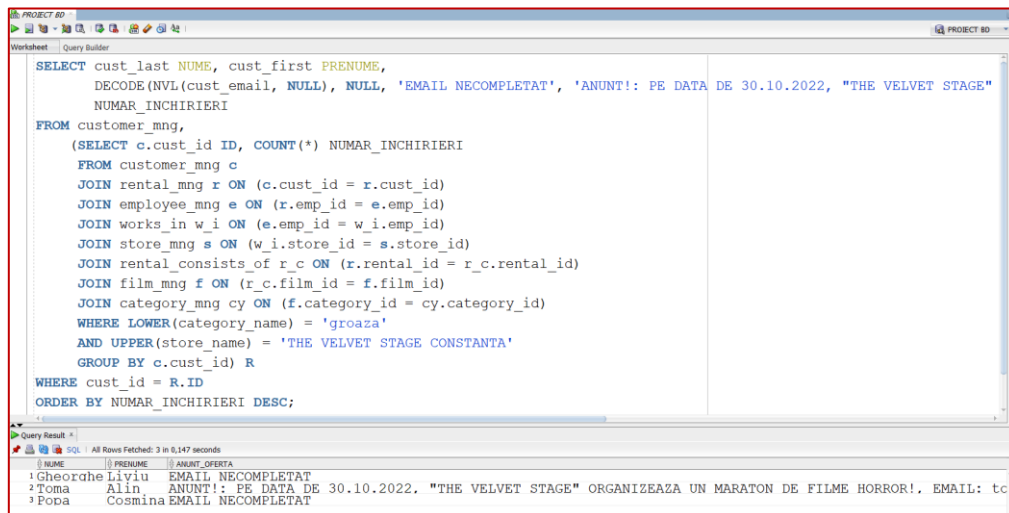
WHERE cust_id = R.ID

```

```

ORDER BY NUMAR_INCHIRIERI DESC;

```



The screenshot shows a SQL query editor window with a query builder interface. The query is as follows:

```

SELECT cust_last NUME, cust_first PRENUME,
       DECODE(NVL(cust_email, NULL), NULL, 'EMAIL NECOMPLETAT', 'ANUNT!: PE DATA DE 30.10.2022, "THE VELVET STAGE"
       NUMAR_INCHIRIERI
FROM customer_mng,
     (SELECT c.cust_id ID, COUNT(*) NUMAR_INCHIRIERI
      FROM customer_mng c
      JOIN rental_mng r ON (c.cust_id = r.cust_id)
      JOIN employee_mng e ON (r.emp_id = e.emp_id)
      JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
      JOIN store_mng s ON (w_i.store_id = s.store_id)
      JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
      JOIN film_mng f ON (r_c.film_id = f.film_id)
      JOIN category_mng cy ON (f.category_id = cy.category_id)
      WHERE LOWER(category_name) = 'groaza'
      AND UPPER(store_name) = 'THE VELVET STAGE CONSTANTA'
      GROUP BY c.cust_id) R
WHERE cust_id = R.ID
ORDER BY NUMAR_INCHIRIERI DESC;

```

The query results are displayed in a table with 3 rows and 3 columns: NUME, PRENUME, and ANUNT_OFERTA.

NUME	PRENUME	ANUNT_OFERTA
Gheorghe	Liviu	EMAIL NECOMPLETAT
Toma	Alin	ANUNT!: PE DATA DE 30.10.2022. "THE VELVET STAGE" ORGANIZEAZA UN MARATON DE FILME HORROR!, EMAIL: tc
Poda	Cosmina	EMAIL NECOMPLETAT

- ☞ grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri), ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri), utilizarea a două funcții pe șiruri de caractere;

~ CEREREA 3 ~

Cerință: Pentru comenzile plasate în perioada 1 – 14 februarie, se va aplica un discount asupra filmelor din categoria „Romantic”:

- ⊗ prețul comenzii mai mic sau egal cu 50 lei, discount-ul va fi în valoare de 10%;
- ⊗ prețul comenzii între 50 și 100 de lei, discount-ul va fi în valoare de 15%;
- ⊗ prețul comenzii între 100 și 200 de lei, discount-ul va fi în valoare de 20%;
- ⊗ prețul comenzii peste 200 de lei, discount-ul va fi în valoare de 25%.

Să se listeze numele clientului, prețul inițial al comenzii, discount-ul ce va fi aplicat asupra filmelor romantice din comandă și noul preț.

```

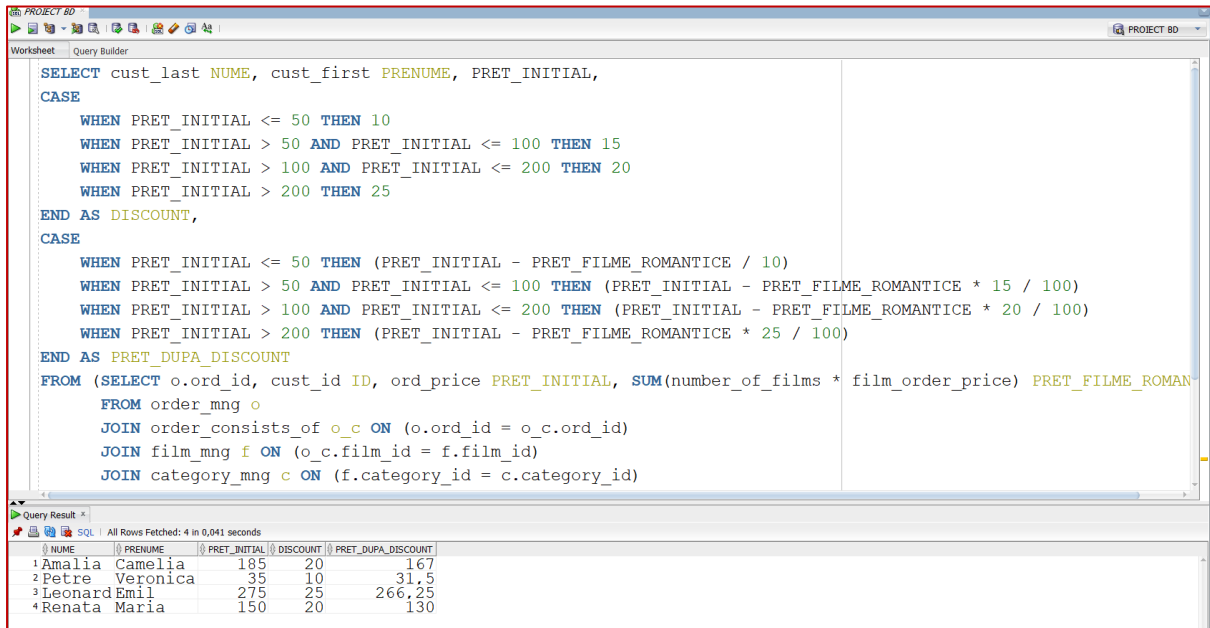
SELECT cust_last NUME, cust_first PRENUME, PRET_INITIAL,
CASE
    WHEN PRET_INITIAL <= 50 THEN 10
    WHEN PRET_INITIAL > 50 AND PRET_INITIAL <= 100 THEN 15
    WHEN PRET_INITIAL > 100 AND PRET_INITIAL <= 200 THEN 20
    WHEN PRET_INITIAL > 200 THEN 25
END AS DISCOUNT,
CASE
    WHEN    PRET_INITIAL    <=    50    THEN    (PRET_INITIAL    -
PRET_FILME_ROMANTICE / 10)
    WHEN PRET_INITIAL > 50 AND PRET_INITIAL <= 100 THEN (PRET_INITIAL
- PRET_FILME_ROMANTICE * 15 / 100)
    WHEN PRET_INITIAL > 100 AND PRET_INITIAL <= 200 THEN (PRET_INITIAL
- PRET_FILME_ROMANTICE * 20 / 100)
    WHEN    PRET_INITIAL    >    200    THEN    (PRET_INITIAL    -
PRET_FILME_ROMANTICE * 25 / 100)
END AS PRET_DUPA_DISCOUNT
FROM (SELECT o.ord_id, cust_id ID, ord_price PRET_INITIAL, SUM(number_of_films *
film_order_price) PRET_FILME_ROMANTICE
    FROM order_mng o
    JOIN order_consists_of o_c ON (o.ord_id = o_c.ord_id)
    JOIN film_mng f ON (o_c.film_id = f.film_id)
    JOIN category_mng c ON (f.category_id = c.category_id)
    WHERE UPPER(category_name) = 'ROMANTIC'

```

