

# PROIECT SISTEME DE GESTIUNE A BAZELOR DE DATE

## **TEMĂ: GESTIUNEA UNUI LANȚ DE MAGAZINE DE DVD-URI**

LECTOR MIHAI GABRIELA

STUDENTĂ: NEAGU MARA TEODORA

SERIA 23, GRUPA 231

ANUL II, SEMESTRUL I

2023

## CUPRINS

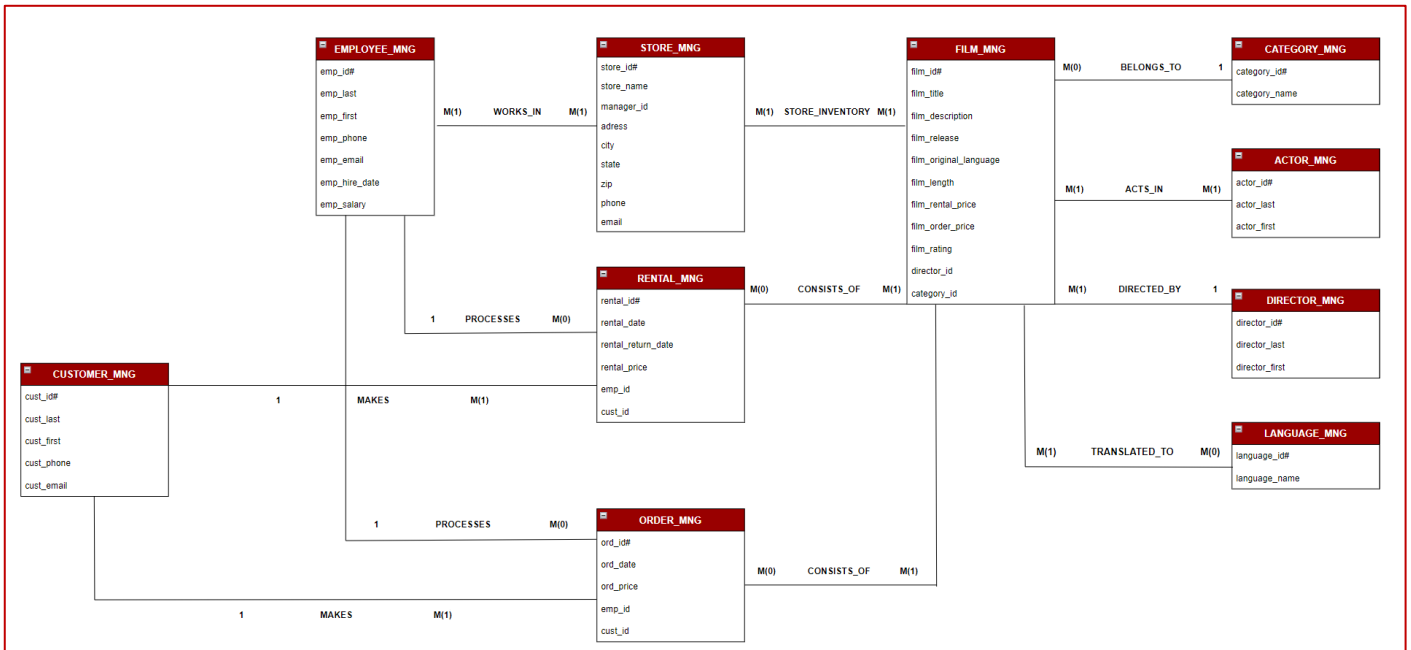
1. Prezentarea pe scurt a bazei de date.
2. Diagrama entitate – relație (ERD).
3. Diagrama conceptuală.
4. Definirea tabelor; implementarea constrângerilor.
5. Adăugarea informațiilor coerente în tabele.
6. Subprogram stocat independent care utilizează două tipuri diferite de colecții.
7. Subprogram stocat independent care utilizează două tipuri diferite de cursoare.
8. Subprogram stocat independent de tip funcție, utilizând 3 tabele definite într-o singură comandă SQL.
9. Subprogram stocat independent de tip procedură, utilizând 5 tabele definite într-o singură comandă SQL.
10. Trigger de tip LMD la nivel de comandă.
11. Trigger de tip LMD la nivel de linie.
12. Trigger de tip LDD.
13. Pachet care conține toate obiectele definite în cadrul proiectului.
14. Pachet care include tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate.

## 1. Prezentarea pe scurt a bazei de date.

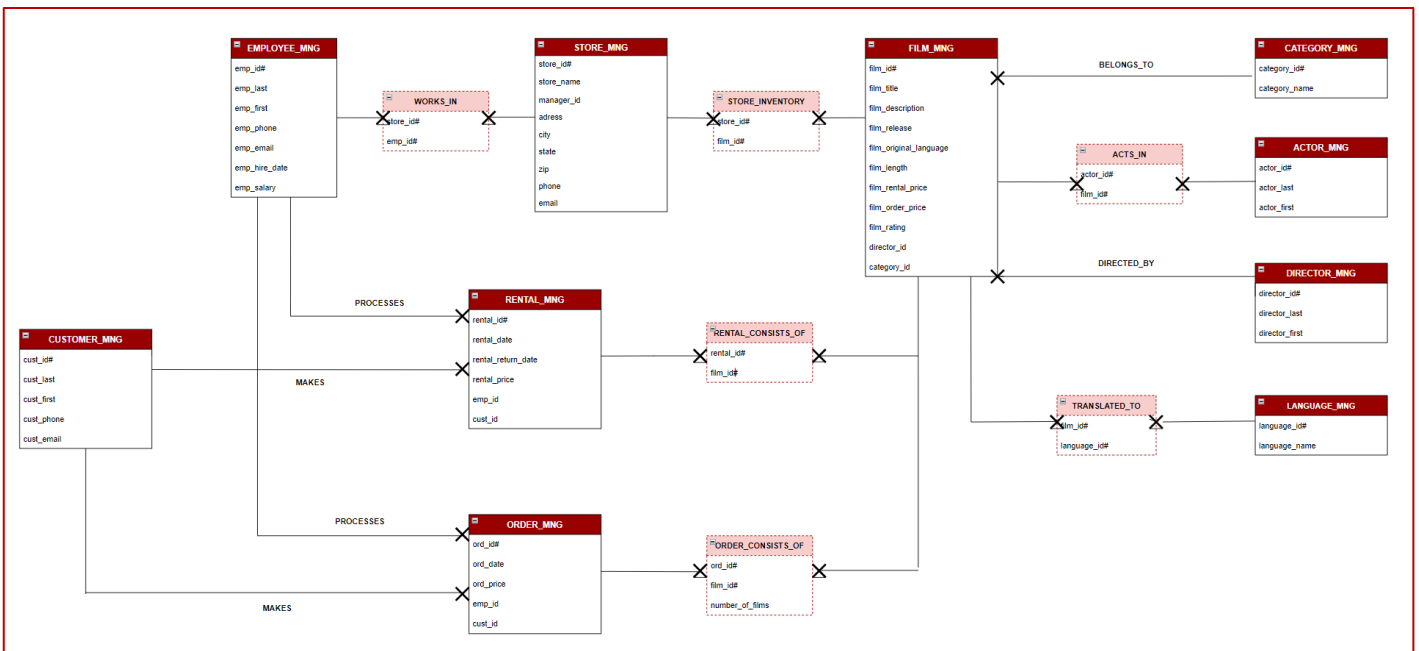
În acest proiect vom analiza gestiunea unui lanț de magazine de DVD-uri. Acesta va ajuta la o mai bună funcționare, la posibilitatea de a ține evidența atât a comenzilor plasate în magazine, cât și a închirierilor de filme. Regulile de funcționare pentru acest model sunt următoarele:

- ⊗ Fiind o franciză, magazinele vor avea numele comun: „*The Velvet Stage*“. Totuși, acestea pot avea variații în funcție oraș sau județ; astfel, numele este consemnat în baza de date.
- ⊗ Un magazin trebuie să aibă minim un film în inventarul lor.
- ⊗ Un angajat poate lucra în mai multe magazine; fiecare magazin trebuie să aibă minim un angajat.
- ⊗ Angajații magazinului 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 unei comenzi 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.
- ⊗ Filmele pot fi încadrate într-o singură categorie.
- ⊗ Categoriile de filme sunt: *ACȚIUNE*, *AVENTURĂ*, *COMEDIE*, *COPII*, *DRAMATIC*, *GROAZĂ*, *MUZICALE*, *POLIȚISTE*, *ROMANTICE*, *SCIENCE-FICTION*.
- ⊗ Filmele pot fi regizate de un singur regizor.
- ⊗ Rating-ul unui film poate fi între 1 și 10.

## 2. Diagrama entitate – relație (ERD).



## 3. Diagrama conceptuală.



## 4. Definirea tabelelor; implementarea constrângerilor.

```
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
NOCYCLE;
```

Below the script, the 'Script Output' window displays the message: 'Sequence SEQUENCE\_STORE\_MNG created.' and 'Task completed in 0,05 seconds'.

```
CREATE TABLE store_mng(
  store_id    NUMBER(5)    PRIMARY KEY,
  store_name  VARCHAR2(50) NOT NULL,
  manager_id  NUMBER(5)    NOT NULL UNIQUE,
  adress      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
);
```

The screenshot shows a SQL script editor with the following text:

```
CREATE TABLE store_mng(
  store_id    NUMBER(5)    PRIMARY KEY,
  store_name  VARCHAR2(50) NOT NULL,
  manager_id  NUMBER(5)    NOT NULL UNIQUE,
  adress      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
);
```

Below the script, the 'Script Output' window displays the message: 'Table STORE\_MNG created.' and 'Task completed in 0,08 seconds'.

```
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.

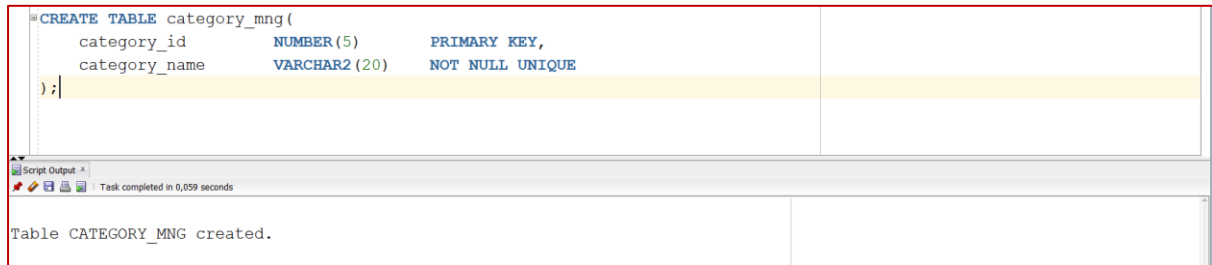
```
CREATE TABLE works_in(
    store_id    NUMBER(5)    REFERENCES store_mng(store_id) ON DELETE
    CASCADE,
    emp_id      NUMBER(5)    REFERENCES employee_mng(emp_id) ON DELETE
    CASCADE,
    CONSTRAINT pk_works_in PRIMARY KEY(store_id, emp_id)
);
```

```
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)
);
```

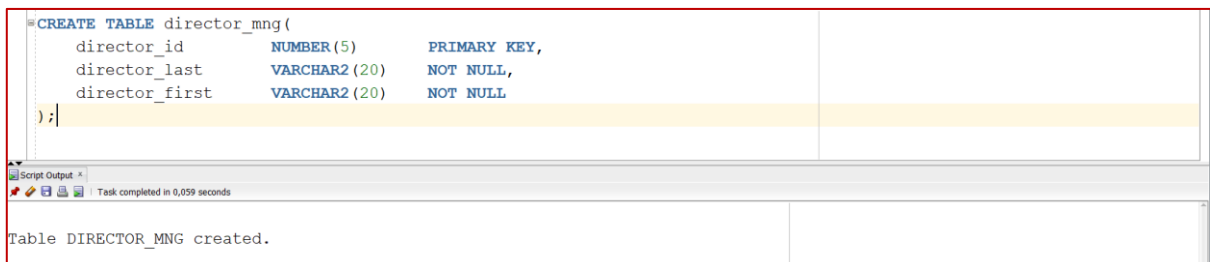
Script Output: Task completed in 0,048 seconds

Table WORKS\_IN created.

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



```
CREATE TABLE 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 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
);

```

Script Output

Task completed in 0,118 seconds

Table FILM\_MNG created.

```

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 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)
);

```

Script Output

Task completed in 0,062 seconds

Table STORE\_INVENTORY created.

```

CREATE TABLE actor_mng(

    actor_id      NUMBER(5)      PRIMARY KEY,

    actor_last    VARCHAR2(20)   NOT NULL,

    actor_first   VARCHAR2(20)   NOT NULL

);

```



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

Script Output: Task completed in 0,051 seconds

Table ACTOR\_MNG created.

```
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)
);

```

The screenshot shows a SQL script window with the following code:

```

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)
);

```

Below the script window, the 'Script Output' window displays the message: 'Table TRANSLATED\_TO created.' and 'Task completed in 0,055 seconds'.