```

AND EXTRACT (MONTH FROM ord_date) = 2
AND EXTRACT (DAY FROM ord_date) >= 1 AND EXTRACT (DAY FROM
ord_date) <= 14
GROUP BY o.ord_id, cust_id, ord_price) F_R,
customer_mng c
WHERE cust_id = F_R.ID;

```



The screenshot shows a SQL query editor window titled 'PROJECT 80'. The query is as follows:

```

SELECT cust_last NUME, cust_first PRENUME, PRET_INITIAL,
CASE
  WHEN PRET_INITIAL <= 50 THEN 10
  WHEN PRET_INITIAL > 50 AND PRET_INITIAL <= 100 THEN 15
  WHEN PRET_INITIAL > 100 AND PRET_INITIAL <= 200 THEN 20
  WHEN PRET_INITIAL > 200 THEN 25
END AS DISCOUNT,
CASE
  WHEN PRET_INITIAL <= 50 THEN (PRET_INITIAL - PRET_FILME_ROMANTICE / 10)
  WHEN PRET_INITIAL > 50 AND PRET_INITIAL <= 100 THEN (PRET_INITIAL - PRET_FILME_ROMANTICE * 15 / 100)
  WHEN PRET_INITIAL > 100 AND PRET_INITIAL <= 200 THEN (PRET_INITIAL - PRET_FILME_ROMANTICE * 20 / 100)
  WHEN PRET_INITIAL > 200 THEN (PRET_INITIAL - PRET_FILME_ROMANTICE * 25 / 100)
END AS PRET_DUPA_DISCOUNT
FROM (SELECT o.ord_id, cust_id ID, ord_price PRET_INITIAL, SUM(number_of_films * film_order_price) PRET_FILME_ROMAN
FROM order_mng o
JOIN order_consists_of o_c ON (o.ord_id = o_c.ord_id)
JOIN film_mng f ON (o_c.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)

```

The result table shows the following data:

	NUME	PRENUME	PRET_INITIAL	DISCOUNT	PRET_DUPA_DISCOUNT
1	Amalia	Camelia	185	20	167
2	Petre	Veronica	35	10	31,5
3	Leonard	Emil	275	25	266,25
4	Renata	Maria	150	20	130

- ☞ subcereri nesincronizate în clauza FROM, grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri), 2 funcții pe date calendaristice, utilizarea a cel puțin unei expresii CASE;

~ CEREREA 4 ~

Cerință: Pentru fiecare magazin din franciză, să se listeze numele magazinului, numărul de angajați, numărul de vânzări din ultimul an, filmul ce a fost comandat cel mai mult în ultimul an, numărul de închirieri din ultimul an și filmul ce a fost închiriat de cel mai multe ori.

```

SELECT store_name NUME,
      (SELECT COUNT(*)
      FROM employee_mng e
      JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
      WHERE w_i.store_id = s.store_id) NUMAR_ANGAJATI,

      (SELECT COUNT(*)
      FROM order_mng o
      JOIN employee_mng e ON (o.emp_id = e.emp_id)
      JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
      WHERE w_i.store_id = s.store_id
      AND EXTRACT (YEAR FROM ord_date) = 2022) NUMAR_VANZARI_2022,

      (SELECT 'TITLU: ' || TITLU || ' - NUMAR VANZARI: ' || NUMAR_VANZARI
      FROM store_mng s_m,

      (SELECT      w_i.store_id      ID,      film_title      TITLU,      COUNT(*)
      NUMAR_VANZARI

      FROM film_mng f
      JOIN order_consists_of o_c ON (f.film_id = o_c.film_id)
      JOIN order_mng o ON (o_c.ord_id = o.ord_id)
      JOIN employee_mng e ON (o.emp_id = e.emp_id)
      JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
      WHERE EXTRACT (YEAR FROM ord_date) = 2022
      GROUP BY w_i.store_id, film_title
      ORDER BY ID, NUMAR_VANZARI DESC) F

```

WHERE s_m.store_id = F.ID AND s_m.store_id = s.store_id AND ROWNUM = 1)
FILM_VANZARI_2022,

(SELECT COUNT(*)
FROM rental_mng r
JOIN employee_mng e ON (r.emp_id = e.emp_id)
JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
WHERE w_i.store_id = s.store_id
AND EXTRACT (YEAR FROM rental_date) = 2022)
NUMAR_INCHIRIERI_2022,

(SELECT 'TITLU: ' || TITLU || ' - NUMAR INCHIRIERI: ' || NUMAR_INCHIRIERI
FROM store_mng s_m,
(SELECT w_i.store_id ID, film_title TITLU, COUNT(*)
NUMAR_INCHIRIERI
FROM film_mng f
JOIN rental_consists_of r_c ON (f.film_id = r_c.film_id)
JOIN rental_mng r ON (r_c.rental_id = r.rental_id)
JOIN employee_mng e ON (r.emp_id = e.emp_id)
JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
WHERE EXTRACT (YEAR FROM rental_date) = 2022
GROUP BY w_i.store_id, film_title
ORDER BY ID, NUMAR_INCHIRIERI DESC) F
WHERE s_m.store_id = F.ID AND s_m.store_id = s.store_id AND ROWNUM = 1)
FILM_INCHIRIERI_2022
FROM store_mng s
ORDER BY (NUMAR_VANZARI_2022 + NUMAR_INCHIRIERI_2022) DESC;

- ☞ subcereri sincronizate în care intervin cel puțin 3 tabele, subcereri nesincronizate în clauza FROM, grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri), ordonări, utilizarea unei funcții pe date calendaristice;

The screenshot shows a SQL query builder window with a query and its results. The query is as follows:

```

SELECT store_name NUME,
(SELECT COUNT(*)
FROM employee_mng e
JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
WHERE w_i.store_id = s.store_id) NUMAR_ANGAJATI,

(SELECT COUNT(*)
FROM order_mng o
JOIN employee_mng e ON (o.emp_id = e.emp_id)
JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
WHERE w_i.store_id = s.store_id
AND EXTRACT (YEAR FROM ord_date) = 2022) NUMAR_VANZARI_2022,

(SELECT 'TITLU: ' || TITLU || ' - NUMAR VANZARI: ' || NUMAR_VANZARI
FROM store_mng s_m,
(SELECT w_i.store_id ID, film_title TITLU, COUNT(*) NUMAR_VANZARI
FROM film_mng f

```

The results are displayed in a table with 5 columns: NUME, NUMAR_ANGAJATI, NUMAR_VANZARI_2022, FILM_VANZARI_2022, and FILM_INCHIRIERI_2022. The data is as follows:

NUME	NUMAR_ANGAJATI	NUMAR_VANZARI_2022	FILM_VANZARI_2022	FILM_INCHIRIERI_2022
1 The Velvet Stage Sibiu	6	5	TITLU: Mulan - NUMAR VANZARI: 1	6
2 The Velvet Stage Brasov	6	3	TITLU: Mulan - NUMAR VANZARI: 1	6
3 The Velvet Stage	6	2	TITLU: Greu de ucis - NUMAR VANZARI: 1	5
4 The Velvet Stage Constanta	6	1	TITLU: Stralucirea - NUMAR VANZARI: 1	4
5 The Velvet Stage Cluj-Napoca	6	1	TITLU: Inchisoarea ingerilor - NUMAR VANZARI: 1	1

~ CEREREA 5 ~

Cerință: Pentru fiecare categorie de film, să se listeze atât filmul din acea categorie ce are cel mai mare rating, cât și regizorul acestuia.