```

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
);

```

The screenshot shows a SQL script window with the following code:

```

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
);

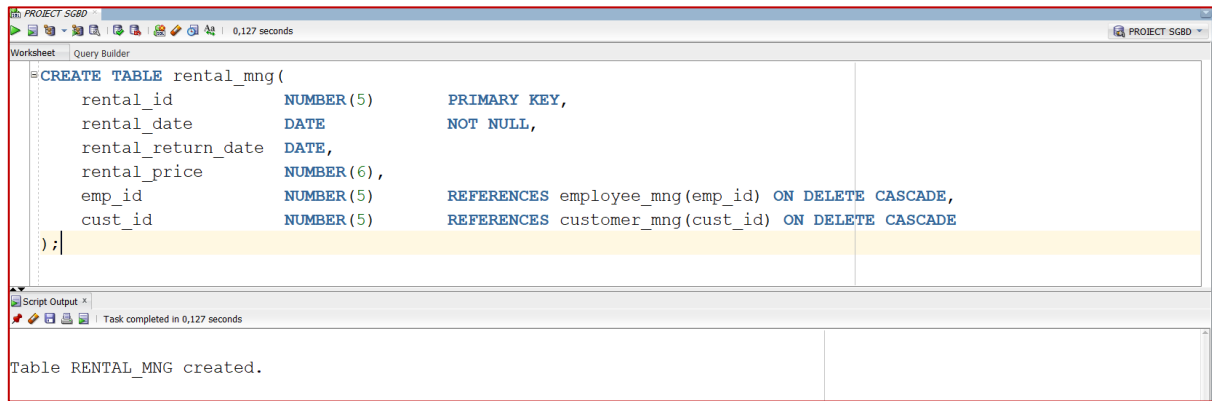
```

Below the script window, the 'Script Output' window displays the message: 'Table CUSTOMER\_MNG created.' and 'Task completed in 0,053 seconds'.

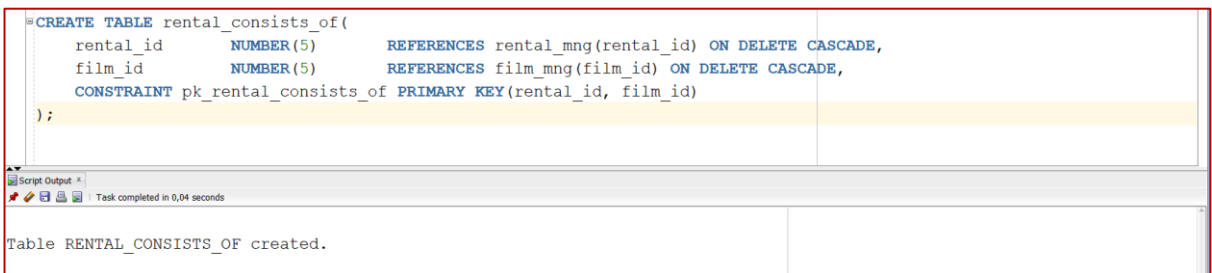
```

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
);
```

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL statement in the editor is:

```
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
);
```

The 'Script Output' tab at the bottom shows the message: 'Table ORDER\_MNG created.' The execution time is 0,115 seconds.

```
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)
);
```

The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL statement in the editor is:

```
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)
);
```

The 'Script Output' tab at the bottom shows the message: 'Table ORDER\_CONSISTS\_OF created.' The execution time is 0,086 seconds.

## 5. Adăugarea informațiilor coerente în tabele.

```
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
```

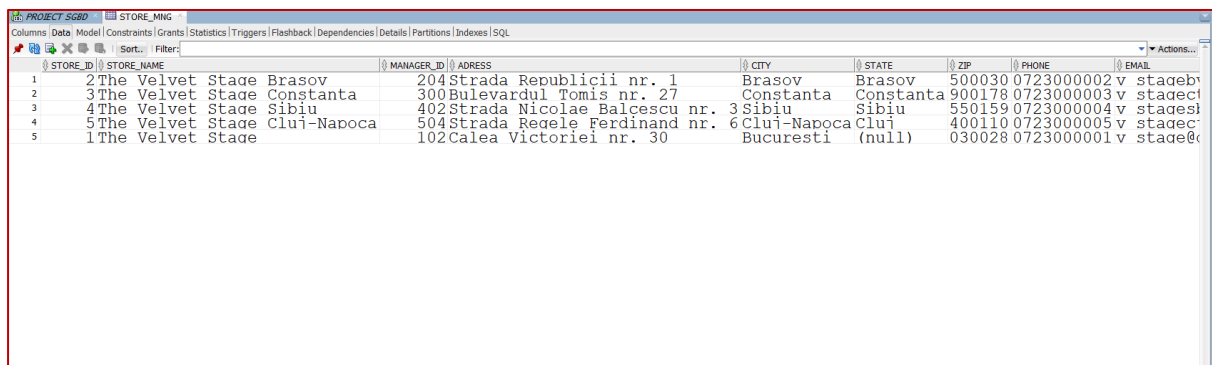
```
values (SEQUENCE_STORE_MNG.nextval, 'The Velvet Stage Constanta', 300, 'Bulevardul Tomis nr. 27', 'Constanta', 'Constanta', '900178', '0723000003', 'v_stagect@gmail.com');
```

```
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	The Velvet Stage		102 Calea Victoriei nr. 30	Bucuresti	(null)	030028	0723000001	v_stage@gmail.com
2	The Velvet Stage Brasov	204	Strada Republicii nr. 1	Brasov	Brasov	500030	0723000002	v_stagebv@gmail.com
3	The Velvet Stage Constanta	300	Bulevardul Tomis nr. 27	Constanta	Constanta	900178	0723000003	v_stagect@gmail.com
4	The Velvet Stage Sibiu	402	Strada Nicolae Balcescu nr. 3	Sibiu	Sibiu	550159	0723000004	v_stagesb@gmail.com
5	The Velvet Stage Cluj-Napoca	504	Strada Regele Ferdinand nr. 6	Cluj-Napoca	Cluj	400110	0723000005	v_stagecj@gmail.com

```
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);
```

EMP_ID	EMP_LAST	EMP_FIRST	EMP_PHONE	EMP_EMAIL	EMP_HIRE_DATE	EMP_SALARY
100	Iuliu	Carmen	0723100000	iuliu.carmen@gmail.com	06-10-2015	2500
101	Radu	Nicolae	0723101000	radu.nicolae@gmail.com	23-12-2013	2500
102	Sorin	Daciana	0723102000	sorin.daciana@gmail.com	31-03-2020	2100
103	Leonard	Floriana	0723103000	leonard.floriana@gmail.com	17-03-2020	2300
104	Valuca	Natalia	0723104000	valuca.natalia@gmail.com	26-10-2021	2100
105	Ciprian	Amalia	0723105000	ciprian.amalia@gmail.com	12-03-2019	2500
200	Rozalia	Mariana	0723200000	rozalia.mariana@gmail.com	18-08-2017	2100
201	Tereza	Cosmin	0723201000	tereza.cosmin@gmail.com	02-01-2018	2500
202	Ghenadie	Remus	0723202000	ghenadie.remus@gmail.com	24-06-2020	2500
203	Adam	Cristina	0723203000	adam.cristina@gmail.com	16-09-2021	2100
204	Marin	Tudor	0723204000	marin.tudor@gmail.com	13-07-2019	2500
205	Apostol	Alexandra	0723205000	apostol.alexandra@gmail.com	12-04-2017	2300
300	Dragomir	Ruxandra	0723300000	dragomir.ruxandra@gmail.com	08-05-2022	2100
301	Radulca	Miruna	0723301000	radulca.miruna@gmail.com	23-03-2018	2500
302	Filip	Cezar	0723302000	filip.cezar@gmail.com	17-10-2015	2500
303	Codrin	Elena	0723303000	codrin.elena@gmail.com	13-02-2017	2100
304	Isac	Emanuel	0723304000	isac.emmanuel@gmail.com	08-03-2016	2500
305	Mihail	Ivan	0723305000	mihail.ivan@gmail.com	11-07-2020	2500
400	Sebastian	Andrei	0723400000	sebastian.andrei@gmail.com	04-12-2021	2500
401	Filimon	Cosmina	0723401000	filimon.cosmina@gmail.com	02-01-2018	2500
402	Mihai	Catalina	0723402000	mihai.catalina@gmail.com	01-03-2022	2100
403	Horea	Stefan	0723403000	horea.stefan@gmail.com	20-09-2020	2100
404	Theodor	Felicia	0723404000	theodor.felicia@gmail.com	13-06-2016	3000
405	Petru	Vlad	0723405000	petru.vlad@gmail.com	14-10-2021	2100
500	Ilie	Adrian	0723500000	ilie.adrian@gmail.com	03-03-2020	2500
501	Radu	Claudia	0723501000	radu.claudia@gmail.com	18-04-2020	2500
502	Pompeiu	Alexandru	0723502000	pompeiu.alexandru@gmail.com	02-09-2021	3000
503	Soare	Eugen	0723503000	soare.eugen@gmail.com	07-03-2019	2500
504	Renata	Anca	0723504000	renata.anca@gmail.com	19-09-2019	2800
505	Neagu	Mara	0723505000	neagu.mara@gmail.com	05-02-2021	3000

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
19	4	400
20	4	401
21	4	402
22	4	403
23	4	404
24	4	405
25	5	500
26	5	501
27	5	502
28	5	503
29	5	504
30	5	505

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');

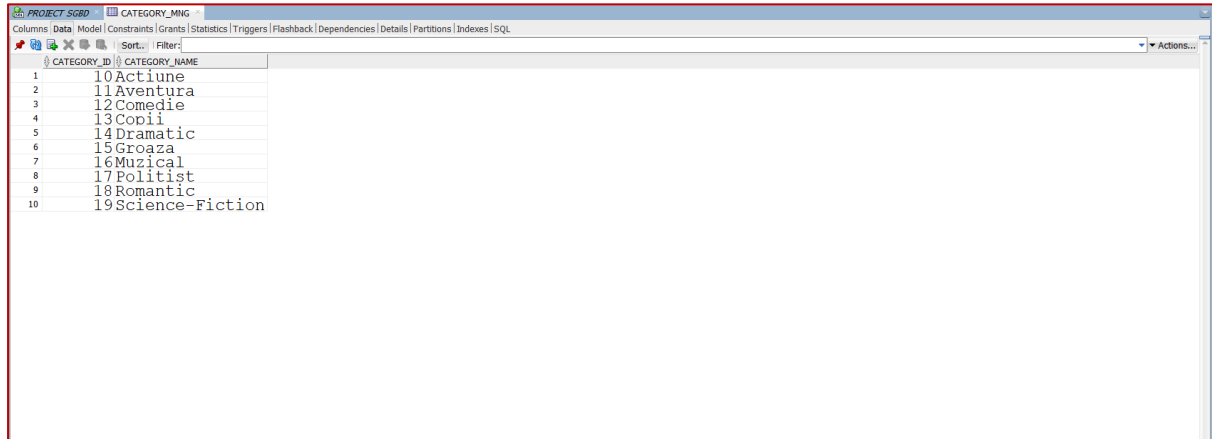
insert into CATEGORY\_MNG values (15,'Groaza');

insert into CATEGORY\_MNG values (16,'Muzical');

insert into CATEGORY\_MNG values (17,'Politist');

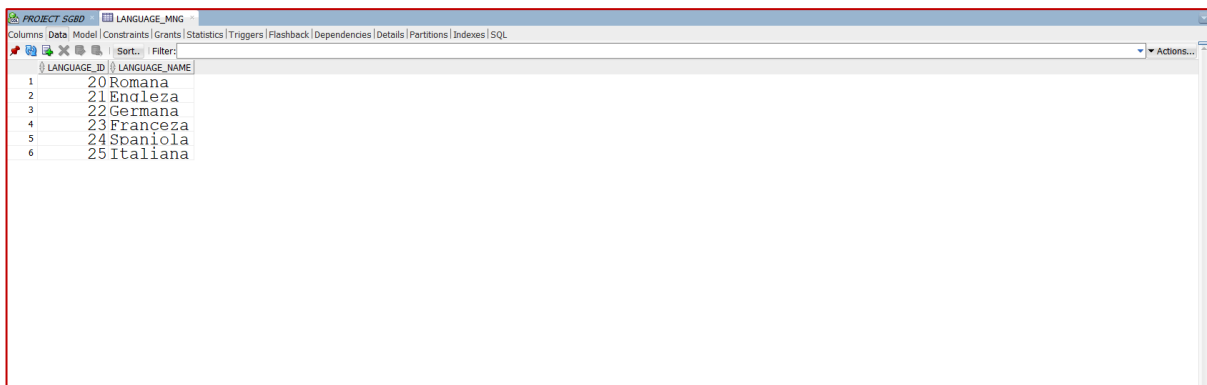
insert into CATEGORY\_MNG values (18,'Romantic');

insert into CATEGORY\_MNG values (19,'Science-Fiction');