```

SELECT film_title TITLU_FILM,

(SELECT director_first || ' ' || director_last

FROM director_mng d

JOIN film_mng f_m ON (d.director_id = f_m.director_id)

WHERE UPPER(f_m.film_title) = UPPER(f.film_title)) REGIZOR,

(SELECT category_name

FROM category_mng

WHERE category_id = R.ID_CATEGORIE) NUME_CATEGORIE,

R.RATING_MAXIM RATING_MAXIM

FROM film_mng f,

```

```

(SELECT MAX(f_m.film_rating) RATING_MAXIM, f_m.category_id
ID_CATEGORIE

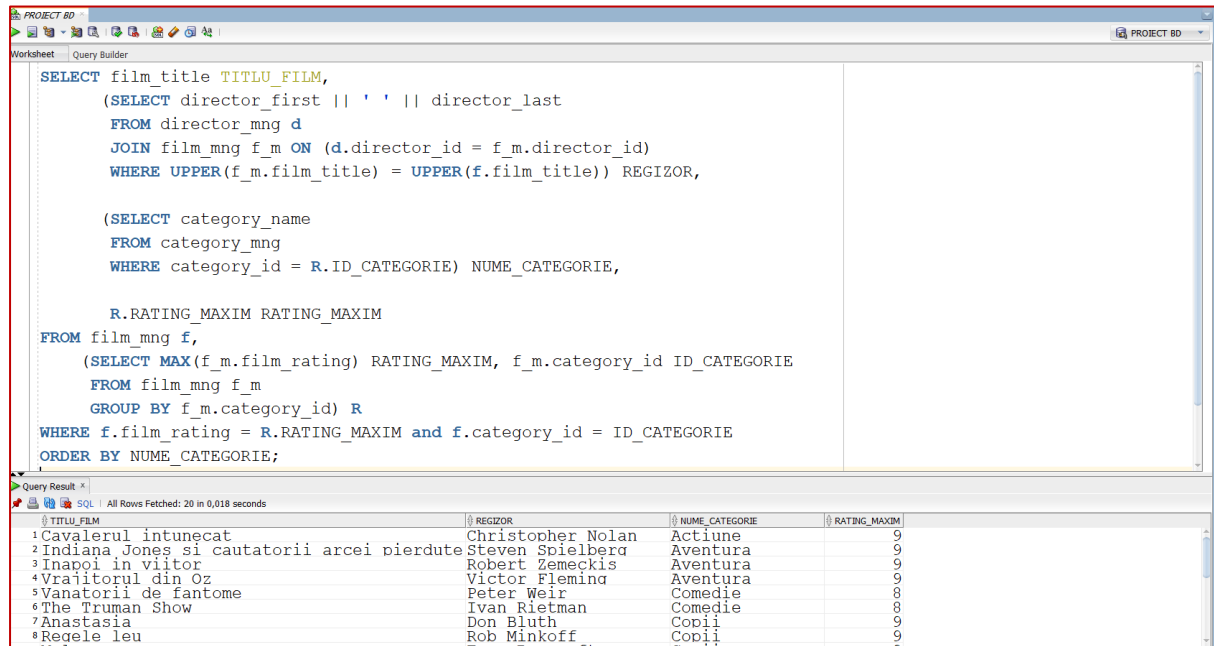
FROM film_mng f_m

GROUP BY f_m.category_id) R

WHERE f.film_rating = R.RATING_MAXIM and f.category_id = ID_CATEGORIE

ORDER BY NUME_CATEGORIE;

```



Query Result: All Rows Fetched: 20 in 0,018 seconds

TITLU_FILM	REGIZOR	NUME_CATEGORIE	RATING_MAXIM
1 Cavalerul întunecat	Christopher Nolan	Actiune	9
2 Indiana Jones si cautatorii arcei pierdute	Steven Spielberg	Aventura	9
3 Inapoi in viitor	Robert Zemeckis	Aventura	9
4 Vrajitorul din Oz	Victor Fleming	Aventura	9
5 Vanatorii de fantome	Peter Weir	Comedie	8
6 The Truman Show	Ivan Reitman	Comedie	8
7 Anastasia	Don Bluth	Copii	9
8 Regele leu	Rob Minkoff	Copii	9

- ☞ subcereri nesincronizate în clauza FROM, utilizarea unei funcții pe șiruri de caractere, ordonări, grupări, funcții grup.

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

~ OPERAȚIA 1 ~

Cerință: Să se actualizeze prețul unei închirieri care are în componența sa un film din categoria „Muzical”, astfel:

- ⊗ prețul închirierii mai mic sau egal cu 30 lei, discount-ul va fi în valoare de 10%;
- ⊗ prețul închirierii între 30 și 70 de lei, discount-ul va fi în valoare de 15%;
- ⊗ prețul închirierii între 70 și 100 de lei, discount-ul va fi în valoare de 20%;

```
UPDATE rental_mng
SET rental_price =
    (CASE
        WHEN rental_price <= 30 THEN rental_price - rental_price * 10 / 100
        WHEN rental_price > 30 AND rental_price <= 70 THEN rental_price -
rental_price * 15 / 100
        WHEN rental_price > 70 AND rental_price <= 100 THEN rental_price -
rental_price * 20 / 100
        WHEN rental_price > 100 THEN rental_price - rental_price * 25 / 100
    END)
WHERE rental_id IN (SELECT r.rental_id
                    FROM rental_mng r
                    JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
                    JOIN film_mng f ON (r_c.film_id = f.film_id)
                    JOIN category_mng c ON (f.category_id = c.category_id)
                    WHERE UPPER(category_name) = 'MUZICAL');
```

The screenshot shows the SQL Developer interface with a query window containing an UPDATE statement. The query updates the rental_price in the rental_mng table based on a CASE statement. The WHERE clause filters for musical films. The results pane shows 15 rows with their initial rental prices.

```

UPDATE rental_mng
SET rental_price =
(
CASE
WHEN rental_price <= 30 THEN rental_price - rental_price * 10 / 100
WHEN rental_price > 30 AND rental_price <= 70 THEN rental_price - rental_price * 15 / 100
WHEN rental_price > 70 AND rental_price <= 100 THEN rental_price - rental_price * 20 / 100
WHEN rental_price > 100 THEN rental_price - rental_price * 25 / 100
END)
WHERE rental_id IN (SELECT r.rental_id
FROM rental_mng r
JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
JOIN film_mng f ON (r_c.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE UPPER(category_name) = 'MUZICAL');

```

ID	PRICE
1 3007	35
2 3008	20
3 3012	35
4 3023	55
5 3025	50
6 3030	95
7 3030	95
8 3043	15
9 3050	65
10 3063	20
11 3074	20

~ valorile inițiale ~

The screenshot shows the same SQL Developer interface with the same UPDATE query. Below the query window, a message states "14 rows updated.".

```

UPDATE rental_mng
SET rental_price =
(
CASE
WHEN rental_price <= 30 THEN rental_price - rental_price * 10 / 100
WHEN rental_price > 30 AND rental_price <= 70 THEN rental_price - rental_price * 15 / 100
WHEN rental_price > 70 AND rental_price <= 100 THEN rental_price - rental_price * 20 / 100
WHEN rental_price > 100 THEN rental_price - rental_price * 25 / 100
END)
WHERE rental_id IN (SELECT r.rental_id
FROM rental_mng r
JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
JOIN film_mng f ON (r_c.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE UPPER(category_name) = 'MUZICAL');

```

14 rows updated.

The screenshot shows the same SQL Developer interface with the same UPDATE query. The results pane now shows the updated rental prices for the 11 rows.