CATEGORY_ID	CATEGORY_NAME
10	Actione
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');
```



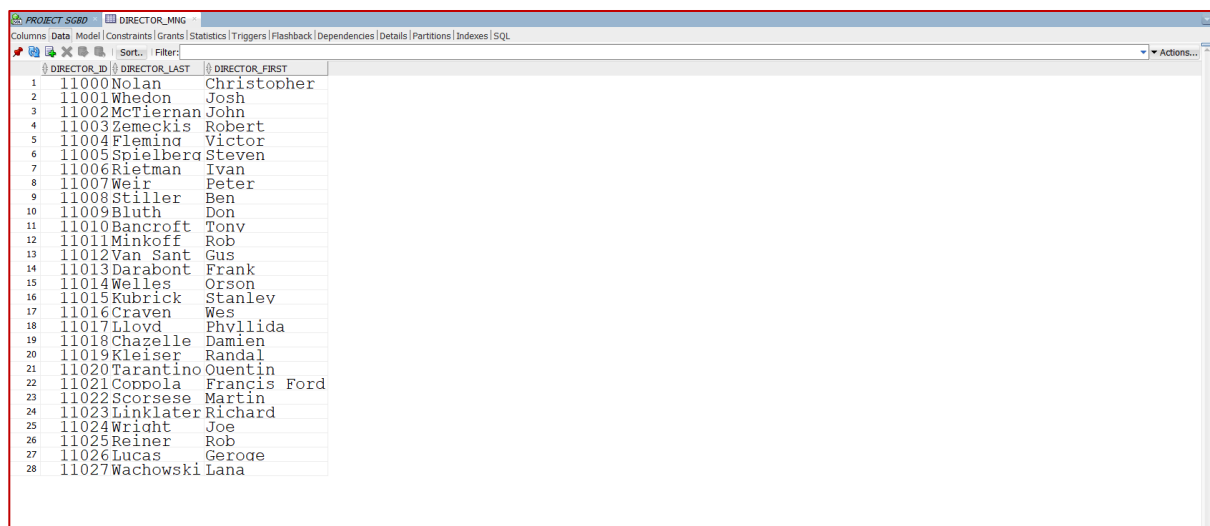
LANGUAGE_ID	LANGUAGE_NAME
20	Romana
21	Engleza
22	Germana
23	Franceza
24	Spaniola
25	Italiana

```
insert into DIRECTOR_MNG values (11000,'Nolan','Christopher');
insert into DIRECTOR_MNG values (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	Flemin 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
21	11020	Tarantino Quentin
22	11021	Coppola Francis Ford
23	11022	Scorsese Martin
24	11023	Linklater Richard
25	11024	Wright Joe
26	11025	Reiner Rob
27	11026	Lucas Geroge
28	11027	Wachowski Lana

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 tabelele 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 într-o lume incantatoare a palatelor elegante și a petrecerilor fastuoase. Dar cumplitul Rasputin, întors la Sankt Petersburg din dorința de a se răzbuna, va aprinde focul revoluției ce va cuprinde întreaga țară, în cele din urmă distrugând familia Romanov.

În urma atacului asupra Palatului Regal, singurele care vor supraviețui sunt împărăteasa Marie și nepoata ei, Anastasia. Urmăriți de Rasputin, singura lor speranță este să prindă ultimul tren...'

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

insert into FILM\_MNG

values (910,'Mulan',

'Într-o China plină de prejudecăți, o tânără pregătită pentru întâlnirea cu petitoarea se face de ras, neaducând onoare familiei. În același timp, hunii atacă teritoriul chinez, iar împăratul decretează mobilizarea recrutilor. Trista că nu și-a multumit tatăl, Mulan pleacă în locul acestuia la incorporare, deși pedeapsa pentru fetele care îndrăznesc să se încorporeze în armată este moartea. Mulan este ajutată în drumul ei de un dragon și de un greieras norocos, precum și de un cal falnic.'

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

insert into FILM\_MNG

values (911,'Regele leu',

'Acțiunea se petrece în locul fictiv Pride Lands din Africa, unde un leu este conducătorul tuturor animalelor ca rege. La începutul filmului, Rafiki, un mandril shaman, îl prezintă pe Simba, noul pui născut al regelui Mufasa și al reginei Sarabi, unei adunări a animalelor la Pride Rock. Între timp, fratele mai tânăr al lui Mufasa, Scar, realizează că nu mai este mostenitorul tronului și pune la cale să îl ucidă pe Simba, și 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 poseda 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 avand 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 primeste 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 aceeasi scoala 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, in regia lui Francis Ford Coppola. Este o adaptare dupa romanul lui Mario Puzo, "Nasul". Actiunea filmului se petrece in jurul anilor 1940-1950 si prezinta viata fictiva a familiei mafiote Corleone. Distributia filmului este impresionanta: Marlon Brando, Al Pacino, James Caan, Robert Duvall, Diane Keaton. Filmul a primit foarte multe premii, printre care si 3 Oscaruri si 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 semnata Martin Scorsese. Henry Hill (Ray Liotta) este un gangster de mana 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 sa devina personaje importante ale Mafiei, in timp ce Henry este afectat de succesul lor. Filmul este inspirat de un caz real si povestea lui este conturata in jurul lui Henry, mafiotul care isi vede prietenii urcand pe scara ierarhica a crimei.',

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

insert into FILM\_MNG

values (924,'Inainte de rasarit',

'O studenta frantuzoaica, aflata in ultimul an, Celine, interpretata de Julie Deply, intalneste un baiat american pe nume Jesse (Ethan Hawke) in trenul pe ruta Budapesta - Viena. In Viena coboara amandoi si ies impreuna. Idila lor de 14 ore este foarte impresionanta, doar ca la sfarsitul calatoriei, vor trebui sa se desparta. Sau... nu?',

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

insert into FILM\_MNG

values (925,'Mandrie si prejudecata',

'Povestea clasica de dragoste este plasata în Anglia sfarsitului secolului 18. Cele cinci surori Bennet - Elizabeth (Keira Knightley), Jane (Rosamund Pike), Lydia, Mary și Kitty – au fost crescute sub stricta ingrijire a mamei lor care se concentreaza doar asupra gasirii unor sotii fiicelor ei care sa le asigure din punct de vedere financiar. Insa, inteligenta si curajoasa Elizabeth se straduiește sa-si traiasca viata intr-o perspectiva mai larga, incurajata fiind de tatal 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 întâlnește 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 locuibila, 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
1	900Cavalerul intunecat	"Cavalerul intunecat" este un film de actiune scris si regizat de Chr
2	901Razbunatorii	Filmul este primul din franciza "Razbunatorii" si al saselea din Marv
3	902Greu de ucis	"Greu de ucis" este un film de actiune american din 1988 regizat de J
4	903Inapoi in viitor	Michael J. Fox joaca in rolul lui Marty McFly, adolescentul american
5	904Vrajitorul din Oz	Ca sa-si salveze catelusul Toto de o vecina rea, Dorothy vrea sa plec
6	905Indiana Jones si cautatorii arcei pierdute	Renumitul arheolog si expert in stiintele oculte, Dr. Indiana Jones (
7	906The Truman Show	"The Truman Show" este un film american SF de comedie din 1998 regiza
8	907Vanatorii de fantome	Dr. Peter Venkman (Bill Murray), Dr. Raymond Stantz (Dan Aykroyd), Dr
9	908Zoolander	Comedia regizata de Ben Stiller este o parodie la adresa superficiali
10	909Anastasia	Candva, nu foarte demult, familia Romanov traia intr-o lume incantato
11	910Mulan	Intr-o China plina de prejudecati, o tanara pregatita pentru intalnir
12	911Recede leu	Actiunea se petrece in locul fictiv Pride Lands din Africa, unde un l
13	912Good Will Hunting	"Good Will Hunting" este un film de drama american din 1997 regizat d
14	913Inchisoarea ingerilor	"Inchisoarea ingerilor" este un film american din anul 1994, scris si
15	914Cetateanul Kane	Un modul media moare singur in resedinta sa super-luxoasa. Jurnalisti
16	915Stralucirea	"Stralucirea" este un film horror psihologic din 1980, produs si reu
17	916Cosmar pe strada Elm	Nancv are un cosmar teribil legat de o figura inspaimantatoare care i
18	917Scream	Amestec de umor negru si mister. Un ucigas in serie hartuieste un gru
19	918Mamma Mia	Sophie Sheridan are 20 de ani si locuieste cu mama sa, pe o insula di
20	919La La Land	"La La Land" este un film american muzical de dragoste dramatic din 2
21	920Greas	"Greas" este un film muzical din 1978, stabilit in anii 1950, regiza
22	921Pulp Fiction	"Pulp Fiction" este un film american regizat de Quentin Tarantino, ca
23	922Nasul	"Nasul" este un film american din 1972, in regia lui Francis Ford Cop
24	923Baieti buni	O capodopera a genului semnata Martin Scorsese. Henry Hill (Ray Liott
25	924Inainte de rasarit	O studenta frantuzoaica, aflata in ultimul an, Celine, interpretata d
26	925Mandrie si prejudecata	Povestea clasica de dragoste este plasata in Anglia sfarsitului secol
27	926Cand Harry o intalneste pe Sally	Appreciatul regizor Rob Reiner contureaza aceasta comedie clasica roma
28	927Interstellar: Calatorind prin univers	"Interstellar: Calatorind prin univers" este un film stiintifico-fant
29	928Star Wars	Luke Skywalker (Mark Hamill) locuieste intr-o ferma de pe Planeta Tat
30	929Matrix	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);
```



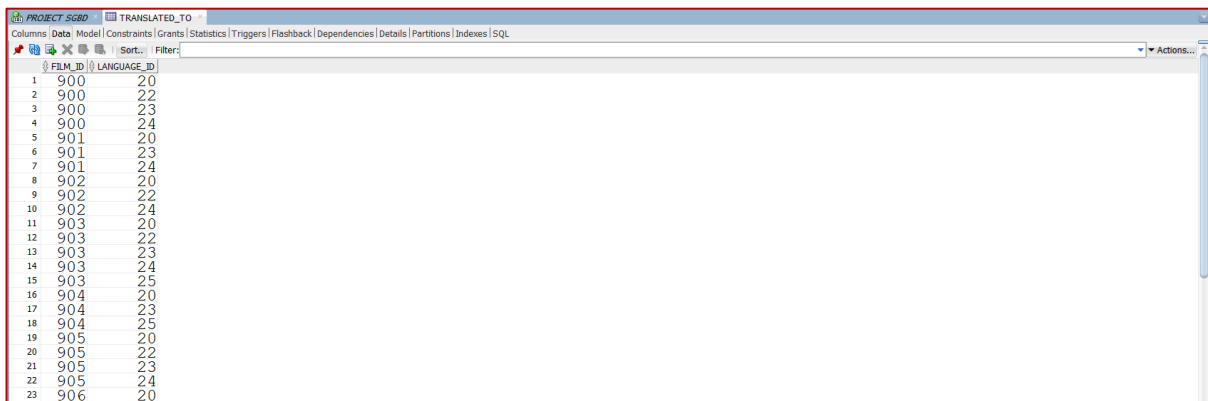
```
insert into TRANSLATED_TO values (909,20);
insert into TRANSLATED_TO values (909,22);
insert into TRANSLATED_TO values (909,23);
insert into TRANSLATED_TO values (910,20);
insert into TRANSLATED_TO values (910,23);
insert into TRANSLATED_TO values (910,25);
insert into TRANSLATED_TO values (911,20);
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

```

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','Timothee');
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 Garv
4	10003	Caine Michael
5	10004	Freeman Morgan
6	10005	Murphv Cilian
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 Jeremv
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	Llovd Christopher
22	10021	Thompson Lea
23	10022	Glover Crispin
24	10023	Wilson Thomas F.
25	10024	Garland Judy
26	10025	Lahr Bert
27	10026	Halev 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	Rhvs-Davies John
37	10036	Carrev Jim
38	10037	Harris Ed
39	10038	Linnev 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);
```

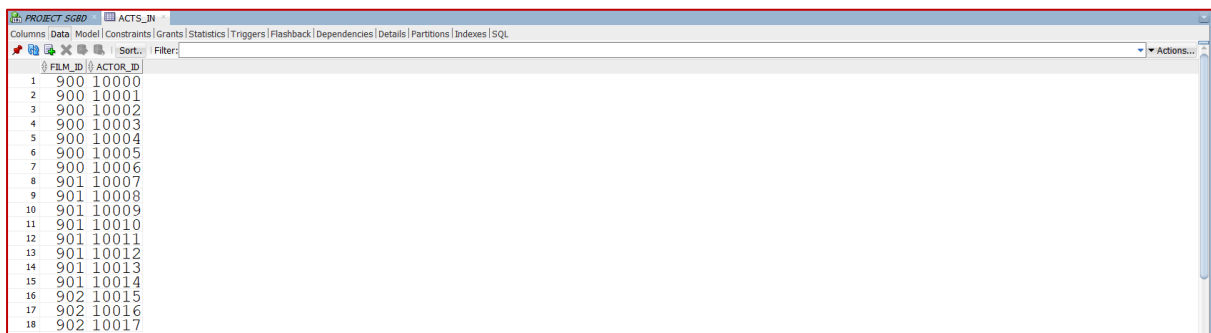
```
insert into ACTS_IN values (914,10073);
insert into ACTS_IN values (914,10074);
insert into ACTS_IN values (914,10075);
insert into ACTS_IN values (914,10076);
insert into ACTS_IN values (914,10077);
insert into ACTS_IN values (915,10078);
insert into ACTS_IN values (915,10079);
insert into ACTS_IN 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);

```



The screenshot shows a database management tool window titled 'PROJECT-5000' with the 'ACTS\_IN' table selected. The table has two columns: 'FILM\_ID' and 'ACTOR\_ID'. The data is displayed in a list view with 18 rows. The first 10 rows have 'FILM\_ID' 900 and 'ACTOR\_ID' from 10000 to 10009. The next 5 rows have 'FILM\_ID' 901 and 'ACTOR\_ID' from 10010 to 10014. The last 3 rows have 'FILM\_ID' 902 and 'ACTOR\_ID' from 10015 to 10017.

	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

```

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);

```

	CUST_ID	CUST_LAST	CUST_FIRST	CUST_PHONE	CUST_EMAIL
1	1000	Petre	Veronica	0732100000	(null)
2	1001	Emil	Alexandru	0732100100	emil.alexandru@gmail.com
3	1002	Anton	Iacob	(null)	(null)
4	1003	Antoni	Isabela	0732100300	antoni.isabela@gmail.com
5	1004	Toma	Alin	(null)	toma.alin@gmail.com
6	1005	Eduard	Carmen	0732100500	(null)
7	1006	Augustin	Iuliana	0732100600	augustin.iuliana@gmail.com
8	1007	Simion	Constantin	(null)	(null)
9	1008	Emanuel	Dragos	0732100800	emanuel.dragos@gmail.com
10	1009	Gheorghe	Liviu	0732100900	(null)
11	1010	Leonard	Emil	0732101000	leonard.emil@gmail.com
12	1011	Florian	Cezar	0732101100	(null)
13	1012	Dinu	Valentin	(null)	(null)
14	1013	Iosif	Stefan	0732101300	iosif.stefan@gmail.com
15	1014	Iancu	Serdhei	(null)	iancu.serdhei@gmail.com
16	1015	Ioan	Paul	0732101500	(null)
17	1016	Adrian	Petru	0732101600	adrian.petru@gmail.com
18	1017	Marius	Albert	(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

```

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);

```

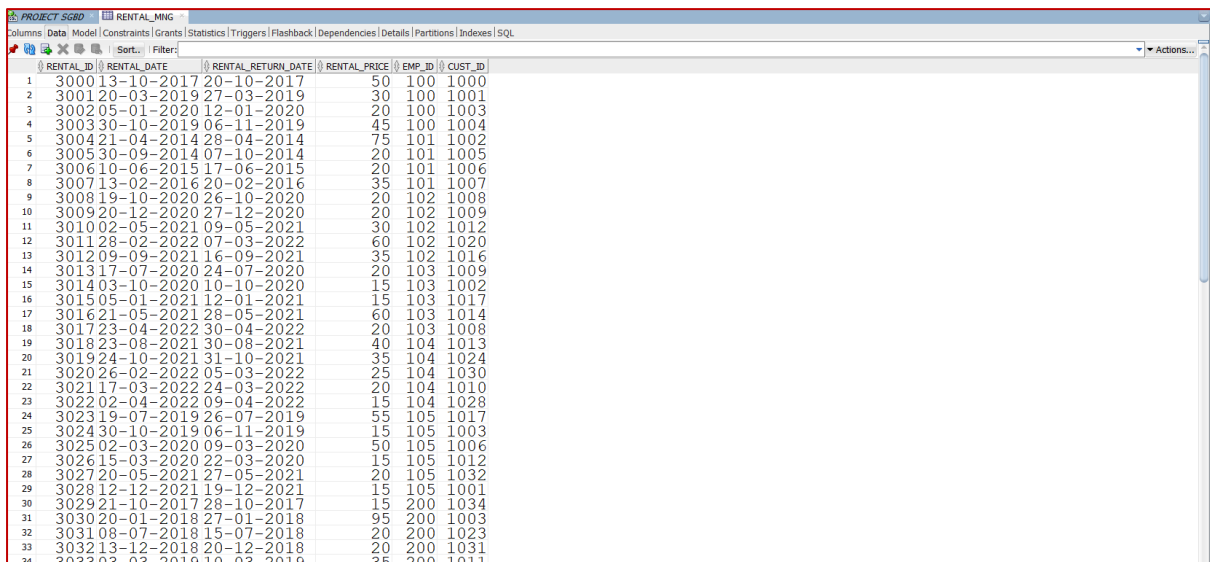


```

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);



	RENTAL_ID	RENTAL_DATE	RENTAL_RETURN_DATE	RENTAL_PRICE	EMP_ID	CUST_ID
1	3000	13-10-2017	20-10-2017	50	100	1000
2	3001	20-03-2019	27-03-2019	30	100	1001
3	3002	05-01-2020	12-01-2020	20	100	1003
4	3003	30-10-2019	06-11-2019	45	100	1004
5	3004	21-04-2014	28-04-2014	75	101	1002
6	3005	30-09-2014	07-10-2014	20	101	1005
7	3006	10-06-2015	17-06-2015	20	101	1006
8	3007	13-02-2016	20-02-2016	35	101	1007
9	3008	19-10-2020	26-10-2020	20	102	1008
10	3009	20-12-2020	27-12-2020	20	102	1009
11	3010	02-05-2021	09-05-2021	30	102	1012
12	3011	28-02-2022	07-03-2022	60	102	1020
13	3012	09-09-2021	16-09-2021	35	102	1016
14	3013	17-07-2020	24-07-2020	20	103	1009
15	3014	03-10-2020	10-10-2020	15	103	1002
16	3015	05-01-2021	12-01-2021	15	103	1017
17	3016	21-05-2021	28-05-2021	60	103	1014
18	3017	23-04-2022	30-04-2022	20	103	1008
19	3018	23-08-2021	30-08-2021	40	104	1013
20	3019	24-10-2021	31-10-2021	35	104	1024
21	3020	26-02-2022	05-03-2022	25	104	1030
22	3021	17-03-2022	24-03-2022	20	104	1010
23	3022	02-04-2022	09-04-2022	15	104	1028
24	3023	19-07-2019	26-07-2019	55	105	1017
25	3024	30-10-2019	06-11-2019	15	105	1003
26	3025	02-03-2020	09-03-2020	50	105	1006
27	3026	15-03-2020	22-03-2020	15	105	1012
28	3027	20-05-2021	27-05-2021	20	105	1032
29	3028	12-12-2021	19-12-2021	15	105	1001
30	3029	21-10-2017	28-10-2017	15	200	1034
31	3030	20-01-2018	27-01-2018	95	200	1003
32	3031	08-07-2018	15-07-2018	20	200	1023
33	3032	13-12-2018	20-12-2018	20	200	1031
34	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);

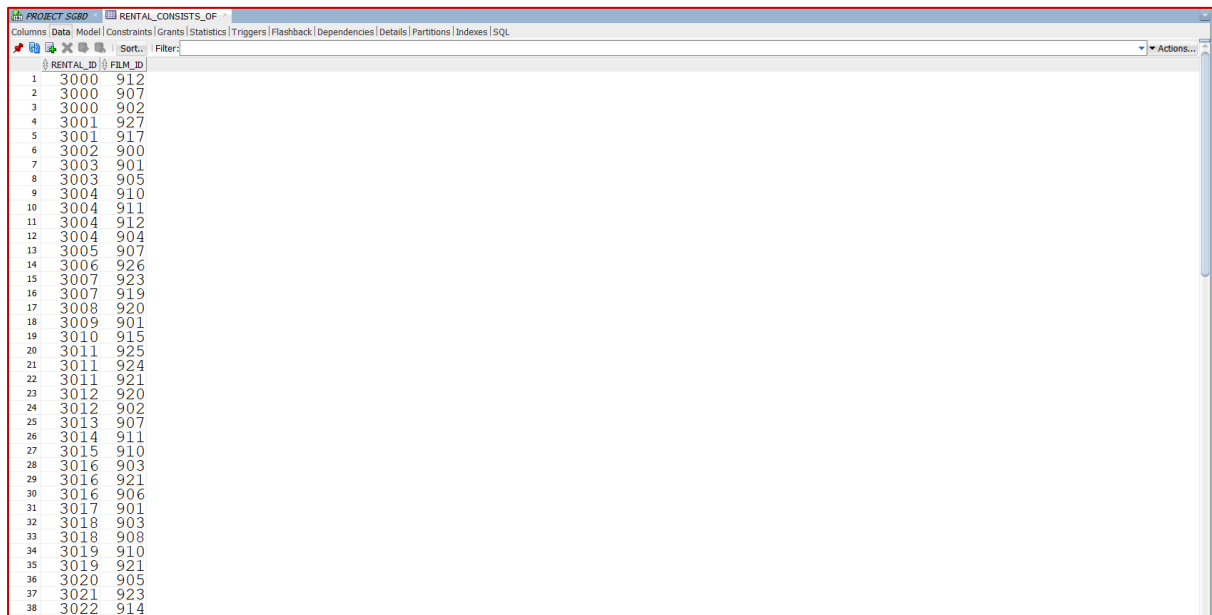
insert into RENTAL\_CONSISTS\_OF values (3042,923);  
insert into RENTAL\_CONSISTS\_OF values (3042,907);  
insert into RENTAL\_CONSISTS\_OF values (3042,911);  
insert into RENTAL\_CONSISTS\_OF values (3043,919);  
insert into RENTAL\_CONSISTS\_OF values (3044,902);  
insert into RENTAL\_CONSISTS\_OF 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

## 6. Subprogram stocat independent care utilizează două tipuri diferite de colecții.

Ceriță: Să se afișeze lista actorilor care joacă în cel puțin un film de comedie.

```
CREATE OR REPLACE PROCEDURE p_6_mng IS
    TYPE vector IS VARRAY(20) OF film_mng.film_id%TYPE;
    TYPE tablou_indexat IS TABLE OF actor_mng.actor_id%TYPE INDEX BY
PLS_INTEGER;

    f_id      vector;
    a_id      tablou_indexat;
    a_nume    actor_mng.actor_last%TYPE;
    a_prenume actor_mng.actor_first%TYPE;
BEGIN
    SELECT film_id BULK COLLECT INTO f_id
    FROM film_mng f
    JOIN category_mng c ON (f.category_id = c.category_id)
    WHERE UPPER(category_name) = 'COMEDIE';

    FOR i IN 1..f_id.COUNT LOOP
        SELECT actor_id BULK COLLECT INTO a_id
        FROM acts_in
        WHERE film_id = f_id(i);

        FOR i IN 1..a_id.COUNT LOOP
            SELECT actor_last INTO a_nume
            FROM actor_mng
            WHERE actor_id = a_id(i);

            SELECT actor_first INTO a_prenume
            FROM actor_mng
            WHERE actor_id = a_id(i);
```

```

        DBMS_OUTPUT.PUT_LINE('Nume: ' || a_nume || '; Prenume: ' ||
a_prenume);

    END LOOP;

END LOOP;

END p_6_mng;

```

```

Worksheet  Query Builder
CREATE OR REPLACE PROCEDURE p_6_mng IS
    TYPE vector IS VARRAY(20) OF film_mng.film_id%TYPE;
    TYPE tablou_indexat IS TABLE OF actor_mng.actor_id%TYPE INDEX BY PLS_INTEGER;

    f_id      vector;
    a_id      tablou_indexat;

    a_nume    actor_mng.actor_last%TYPE;
    a_prenume actor_mng.actor_first%TYPE;

BEGIN
    SELECT film_id BULK COLLECT INTO f_id
    FROM film_mng f
    JOIN category_mng c ON (f.category_id = c.category_id)
    WHERE UPPER(category_name) = 'COMEDIE';

    FOR i IN 1..f_id.COUNT LOOP
        SELECT actor_id BULK COLLECT INTO a_id

```

Script Output - Task completed in 0,18 seconds

Procedure P\_6\_MNG compiled

```

Apel:

BEGIN

    p_6_mng;

END;

/

```

```

PROJECT SGBD  ACTOR_MNG
Worksheet  Query Builder
BEGIN
    p_6_mng;
END;

```

Script Output - Task completed in 0,068 seconds

PL/SQL procedure successfully completed.

Dbms Output - Buffer Size:20000

PROJECT SGBD

Nume: Carrey; Prenume: Jim  
Nume: Harris; Prenume: Ed  
Nume: Linney; Prenume: Laura  
Nume: McElhone; Prenume: Natascha  
Nume: Murray; Prenume: Bill  
Nume: Aykroyd; Prenume: Dan  
Nume: Ramis; Prenume: Harold  
Nume: Weaver; Prenume: Sigourney  
Nume: Hudson; Prenume: Ernie  
Nume: Stiller; Prenume: Ben  
Nume: Wilson; Prenume: Owen  
Nume: Ferrell; Prenume: Will  
Nume: Taylor; Prenume: Christine  
Nume: Jovovitch; Prenume: Milia  
Nume: Vaughn; Prenume: Vince

## 7. Subprogram stocat independent care utilizează două tipuri diferite de cursoare.

Cerință: Pentru fiecare magazin din franciză, să se afișeze lista angajaților care au porcesat cel puțin o comandă în 2022.