```

UPDATE rental_mng
SET rental_price =
(
CASE
WHEN rental_price <= 30 THEN rental_price - rental_price * 10 / 100
WHEN rental_price > 30 AND rental_price <= 70 THEN rental_price - rental_price * 15 / 100
WHEN rental_price > 70 AND rental_price <= 100 THEN rental_price - rental_price * 20 / 100
WHEN rental_price > 100 THEN rental_price - rental_price * 25 / 100
END)
WHERE rental_id IN (SELECT r.rental_id
FROM rental_mng r
JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
JOIN film_mng f ON (r_c.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE UPPER(category_name) = 'MUZICAL');

```

ID	PRICE
1 3007	30
2 3008	18
3 3012	30
4 3023	47
5 3025	43
6 3030	76
7 3030	76
8 3043	14
9 3050	55
10 3063	18
11 3074	18

~ valorile după update ~

~ OPERAȚIA 2 ~

Cerință: Pentru Ziua Mamei și Ziua Femeii, „The Velvet Stage” mărește data de retur a închirierilor făcute în perioada 22 februarie – 8 martie. Să se actualizeze valoarea datei de retur pentru aceste închirieri cu o săptămână în plus.

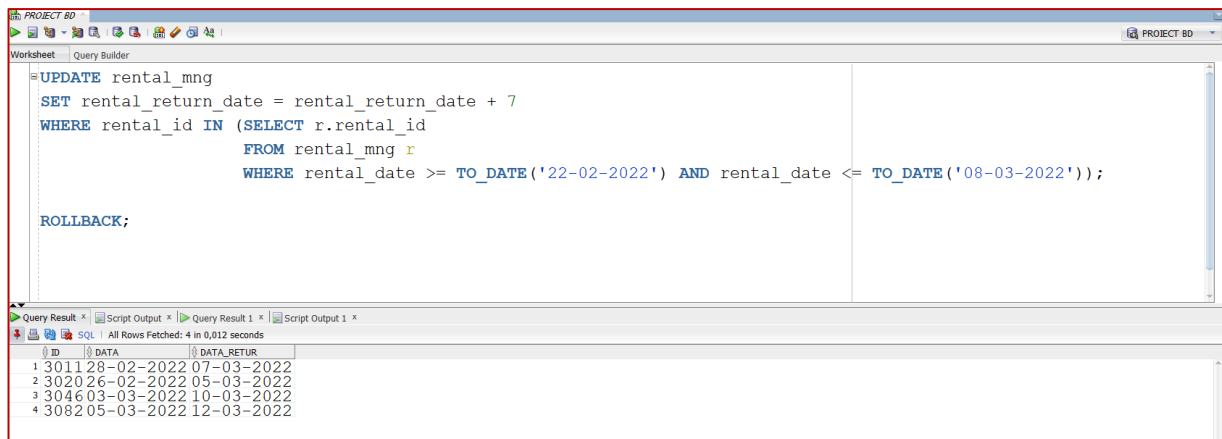
```
UPDATE rental_mng
```

```
SET rental_return_date = rental_return_date + 7
```

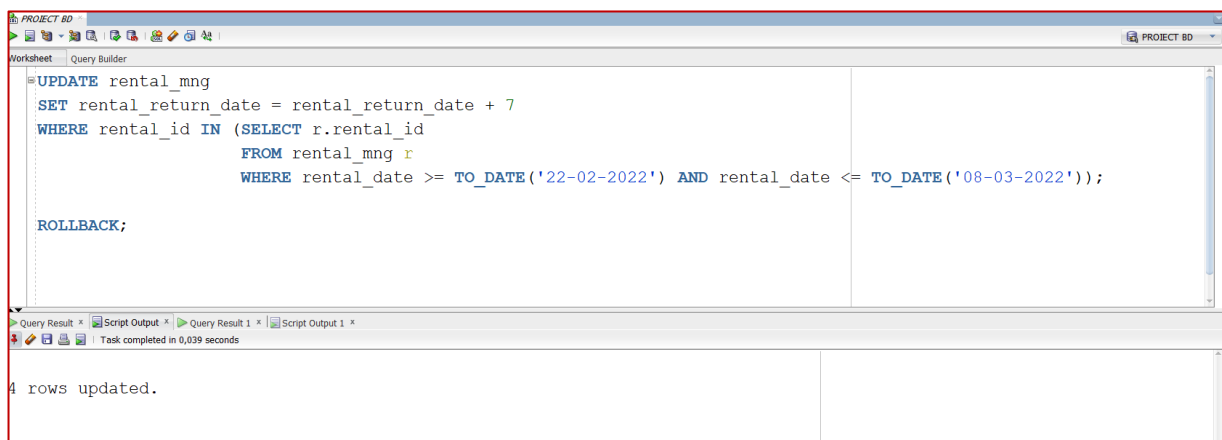
```
WHERE rental_id IN (SELECT r.rental_id
```

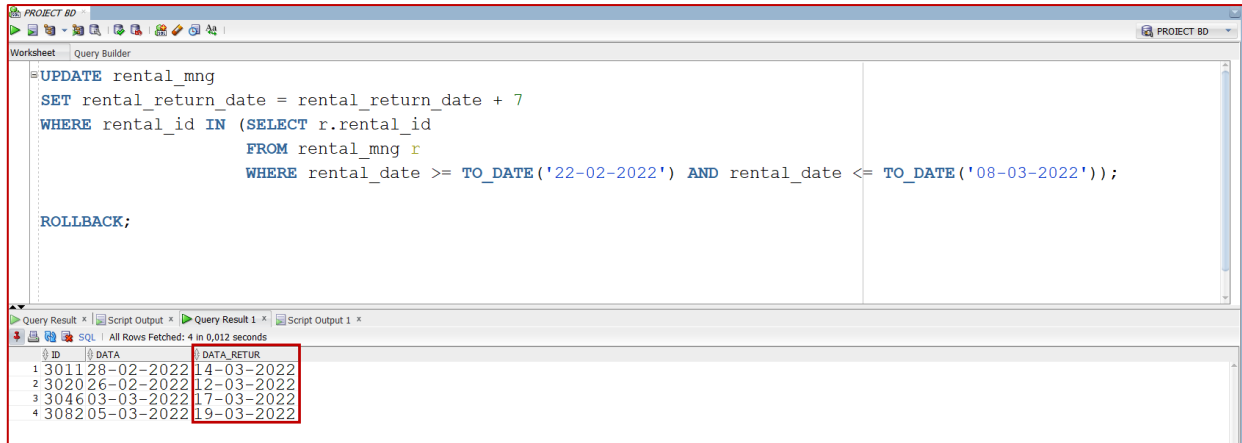
```
FROM rental_mng r
```