```
CREATE OR REPLACE PROCEDURE p_7_mng IS
    CURSOR c_e(s store_mng.store_id%TYPE) IS
        SELECT DISTINCT emp_last nume, emp_first prenume
        FROM employee_mng e
        JOIN works_in w_i ON (w_i.store_id = s AND w_i.emp_id = e.emp_id)
        JOIN order_mng o ON (e.emp_id = o.emp_id)
        WHERE EXTRACT (YEAR FROM o.ord_date) = 2022;

    e_numar    NUMBER(4);
BEGIN
    FOR s IN (SELECT store_id id, store_name nume
              FROM store_mng) LOOP
        DBMS_OUTPUT.PUT_LINE('- ' || s.nume || ' -');
        e_numar := 0;

        DBMS_OUTPUT.PUT_LINE('Lista angajati: ');
        FOR e IN c_e(s.id) LOOP
            EXIT WHEN c_e%NOTFOUND;

            DBMS_OUTPUT.PUT_LINE('Angajat: ' || e.nume || ' ' || e.prenume) ;
            e_numar := e_numar + 1;
        END LOOP;
        IF e_numar = 0 THEN
            DBMS_OUTPUT.PUT_LINE('Niciun angajat nu a procesat o
comanda in 2022!');
        END IF;
        DBMS_OUTPUT.NEW_LINE;
```



```

END LOOP;

END p_7_mng;

```

```

CREATE OR REPLACE PROCEDURE p_7_mng IS
  CURSOR c_e(s store_mng.store_id%TYPE) IS
    SELECT DISTINCT emp_last nume, emp_first prenume
    FROM employee_mng e
    JOIN works_in w_i ON (w_i.store_id = s AND w_i.emp_id = e.emp_id)
    JOIN order_mng o ON (e.emp_id = o.emp_id)
    WHERE EXTRACT (YEAR FROM o.ord_date) = 2022;

  e_numar      NUMBER(4);
BEGIN
  FOR s IN (SELECT store_id id, store_name nume
            FROM store_mng) LOOP
    DBMS_OUTPUT.PUT_LINE('- ' || s.nume || ' -');

    e_numar := 0;

    DBMS_OUTPUT.PUT_LINE('Lista angajati: ');
    FOR e IN c_e(s.id) LOOP

```

Apel:

BEGIN

p\_7\_mng;

END;

/

```

BEGIN
  p_7_mng;
END;

```

PL/SQL procedure successfully completed.

DBMS Output:

```

- The Velvet Stage Brasov -
Lista angajati:
Angajat: Ghenadie Remus
Angajat: Adam Cristina

- The Velvet Stage Constanta -
Lista angajati:
Angajat: Dragomir Ruxandra

- The Velvet Stage Sibiu -
Lista angajati:
Angajat: Sebastian Andrei
Angajat: Petru Vlad
Angajat: Mihai Catalina

- The Velvet Stage Cluj-Napoca -
Lista angajati:
Niciun angajat nu a procesat o comanda in 2022!

- The Velvet Stage -
Lista angajati:
Angajat: Raluca Natalia

```

## 8. Subprogram stocat independent de tip funcție, utilizând 3 tabele definite într-o singură comandă SQL.

Cerință: Pentru un magazin din franciză, să se determine numărul de regizori, care au regizat cel puțin un film, având un rating mai mare sau egal cu un rating dat.

```
CREATE OR REPLACE FUNCTION f_8_mng(s_nume store_mng.store_name%TYPE,
                                   rating film_mng.film_rating%TYPE)

RETURN NUMBER IS

    s_nume_invalid    EXCEPTION;
    rating_invalid    EXCEPTION;

    s_id              store_mng.store_id%TYPE;
    numarator         NUMBER(4) := 0;
BEGIN
    IF UPPER(s_nume) NOT LIKE 'THE VELVET STAGE%' THEN
        RAISE s_nume_invalid;
    END IF;

    IF rating < 1 OR rating > 10 THEN
        RAISE rating_invalid;
    END IF;

    SELECT store_id INTO s_id
    FROM store_mng
    WHERE store_name = s_nume;

    SELECT COUNT(*) INTO numarator
    FROM store_mng s
    JOIN store_inventory s_i ON (s.store_id = s_i.store_id)
    JOIN film_mng f ON (s_i.film_id = f.film_id)
    JOIN director_mng d ON (f.director_id = d.director_id)
```

WHERE s.store\_id = s\_id AND f.film\_rating > rating;

RETURN numarator;

EXCEPTION

WHEN s\_nume\_invalid THEN

RAISE\_APPLICATION\_ERROR(-20000, 'Numele magazinului  
introdus este invalid!');

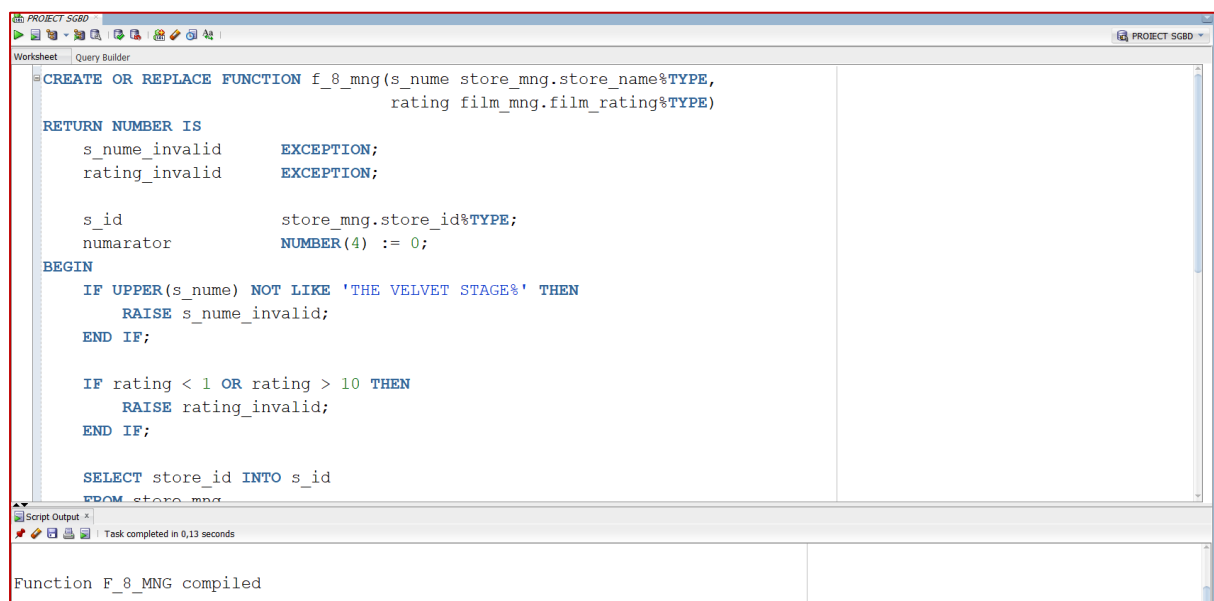
WHEN rating\_invalid THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Rating-ul introdus este  
invalid!');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Alta eroare!');

END f\_8\_mng;



Apel:

DECLARE

numarator                   NUMBER(4);

BEGIN

numarator := f\_8\_mng('The Velvet Stage',8);

DBMS\_OUTPUT.PUT\_LINE(numarator);

END;

/

```

DECLARE
    numarator      NUMBER(4);
BEGIN
    numarator := f_8_mng('The Velvet Stage',8);
    DBMS_OUTPUT.PUT_LINE(numarator);
END;
/

```

18

Task completed in 0,062 seconds

PL/SQL procedure successfully completed.

Eroare – s\_nume\_invalid:

DECLARE

numarator                   NUMBER(4);

BEGIN

numarator := f\_8\_mng('The Velvet',8);

DBMS\_OUTPUT.PUT\_LINE(numarator);

END;

/

```

DECLARE
    numarator      NUMBER(4);
BEGIN
    numarator := f_8_mng('The Velvet',8);
    DBMS_OUTPUT.PUT_LINE(numarator);
END;
/

```

Error starting at line : 1 in command -  
 DECLARE  
 numarator NUMBER(4);  
 BEGIN  
 numarator := f\_8\_mng('The Velvet',8);  
 DBMS\_OUTPUT.PUT\_LINE(numarator);  
 END;  
 Error report -  
 ORA-20000: Numele magazinului introdus este invalid!  
 ORA-06512: la "GRUPA231.F\_8\_MNG", linia 33

Eroare – rating\_invalid:

DECLARE

numarator                   NUMBER(4);

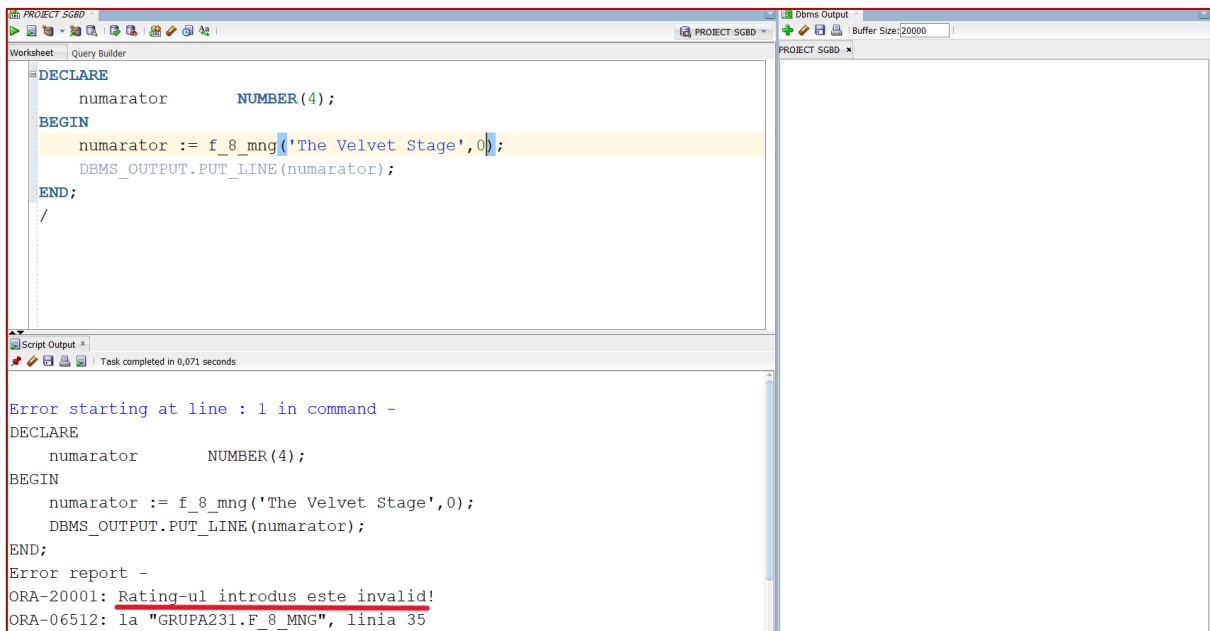
BEGIN

numarator := f\_8\_mng('The Velvet Stage',0);

DBMS\_OUTPUT.PUT\_LINE(numarator);

END;

/



Eroare – altă eroare:

DECLARE

numarator                   NUMBER(4);

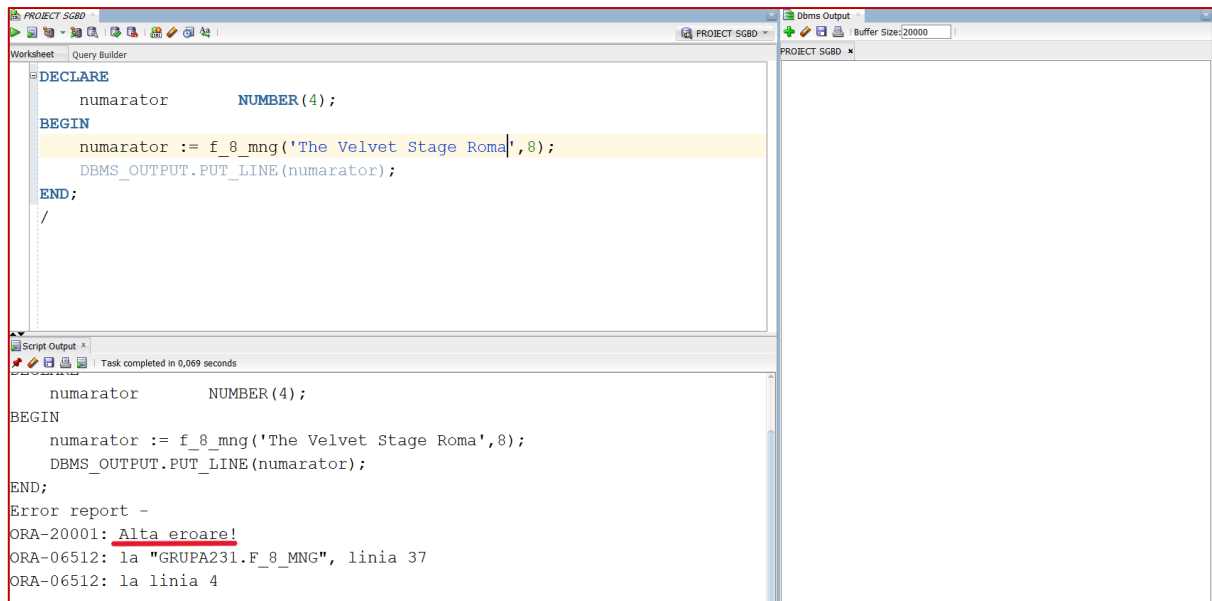
BEGIN

numarator := f\_8\_mng('The Velvet Stage Roma',8);

DBMS\_OUTPUT.PUT\_LINE(numarator);

END;

/



The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Worksheet' tab. The script is as follows:

```
DECLARE
    numarator      NUMBER(4);
BEGIN
    numarator := f_8_mng('The Velvet Stage Roma',8);
    DBMS_OUTPUT.PUT_LINE(numarator);
END;
/
```

The 'Script Output' tab at the bottom shows the execution results. It indicates that the task was completed in 0.069 seconds. Below this, an 'Error report -' section lists three Oracle errors:

```
ORA-20001: Alta eroare!
ORA-06512: la "GRUPA231.F_8_MNG", linia 37
ORA-06512: la linia 4
```

## 9. Subprogram stocat independent de tip procedură, utilizând 5 tabele definite într-o singură comandă SQL.

Cerință: Pentru un client și o limbă, afișați lista filmelor pe care clientul le-a închiriat, care au fost traduse în limba respectivă.

```
CREATE OR REPLACE PROCEDURE p_9_mng (nume customer_mng.cust_last%TYPE,
    prenume customer_mng.cust_first%TYPE,
    limba language_mng.language_name%TYPE) IS
    c_id      customer_mng.cust_id%TYPE;
    l_id      language_mng.language_id%TYPE;

    TYPE vector IS VARRAY(50) OF film_mng.film_title%TYPE;
    f_titlu   vector;
BEGIN
    SELECT cust_id INTO c_id
    FROM customer_mng
    WHERE UPPER(cust_last) = UPPER(nume) AND UPPER(cust_first) =
UPPER(prenume);

    SELECT language_id INTO l_id
    FROM language_mng
    WHERE UPPER(language_name) = UPPER(limba);

    SELECT f.film_title BULK COLLECT INTO f_titlu
    FROM customer_mng c
    JOIN rental_mng r ON (c.cust_id = r.cust_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 translated_to t ON (f.film_id = t.film_id)
    WHERE c.cust_id = c_id AND t.language_id = l_id;
```

```

IF f_titlu.COUNT = 0 THEN

    DBMS_OUTPUT.PUT_LINE('Clientul introdus nu a rezervat niciun film
tradus in ' || LOWER(limba));

ELSE

    DBMS_OUTPUT.PUT_LINE('- Lista filme - ');
    FOR i IN 1..f_titlu.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE('Film: ' || f_titlu(i));
    END LOOP;
END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'Datele introduse nu
exista!');

    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Exista mai multi cu clienti
cu acest nume!');

    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20003, 'Alta eroare!');

END p_9_mng;

```

```

CREATE OR REPLACE PROCEDURE p_9_mng (nume customer_mng.cust_last%TYPE,
                                     prenume customer_mng.cust_first%TYPE,
                                     limba language_mng.language_name%TYPE) IS
    c_id          customer_mng.cust_id%TYPE;
    l_id          language_mng.language_id%TYPE;

    TYPE vector IS VARRAY(50) OF film_mng.film_title%TYPE;
    f_titlu       vector;
BEGIN
    SELECT cust_id INTO c_id
    FROM customer_mng
    WHERE UPPER(cust_last) = UPPER(nume) AND UPPER(cust_first) = UPPER(prenume);

    SELECT language_id INTO l_id
    FROM language_mng
    WHERE UPPER(language_name) = UPPER(limba);

    SELECT f.film_title BULK COLLECT INTO f_titlu

```

Script Output \*  
Task completed in 0.159 seconds

Procedure P\_9\_MNG compiled



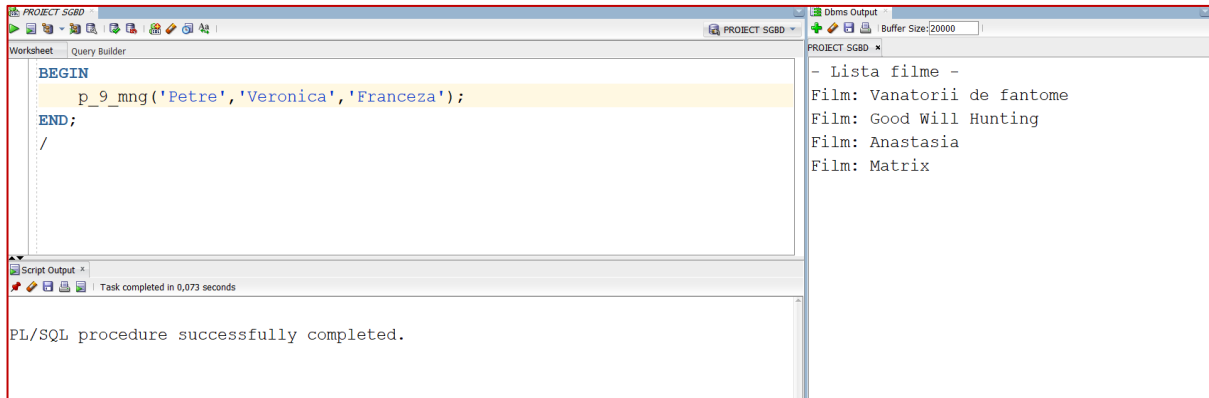
Apel:

BEGIN

p\_9\_mng('Petre','Veronica','Franceza');

END;

/



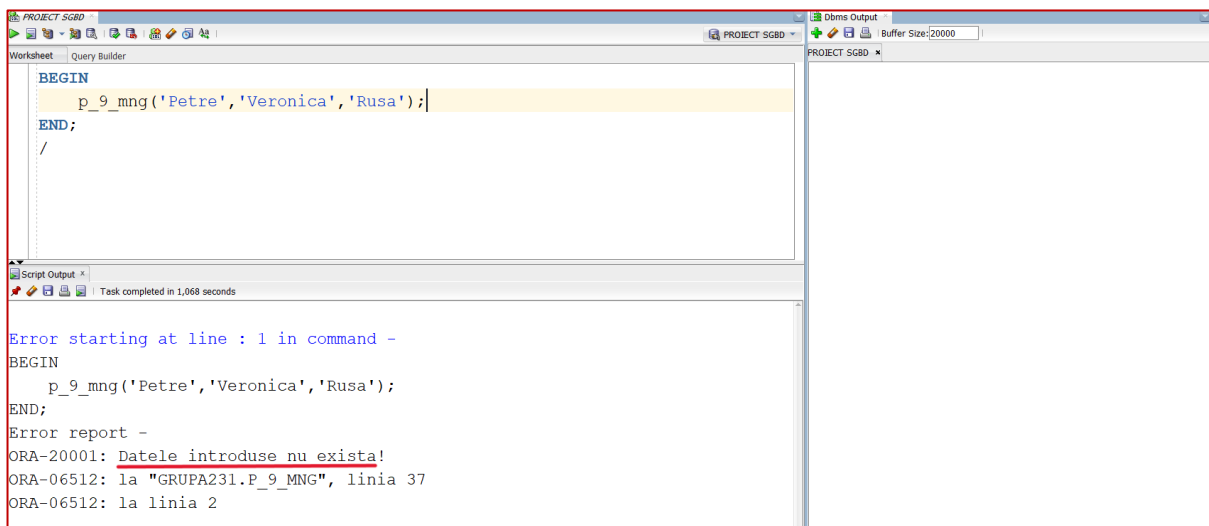
Eroare NO\_DATA\_FOUND:

BEGIN

p\_9\_mng('Petre','Veronica','Rusa');

END;

/



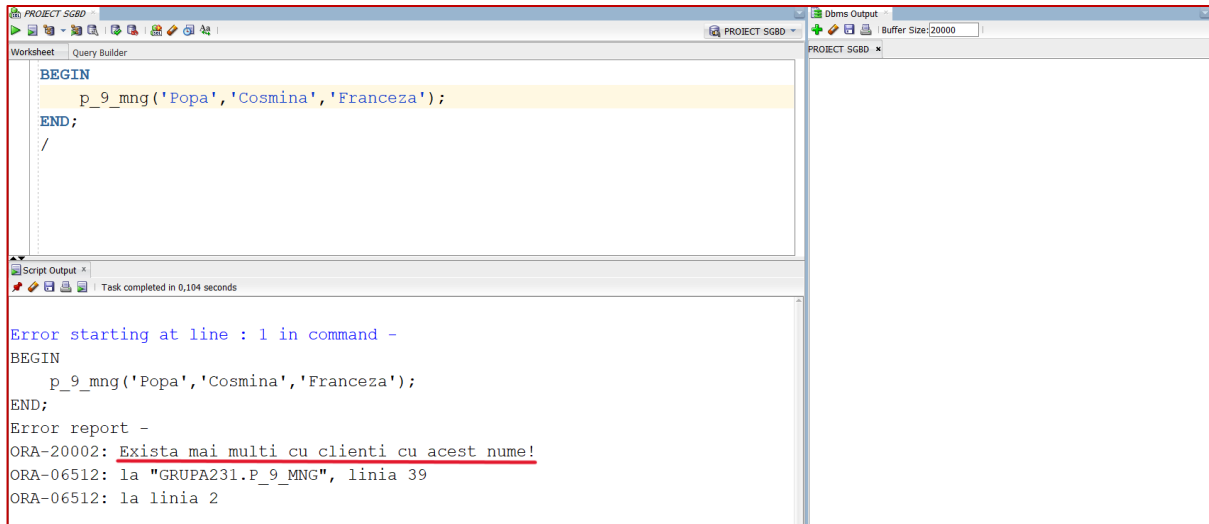
Eroare TOO\_MANY\_ROWS:

BEGIN

    p\_9\_mng('Popa','Cosmina','Franceza');

END;

/

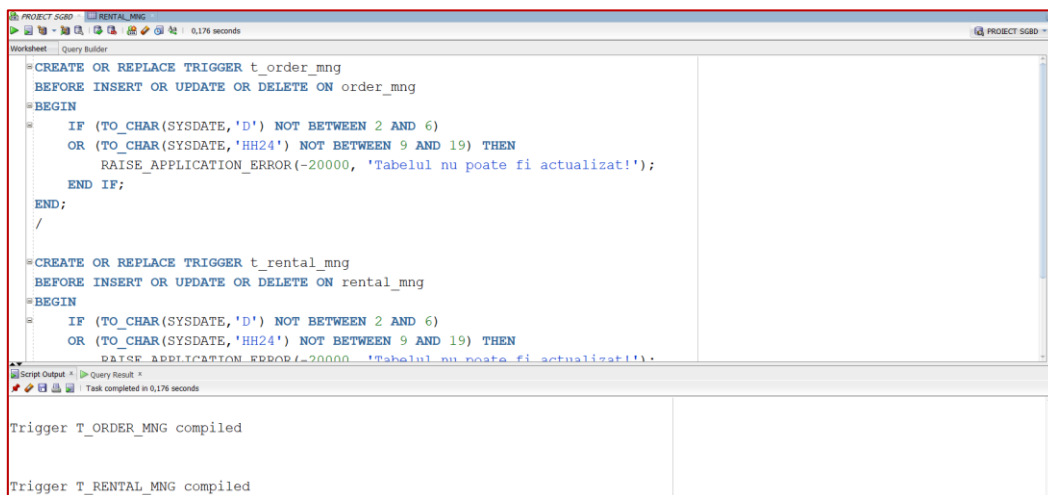


## 10. Trigger de tip LMD la nivel de comandă.

Cerință: Declanșator de tip LMD la nivel de comandă care nu permite inserarea unei noi comenzi sau înregistrări, decât în intervalul orar 09:00 – 19:00, de luni până vineri.