```
WHERE rental_date >= TO_DATE('22-02-2022') AND rental_date <= TO_DATE('08-03-2022'));
```



~ valorile inițiale ~

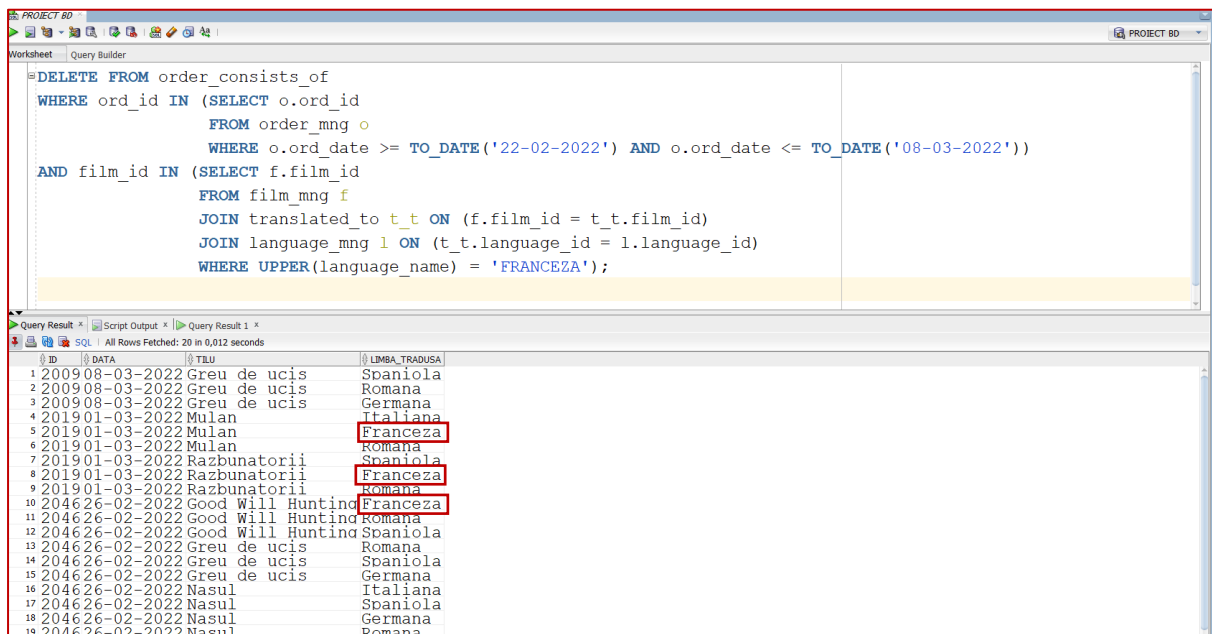




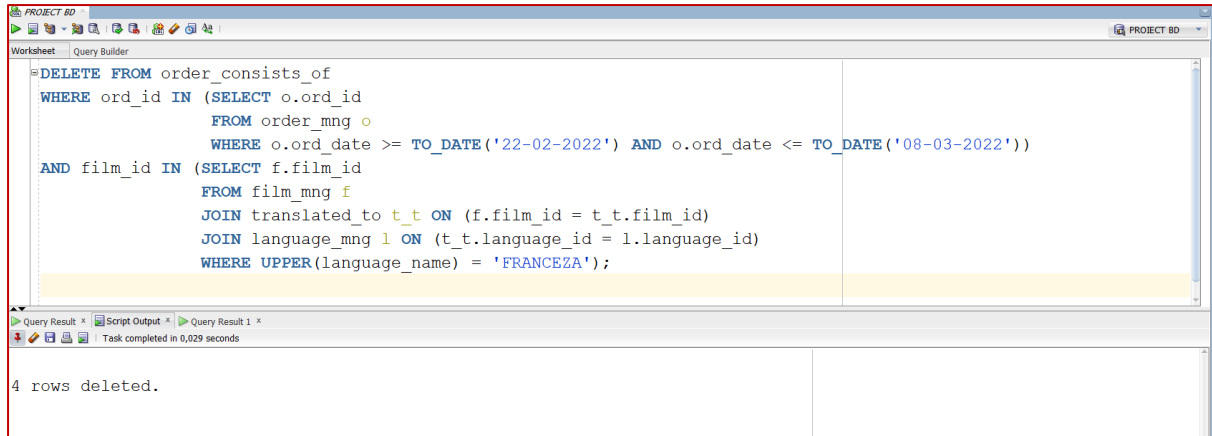
~ valorile după update ~

~ OPERAȚIA 3 ~

Cerință: Să se șteargă din comenzile făcute în perioada 22 februarie – 8 martie, filmele care au traduceri în limba franceză.



~ valorile inițiale ~

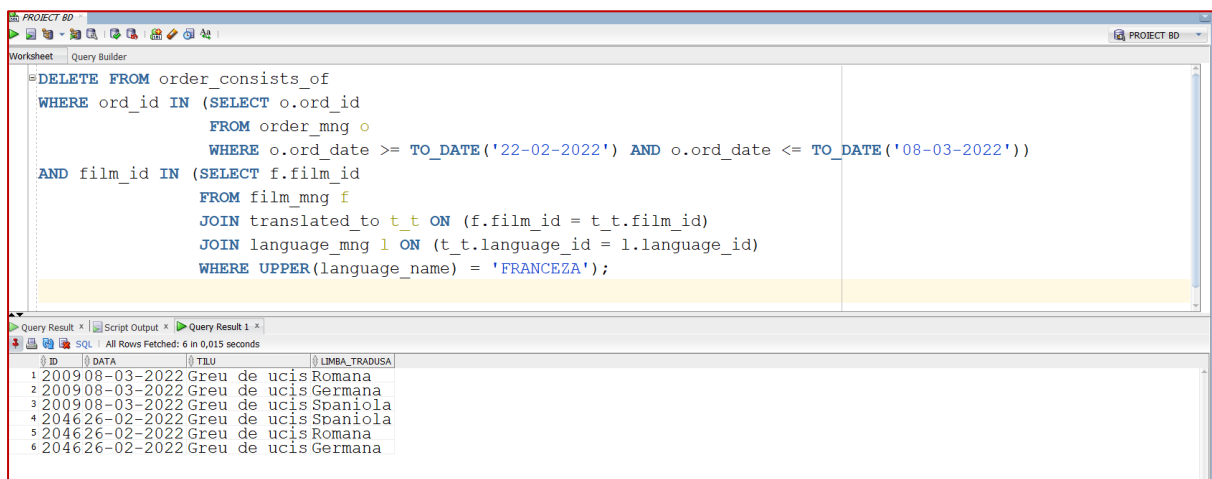


The screenshot shows the SQL Developer interface with a query window titled 'PROJECT BD'. The query is a DELETE statement targeting the 'order_consists_of' table. The WHERE clause uses subqueries to filter records based on 'ord_id' and 'film_id' from other tables, with additional filters for dates and language. The status bar indicates 'Task completed in 0,029 seconds'. The query result pane shows '4 rows deleted.'

```

DELETE FROM order_consists_of
WHERE ord_id IN (SELECT o.ord_id
                  FROM order_mng o
                  WHERE o.ord_date >= TO_DATE('22-02-2022') AND o.ord_date <= TO_DATE('08-03-2022'))
AND film_id IN (SELECT f.film_id
                FROM film_mng f
                JOIN translated_to t_t ON (f.film_id = t_t.film_id)
                JOIN language_mng l ON (t_t.language_id = l.language_id)
                WHERE UPPER(language_name) = 'FRANCEZA');
    
```

4 rows deleted.



This screenshot shows the same SQL Developer interface with the same DELETE query. The status bar now indicates 'All Rows Fetched: 6 in 0,015 seconds'. The query result pane displays a table with 6 rows of data, including columns for ID, DATA, TIU, and LIMBA_TRADUSA.

```

DELETE FROM order_consists_of
WHERE ord_id IN (SELECT o.ord_id
                  FROM order_mng o
                  WHERE o.ord_date >= TO_DATE('22-02-2022') AND o.ord_date <= TO_DATE('08-03-2022'))
AND film_id IN (SELECT f.film_id
                FROM film_mng f
                JOIN translated_to t_t ON (f.film_id = t_t.film_id)
                JOIN language_mng l ON (t_t.language_id = l.language_id)
                WHERE UPPER(language_name) = 'FRANCEZA');
    