```
CREATE OR REPLACE TRIGGER t_10_order_mng
BEFORE INSERT OR UPDATE OR DELETE ON order_mng
BEGIN
    IF (TO_CHAR(SYSDATE,'D') NOT BETWEEN 2 AND 6)
    OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 10 AND 19) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Tabelul nu poate fi actualizat!');
    END IF;
END;
/

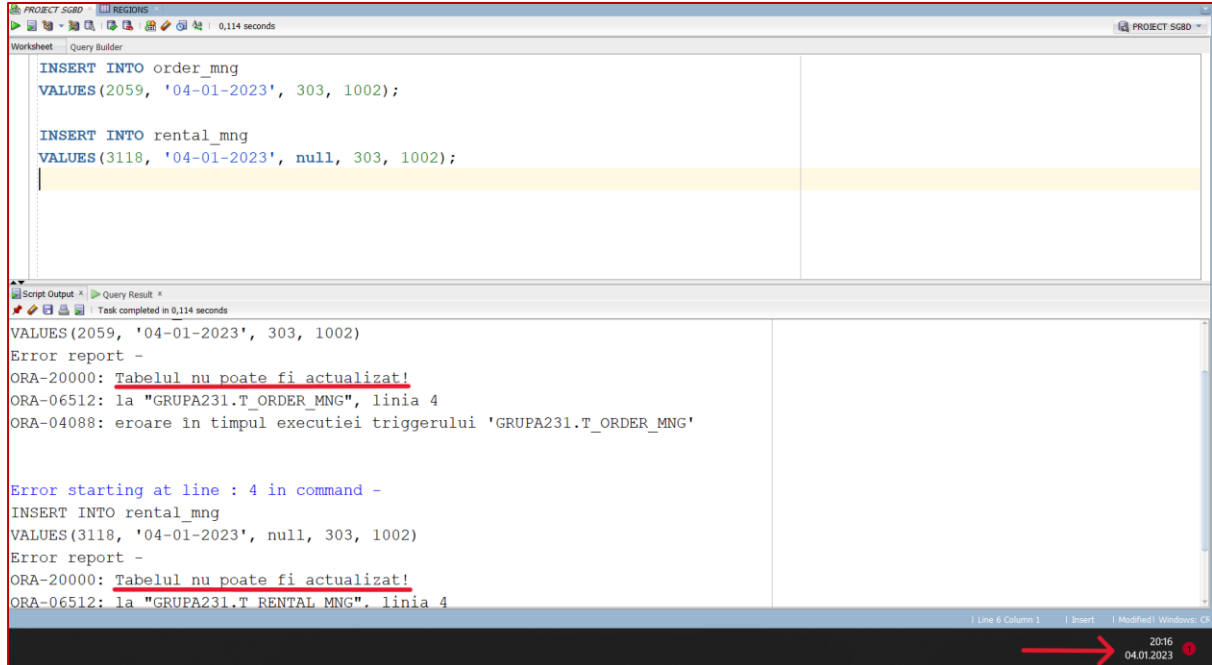
CREATE OR REPLACE TRIGGER t_10_rental_mng
BEFORE INSERT OR UPDATE OR DELETE ON rental_mng
BEGIN
    IF (TO_CHAR(SYSDATE,'D') NOT BETWEEN 2 AND 6)
    OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 10 AND 19) THEN
        RAISE_APPLICATION_ERROR(-20000, 'Tabelul nu poate fi actualizat!');
    END IF;
END;
/
```



Declanșare:

```
INSERT INTO order_mng VALUES(2059, '04-01-2023', 303, 1002);
```

```
INSERT INTO rental_mng VALUES(3118, '04-01-2023', null, 303, 1002);
```



## 11. Trigger de tip LMD la nivel de linie.

Cerință: Declanșator de tip LMD care va actualiza prețul unei comezi/închirieri.

```
CREATE OR REPLACE TRIGGER t_11_order_mng
AFTER INSERT OR UPDATE OR DELETE ON order_consists_of
FOR EACH ROW
DECLARE
    f_pret      film_mng.film_order_price%TYPE;
    f_pret_vechi film_mng.film_order_price%TYPE;
BEGIN
    IF INSERTING THEN
        SELECT film_order_price INTO f_pret
        FROM film_mng
        WHERE film_id = :NEW.film_id;

        UPDATE order_mng
        SET ord_price = NVL(ord_price, 0) + (f_pret * :NEW.number_of_films)
        WHERE ord_id = :NEW.ord_id;
    ELSIF UPDATING('film_id') THEN
        SELECT film_order_price INTO f_pret_vechi
        FROM film_mng
        WHERE film_id = :OLD.film_id;

        SELECT film_order_price INTO f_pret
        FROM film_mng
        WHERE film_id = :NEW.film_id;

        UPDATE order_mng
        SET ord_price = ord_price - (f_pret_vechi * :OLD.number_of_films) +
        (f_pret * :NEW.number_of_films)
        WHERE ord_id = :OLD.ord_id;
```

```

ELSIF UPDATING('number_of_films') THEN
    SELECT film_order_price INTO f_pret
    FROM film_mng
    WHERE film_id = :OLD.film_id;

    UPDATE order_mng
    SET ord_price = ord_price - (f_pret * :OLD.number_of_films) + (f_pret
:NEW.number_of_films)
    WHERE ord_id = :OLD.ord_id;
ELSE
    SELECT film_order_price INTO f_pret_vechi
    FROM film_mng
    WHERE film_id = :OLD.film_id;

    UPDATE order_mng
    SET ord_price = ord_price - (f_pret_vechi * :OLD.number_of_films)
    WHERE ord_id = :OLD.ord_id;
END IF;
END;
/

```

```

CREATE OR REPLACE TRIGGER t_11_rental_mng
AFTER INSERT OR UPDATE OR DELETE ON rental_consists_of
FOR EACH ROW
DECLARE
    f_pret      film_mng.film_rental_price%TYPE;
    f_pret_vechi film_mng.film_rental_price%TYPE;
BEGIN
    IF INSERTING THEN
        SELECT film_rental_price INTO f_pret
        FROM film_mng
        WHERE film_id = :NEW.film_id;

        UPDATE rental_mng

```

```
SET rental_price = NVL(rental_price,0) + f_pret
WHERE rental_id = :NEW.rental_id;

ELSIF UPDATING THEN

SELECT film_rental_price INTO f_pret_vechi
FROM film_mng
WHERE film_id = :OLD.film_id;

SELECT film_rental_price INTO f_pret
FROM film_mng
WHERE film_id = :NEW.film_id;

UPDATE rental_mng
SET rental_price = rental_price - f_pret_vechi + f_pret
WHERE rental_id = :NEW.rental_id;

ELSE

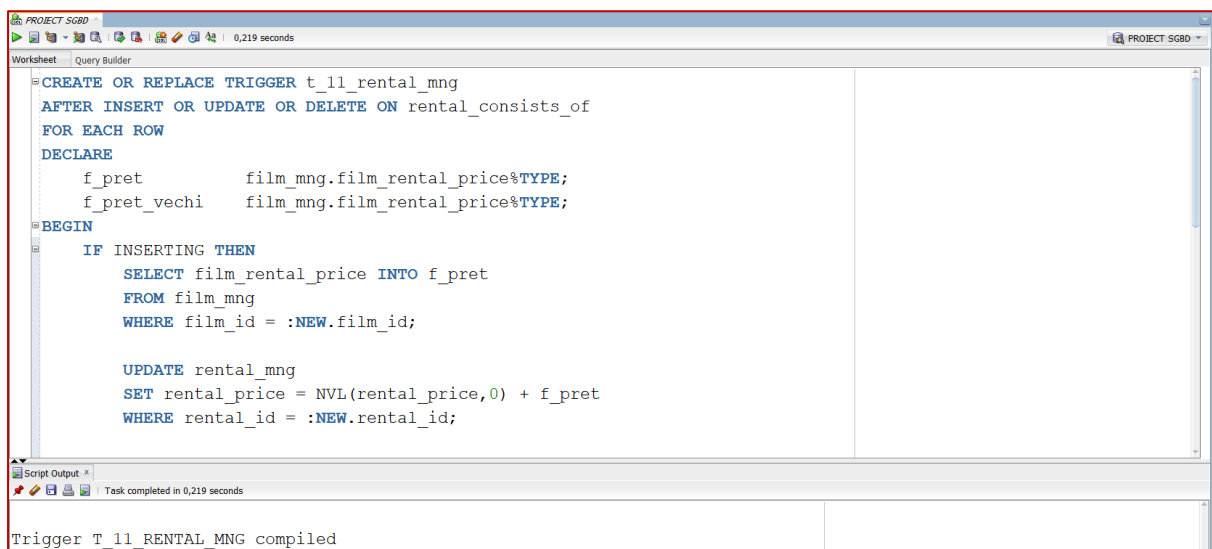
SELECT film_rental_price INTO f_pret_vechi
FROM film_mng
WHERE film_id = :OLD.film_id;

UPDATE rental_mng
SET rental_price = rental_price - f_pret_vechi
WHERE rental_id = :OLD.rental_id;

END IF;

END;

/
```



```

CREATE OR REPLACE TRIGGER t_11_order_mng
AFTER INSERT OR UPDATE OR DELETE ON order_consists_of
FOR EACH ROW
DECLARE
    f_pret          film_mng.film_order_price%TYPE;
    f_pret_vechi    film_mng.film_order_price%TYPE;
BEGIN
    IF INSERTING THEN
        SELECT film_order_price INTO f_pret
        FROM film_mng
        WHERE film_id = :NEW.film_id;

        UPDATE order_mng
        SET ord_price = NVL(ord_price, 0) + (f_pret * :NEW.number_of_films)
    
```

Trigger T\_11\_ORDER\_MNG compiled

Apel – t\_11\_rental\_mng:

```

SELECT *
FROM rental_mng
WHERE rental_id = 3117;

```

RENTAL_ID	RENTAL_DATE	RENTAL_RETURN_DATE	RENTAL_PRICE	EMP_ID	CUST_ID
3117	20-11-2020	27-11-2020	20	505	1015

INSERT INTO rental\_consists\_of  
VALUES(3117, 900);

```

INSERT INTO rental_consists_of
VALUES(3117, 900);

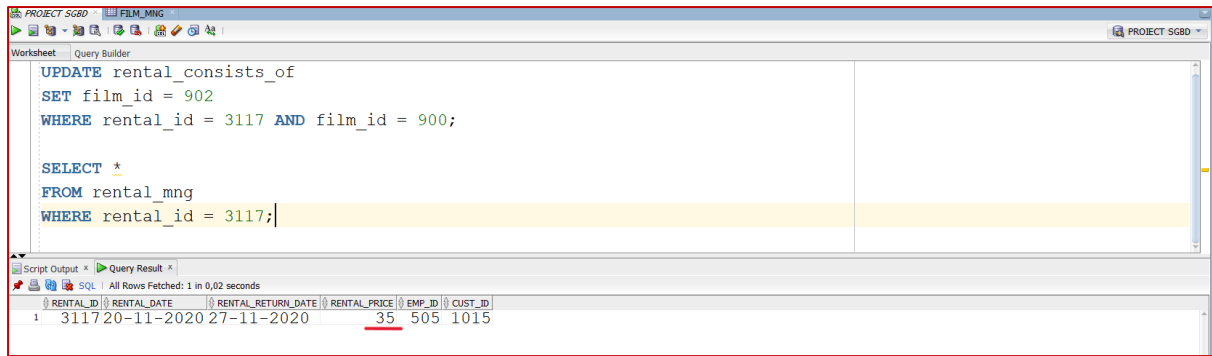
SELECT *
FROM rental_mng
WHERE rental_id = 3117;

```

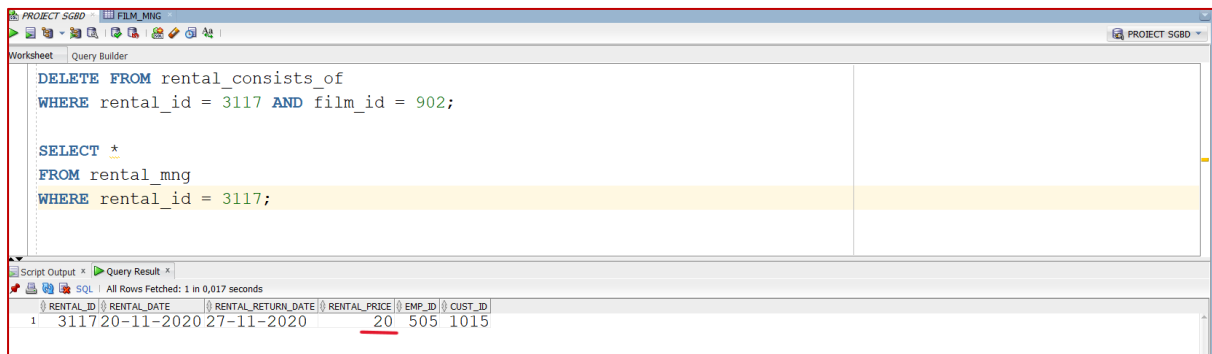
RENTAL_ID	RENTAL_DATE	RENTAL_RETURN_DATE	RENTAL_PRICE	EMP_ID	CUST_ID
3117	20-11-2020	27-11-2020	40	505	1015

UPDATE rental\_consists\_of  
SET film\_id = 902  
WHERE rental\_id = 3117 AND film\_id = 900;

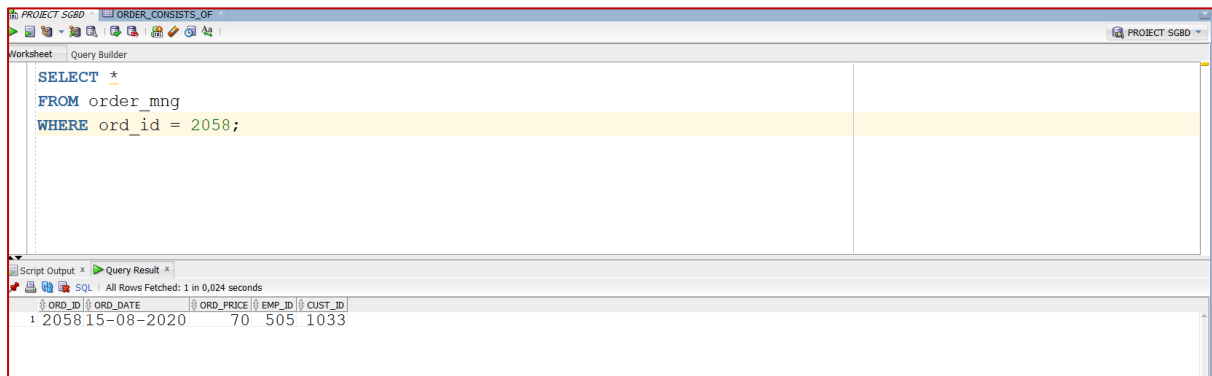




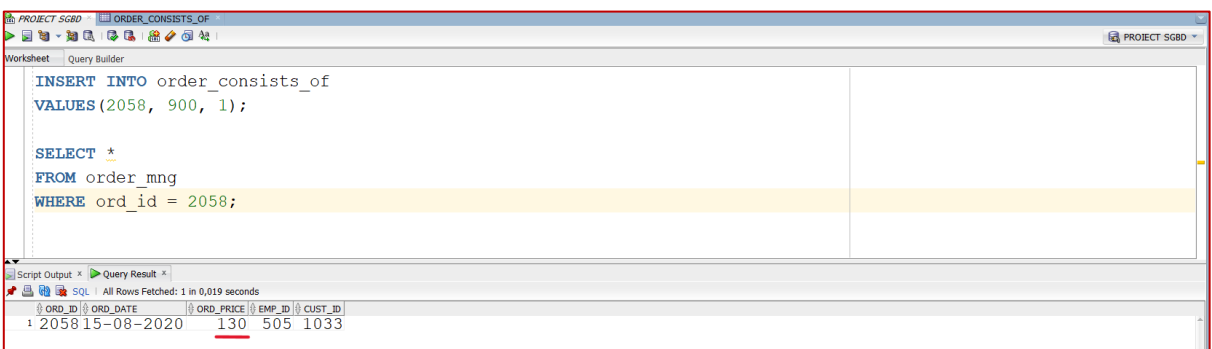
DELETE FROM rental\_consists\_of WHERE rental\_id = 3117 AND film\_id = 902;



Apel t\_11\_order\_mng:



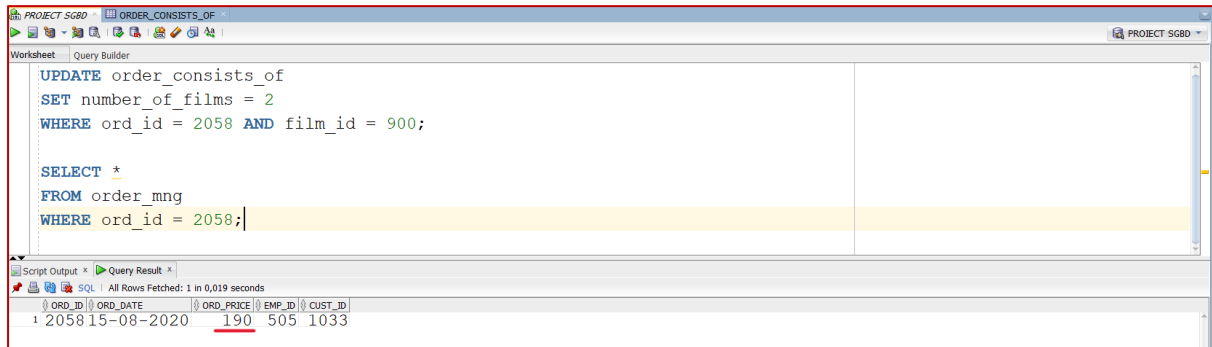
INSERT INTO order\_consists\_of VALUES(2058, 900, 1);



UPDATE order\_consists\_of

SET number\_of\_films = 2

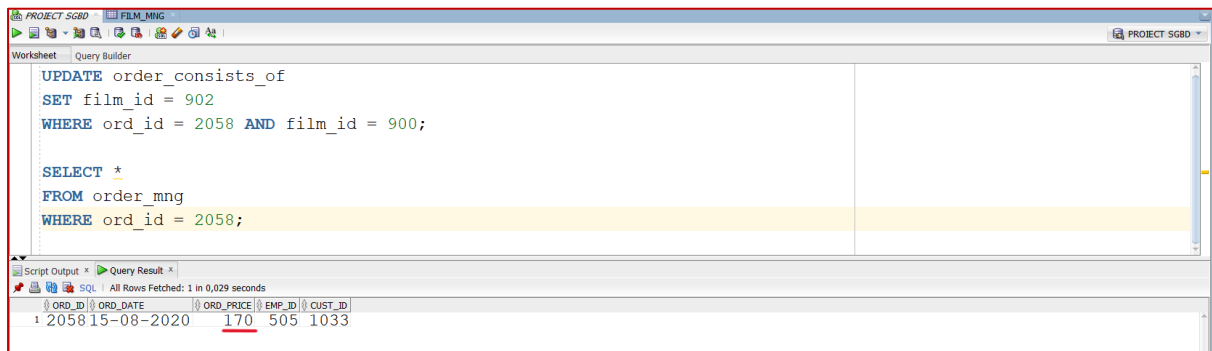
WHERE ord\_id = 2058 AND film\_id = 900;



UPDATE order\_consists\_of

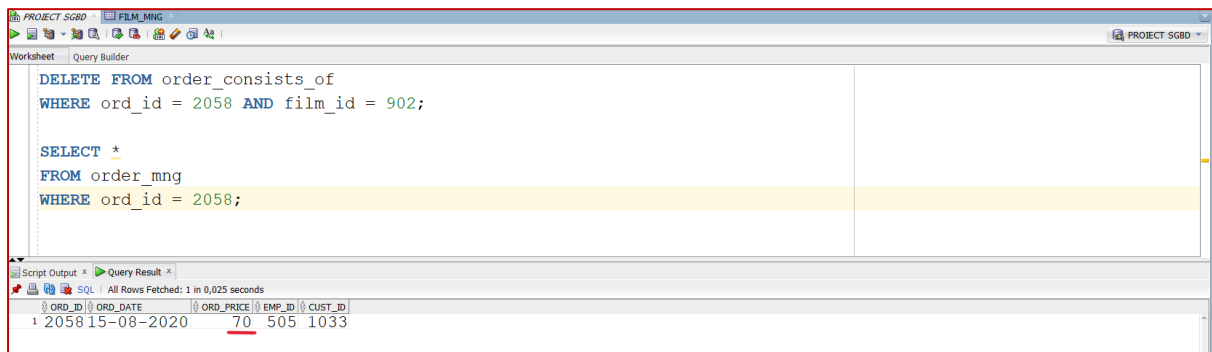
SET film\_id = 902

WHERE ord\_id = 2058 AND film\_id = 900;



DELETE FROM order\_consists\_of

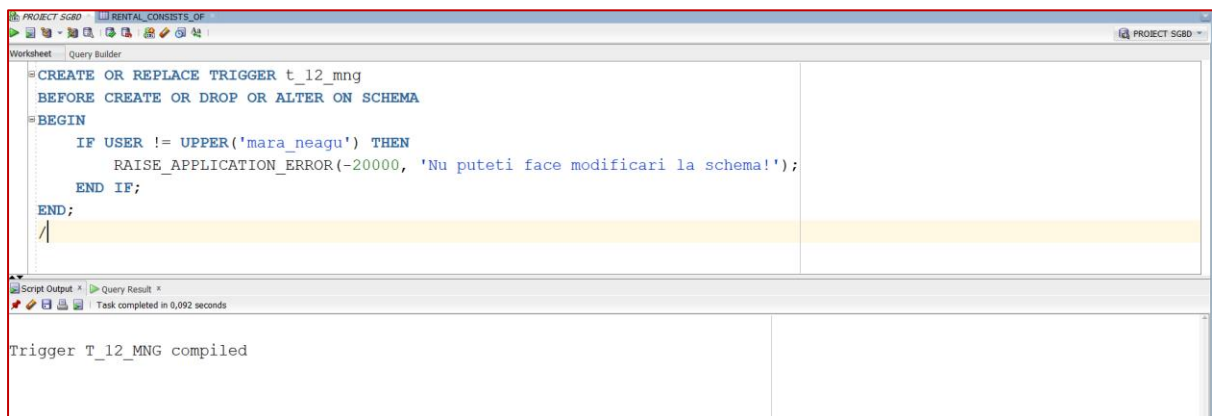
WHERE ord\_id = 2058 AND film\_id = 902;



## 12. Trigger de tip LDD.

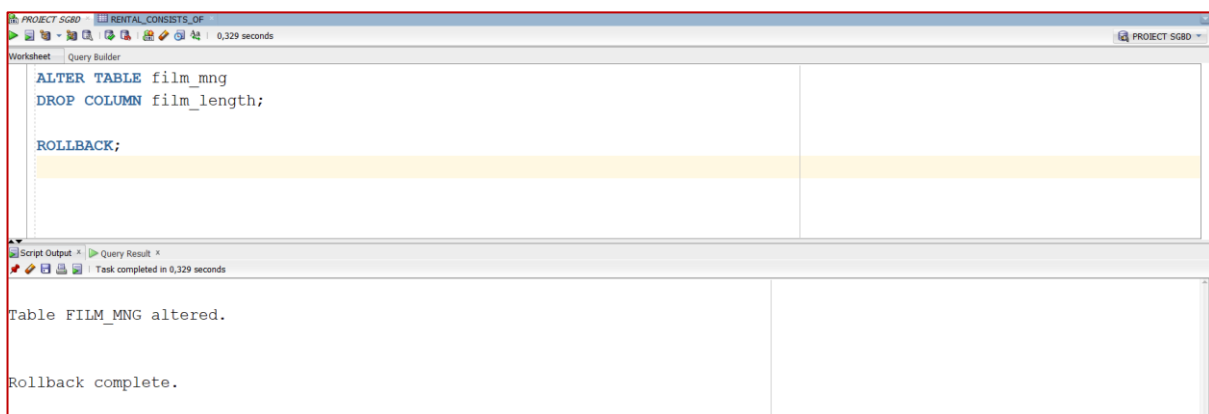
Cerință: Declanșator de tip LDD care permite modificarea schemei doar de către utilizatorul 'mara\_neagu'.

```
CREATE OR REPLACE TRIGGER t_12_mng
BEFORE CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    IF USER != UPPER('mara_neagu') THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu puteti face modificari la
schema!');
    END IF;
END;
/
```



Declanșare:

```
ALTER TABLE film_mng DROP COLUMN film_length;
```



### 13.      **Pachet care conține toate obiectele definite în cadrul proiectului.**

```
CREATE OR REPLACE PACKAGE p_13_mng AS
```

```
  PROCEDURE p_6_mng;
```

```
  PROCEDURE p_7_mng;
```

```
  FUNCTION f_8_mng(s_nume store_mng.store_name%TYPE,  
                  rating film_mng.film_rating%TYPE)
```

```
  RETURN NUMBER;
```

```
  PROCEDURE p_9_mng (nume customer_mng.cust_last%TYPE,  
                    prenume customer_mng.cust_first%TYPE,  
                    limba language_mng.language_name%TYPE);
```

```
END p_13_mng;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY p_13_mng AS
```

```
  PROCEDURE p_6_mng IS
```

```
    TYPE vector IS VARRAY(20) OF film_mng.film_id%TYPE;
```

```
    TYPE tablou_indexat IS TABLE OF actor_mng.actor_id%TYPE INDEX BY  
PLS_INTEGER;
```

```
    f_id      vector;
```

```
    a_id      tablou_indexat;
```

```
    a_nume    actor_mng.actor_last%TYPE;
```

```
    a_prenume actor_mng.actor_first%TYPE;
```

```
  BEGIN
```

```
    SELECT film_id BULK COLLECT INTO f_id
```

```
    FROM film_mng f
```

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

```
    WHERE UPPER(category_name) = 'COMEDIE';
```

```
    FOR i IN 1..f_id.COUNT LOOP
```

```
        SELECT actor_id BULK COLLECT INTO a_id
```

```
        FROM acts_in
```

```

WHERE film_id = f_id(i);

FOR i IN 1..a_id.COUNT LOOP
    SELECT actor_last INTO a_num
    FROM actor_mng
    WHERE actor_id = a_id(i);

    SELECT actor_first INTO a_prenume
    FROM actor_mng
    WHERE actor_id = a_id(i);

    DBMS_OUTPUT.PUT_LINE('Nume: ' || a_num || '; Prenume:
' || a_prenume);

    END LOOP;
END LOOP;
END p_6_mng;

PROCEDURE p_7_mng IS
    CURSOR c_e(s store_mng.store_id%TYPE) IS
        SELECT DISTINCT emp_last nume, emp_first prenume
        FROM employee_mng e
        JOIN works_in w_i ON (w_i.store_id = s AND w_i.emp_id =
e.emp_id)

        JOIN order_mng o ON (e.emp_id = o.emp_id)
        WHERE EXTRACT (YEAR FROM o.ord_date) = 2022;

    e_numar    NUMBER(4);

BEGIN
    FOR s IN (SELECT store_id id, store_name nume
        FROM store_mng) LOOP
        DBMS_OUTPUT.PUT_LINE('- ' || s.nume || ' -');

        e_numar := 0;

```

```

        DBMS_OUTPUT.PUT_LINE('Lista angajati: ');
        FOR e IN c_e(s.id) LOOP
            EXIT WHEN c_e%NOTFOUND;

            DBMS_OUTPUT.PUT_LINE('Angajat: ' || e.numa || ' ' ||
e.prenume);

            e_numa := e_numa + 1;
        END LOOP;

        IF e_numa = 0 THEN
            DBMS_OUTPUT.PUT_LINE('Niciun angajat nu a procesat o
comanda in 2022!');
        END IF;
        DBMS_OUTPUT.NEW_LINE;
    END LOOP;
END p_7_mng;

```

```

FUNCTION f_8_mng(s_nume store_mng.store_name%TYPE,
                rating film_mng.film_rating%TYPE)
RETURN NUMBER IS
    s_nume_invalid    EXCEPTION;
    rating_