```

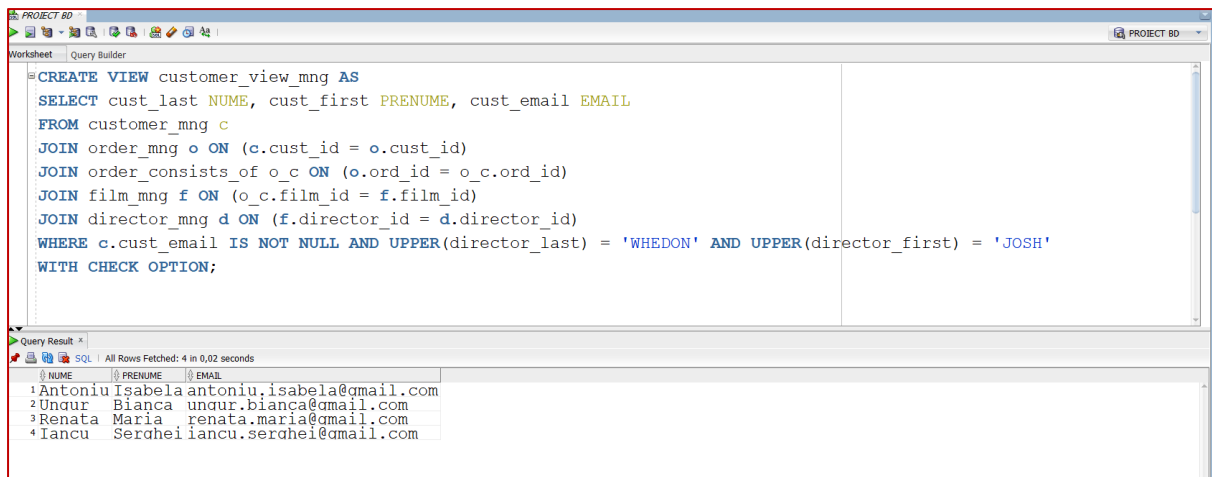
ID	DATA	TIU	LIMBA_TRADUSA
1	200908-03-2022	Greu de ucis	Romana
2	200908-03-2022	Greu de ucis	Germana
3	200908-03-2022	Greu de ucis	Spaniola
4	204626-02-2022	Greu de ucis	Spaniola
5	204626-02-2022	Greu de ucis	Romana
6	204626-02-2022	Greu de ucis	Germana

~ valorile după delete ~

14. Implementarea unei vizualizări complexe.

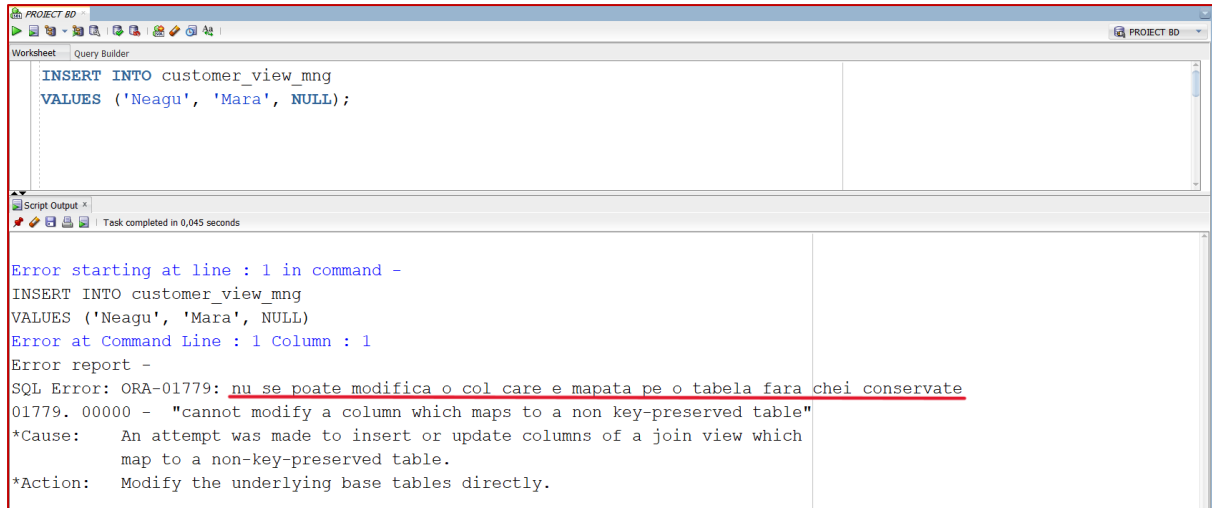
Cerință: Să se implementeze o vizualizare care să listeze clienții care au comandat cel puțin un film regizat de Josh Whedon.

```
CREATE VIEW customer_view_mng AS
SELECT cust_last NUME, cust_first PRENUME, cust_email EMAIL
FROM customer_mng c
JOIN order_mng o ON (c.cust_id = o.cust_id)
JOIN order_consists_of o_c ON (o.ord_id = o_c.ord_id)
JOIN film_mng f ON (o_c.film_id = f.film_id)
JOIN director_mng d ON (f.director_id = d.director_id)
WHERE c.cust_email IS NOT NULL AND UPPER(director_last) = 'WHEDON' AND
UPPER(director_first) = 'JOSH'
WITH CHECK OPTION;
```



```
INSERT INTO customer_view_mng
VALUES ('Neagu', 'Mara', NULL);
```

Această operație va produce o eroare, deoarece o condiție ca o vizualizare a cărei cerere conține un JOIN să fie modificabilă este ca instrucțiunea LMD să afecteze un singur tabel din operația JOIN.



```

INSERT INTO customer_view_mng
VALUES ('Neagu', 'Mara', NULL);

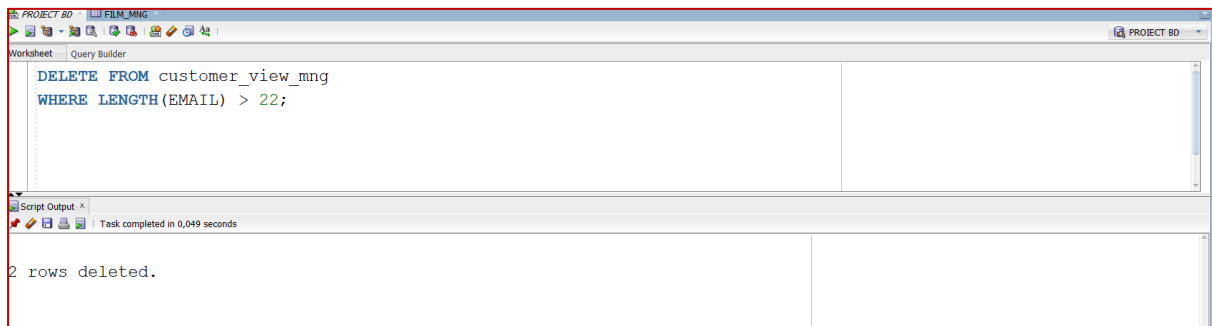
```

Script Output

Task completed in 0,045 seconds

Error starting at line : 1 in command -
 INSERT INTO customer_view_mng
 VALUES ('Neagu', 'Mara', NULL)
 Error at Command Line : 1 Column : 1
 Error report -
 SQL Error: ORA-01779: nu se poate modifica o col care e mapata pe o tabela fara chei conservate
 01779. 00000 - "cannot modify a column which maps to a non key-preserved table"
 *Cause: An attempt was made to insert or update columns of a join view which
 map to a non-key-preserved table.
 *Action: Modify the underlying base tables directly.

DELETE FROM customer_view_mng
 WHERE LENGTH(EMAIL) > 22;



```

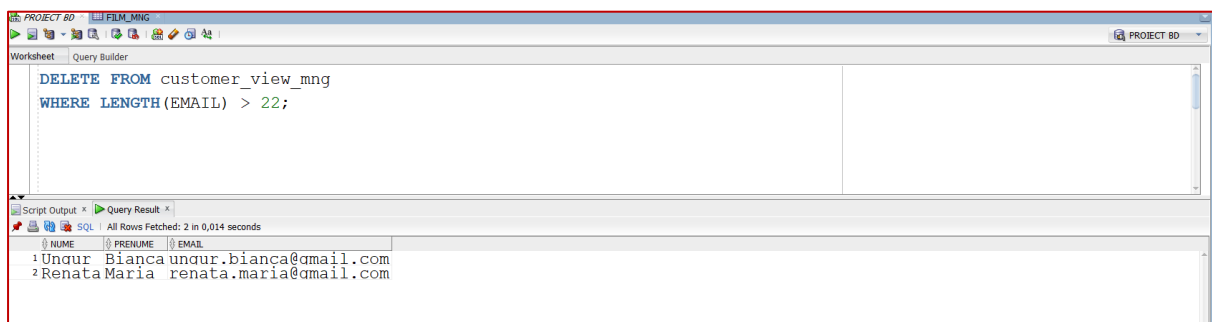
DELETE FROM customer_view_mng
WHERE LENGTH(EMAIL) > 22;

```

Script Output

Task completed in 0,049 seconds

2 rows deleted.



```

DELETE FROM customer_view_mng
WHERE LENGTH(EMAIL) > 22;

```

Script Output

Query Result

SQL | All Rows Fetched: 2 in 0,014 seconds

	NAME	PRENOME	EMAIL
1	Undur	Bianca	undur.bianca@gmail.com
2	RenataMaria	renata	renata.maria@gmail.com

15. Cerere ce utilizează outer – join pe 4 tabele, cerere ce utilizează operația division, cerere ce implementează analiză top – n.

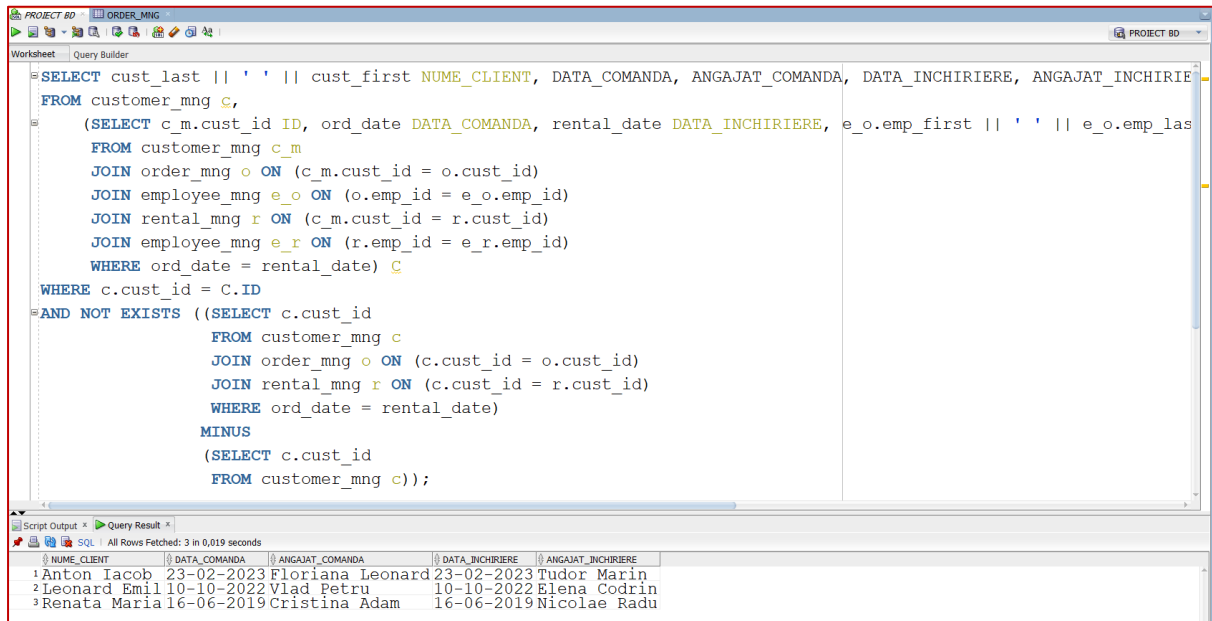
~ DIVISION ~

Cerință: Să se listeze clienții care au dat o comandă în aceeași zi cu o închirierea, împreună cu următoarele informații:

- ☼ numele și prenumele clientului;
- ☼ data în care a fost procesată comanda și angajatul care a procesat-o;
- ☼ data în care a fost procesată închirierea și angajatul care a procesat-o.

```
SELECT cust_last || ' ' || cust_first NUME_CLIENT, DATA_COMANDA,
ANGAJAT_COMANDA, DATA_INCHIRIERE, ANGAJAT_INCHIRIERE
FROM customer_mng c,
      (SELECT c_m.cust_id ID, ord_date DATA_COMANDA, rental_date
DATA_INCHIRIERE, e_o.emp_first || ' ' || e_o.emp_last ANGAJAT_COMANDA,
e_r.emp_first || ' ' || e_r.emp_last ANGAJAT_INCHIRIERE
      FROM customer_mng c_m
      JOIN order_mng o ON (c_m.cust_id = o.cust_id)
      JOIN employee_mng e_o ON (o.emp_id = e_o.emp_id)
      JOIN rental_mng r ON (c_m.cust_id = r.cust_id)
      JOIN employee_mng e_r ON (r.emp_id = e_r.emp_id)
      WHERE ord_date = rental_date) C
WHERE c.cust_id = C.ID
AND NOT EXISTS ((SELECT c.cust_id
                  FROM customer_mng c
                  JOIN order_mng o ON (c.cust_id = o.cust_id)
                  JOIN rental_mng r ON (c.cust_id = r.cust_id)
                  WHERE ord_date = rental_date)
                MINUS
                (SELECT c.cust_id
```


FROM customer_mng c));



~ ANALIZA TOP – N ~

Cerință: Pentru fiecare magazin din franciză, să se listeze clientul ce a plasat cel mai mare număr de comenzi.

SELECT store_name,

(SELECT 'NUME CLIENT: ' || NUME_CLIENT || ' - NUMAR VANZARI: ' ||
NUMAR_COMENZI

FROM (SELECT w_i.store_id ID_MAGAZIN, c.cust_first || ' ' || c.cust_last
NUME_CLIENT, COUNT(o.cust_id) NUMAR_COMENZI

FROM customer_mng c

JOIN order_mng o ON (c.cust_id = o.cust_id)

JOIN employee_mng e ON (o.emp_id = e.emp_id)

JOIN works_in w_i ON (e.emp_id = w_i.emp_id)

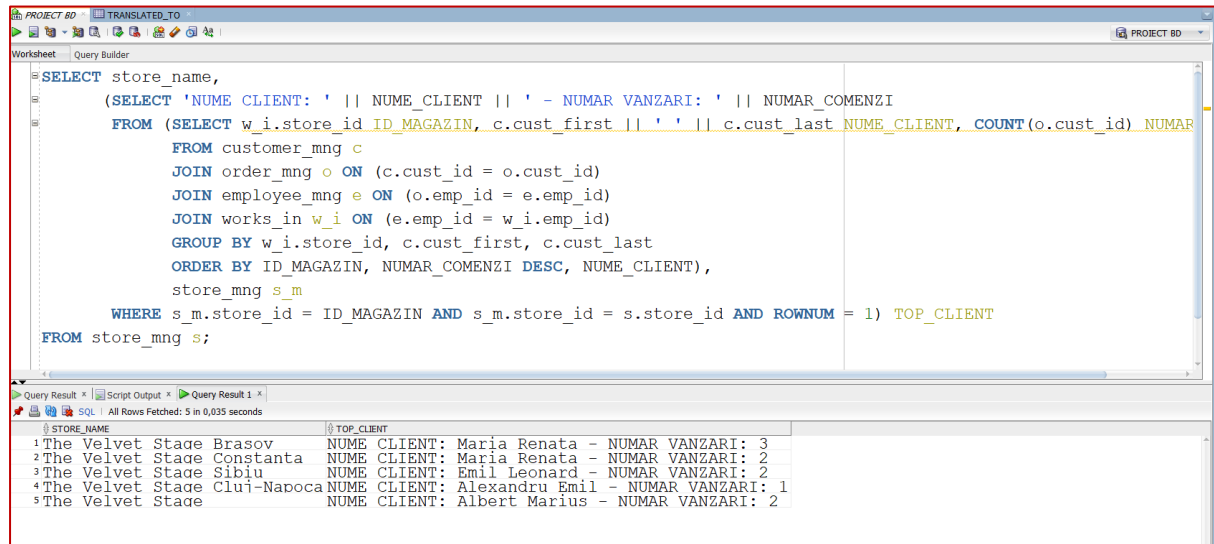
GROUP BY w_i.store_id, c.cust_first, c.cust_last

```
ORDER BY ID_MAGAZIN, NUMAR_COMENZI DESC,
NUME_CLIENT),
```

```
store_mng s_m
```

```
WHERE s_m.store_id = ID_MAGAZIN AND s_m.store_id = s.store_id AND
ROWNUM = 1) TOP_CLIENT
```

```
FROM store_mng s;
```



~ OUTER – JOIN ~

Cerință: Pentru fiecare client, să se listeze numărul de închirieri făcute de acesta care conțin cel puțin un film tradus în limba spaniolă, inclusiv acei clienți care nu au făcut nicio închiriere în general sau nicio închiriere care respectă această condiție.

```
SELECT cust_last || ' ' || cust_first NUME_CLIENT, SUM(CASE WHEN
UPPER(language_name) = 'SPANIOLA' THEN 1 ELSE 0 END) NUMAR_INCHIRIERI
```

```
FROM customer_mng c
```

```
LEFT OUTER JOIN rental_mng r ON (c.cust_id = r.cust_id)
```

```
LEFT OUTER JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
```

```
LEFT OUTER JOIN film_mng f_r ON (r_c.film_id = f_r.film_id)
```

```
LEFT OUTER JOIN translated_to t_t ON (f_r.film_id = t_t.film_id)
```

LEFT OUTER JOIN language_mng l ON (t_t.language_id = l.language_id)

GROUP BY cust_last, cust_first

ORDER BY NUMAR_INCHIRIERI;

Query Builder

```

SELECT cust_last || ' ' || cust_first NUME_CLIENT, SUM(CASE WHEN UPPER(language_name) = 'SPANIOLA' THEN 1 ELSE 0 END) NUMAR_INCHIRIERI
FROM customer_mng c
LEFT OUTER JOIN rental_mng r ON (c.cust_id = r.cust_id)
LEFT OUTER JOIN rental_consists_of r_c ON (r.rental_id = r_c.rental_id)
LEFT OUTER JOIN film_mng f_r ON (r_c.film_id = f_r.film_id)
LEFT OUTER JOIN translated_to t_t ON (f_r.film_id = t_t.film_id)
LEFT OUTER JOIN language_mng l ON (t_t.language_id = l.language_id)
GROUP BY cust_last, cust_first
ORDER BY NUMAR_INCHIRIERI;

```

Query Result: All Rows Fetched: 35 in 0,018 seconds

	NUME_CLIENT	NUMAR_INCHIRIERI
1	Maada Mihaela	0
2	Renata Maria	0
3	Albert Lilliana	0
4	Stan Larisa	0
5	Funar Otilia	1
6	Lunqu Gabriela	1
7	David Amalia	1
8	Antonescu Monica	1
9	Leonard Emil	1
10	Solomon Constantina	1
11	Gavril Vlad	1
12	Unsur Bianca	1
13	Iancu Serdhei	2
14	Petran Elena	2
15	Eduard Carmen	2
16	Lupei Bogdana	2
17	Amalia Camelia	2

16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale.

Cerință: Să se listeze numele filmelor ce fac parte din stocul magazinului "The Velvet Stage", din categoria "Comedie".

~ NEOPTIMIZAT ~

Relația algebrică:

R1 = SELECT(STORE_MNG, store_name = 'THE VELVET STAGE')

R2 = SEMIJOIN(STORE_INVENTORY, R1)

R3 = SEMIJOIN(FILM_MNG, R2)

R4 = PROJECT(R3, film_id, film_title)

R5 = SELECT(CATEGORY_MNG, category_name = 'COMEDIE')

R6 = SEMIJOIN(FILM_MNG, R5)

R7 = PROJECT(R6, film_id, film_title)

R8 = INTERSECT(R4, R7)

REZULTAT = R9 = PROJECT(R8, film_title)

Cod SQL:

WITH

M AS (SELECT f.film_id, film_title

FROM film_mng f

JOIN store_inventory s_i ON (f.film_id = s_i.film_id)

JOIN store_mng s ON (s_i.store_id = s.store_id)

WHERE UPPER(store_name) = 'THE VELVET STAGE'),

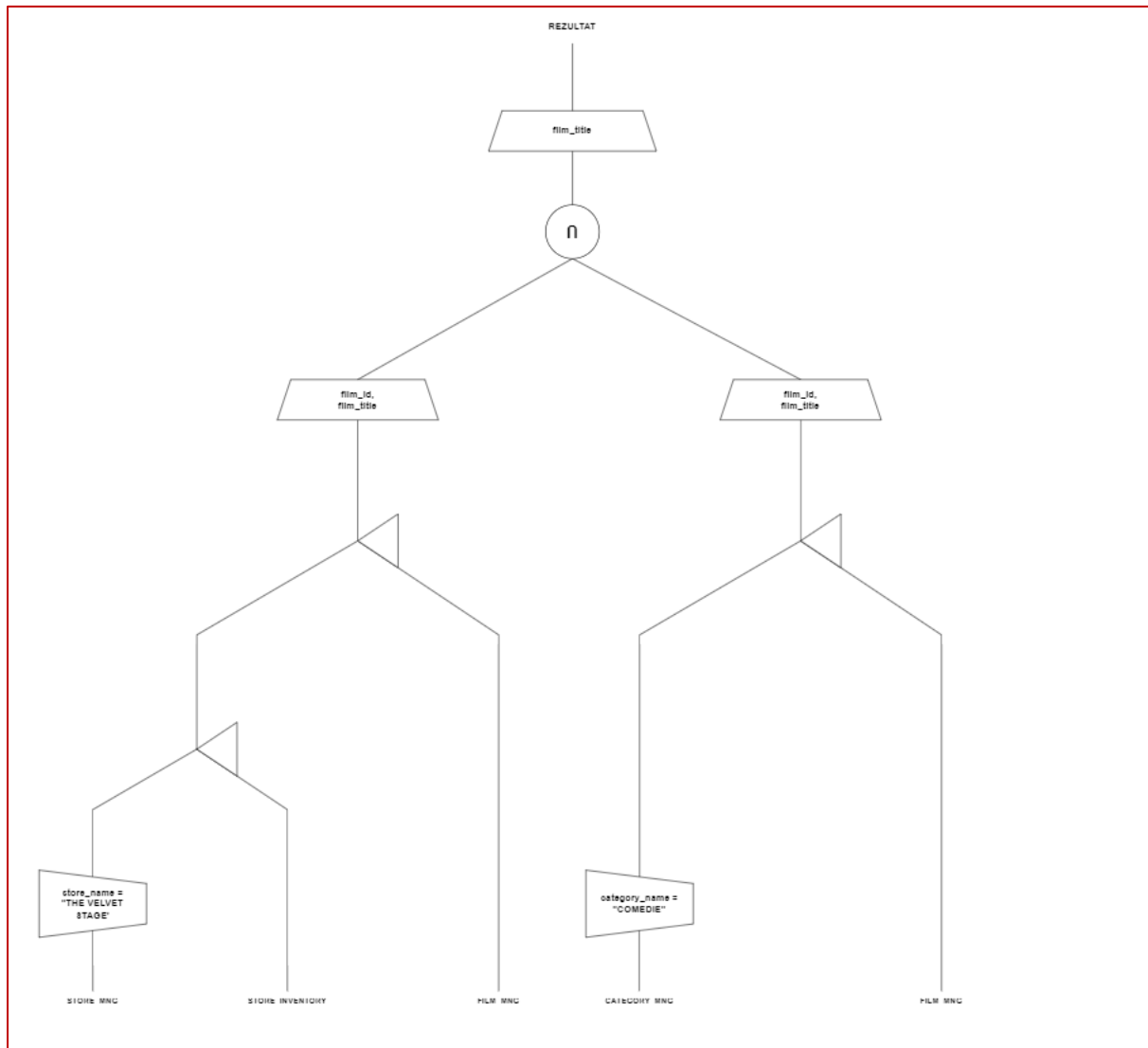
C AS (SELECT film_id, film_title

FROM film_mng f

JOIN category_mng c ON (f.category_id = c.category_id)

```
WHERE UPPER(category_name) = 'COMEDIE')  
  
SELECT M.film_title TITLU_FILM  
  
FROM M JOIN C ON (M.film_id = C.film_id);
```

Arbore algebric:



~ OPTIMIZAT ~

Relație algebrică:

```
R1 = SELECT(STORE_MNG, store_name = 'THE VELVET STAGE')
```

$R2 = \text{SEMIJOIN}(\text{STORE_INVENTORY}, R1)$
 $R3 = \text{SEMIJOIN}(\text{FILM_MNG}, R2)$
 $R4 = \text{PROJECT}(R3, \text{film_id}, \text{film_title}, \text{category_id})$
 $R5 = \text{SELECT}(\text{CATEGORY_MNG}, \text{category_name} = \text{'COMEDIE'})$
 $R6 = \text{PROJECT}(R5, \text{category_id})$
 $R7 = \text{SEMIJOIN}(R4, R6)$
 $\text{REZULTAT} = R8 = \text{PROJECT}(R7, \text{film_title})$

Cod SQL:

```

SELECT film_title
FROM film_mng
WHERE film_id IN (SELECT s_i.film_id FROM store_inventory s_i JOIN store_mng s ON
(s_i.store_id = s.store_id) WHERE UPPER(store_name) = 'THE VELVET STAGE')
AND category_id IN (SELECT c.category_id FROM category_mng c WHERE
UPPER(category_name) = 'COMEDIE');
    
```

Arbore algebric:

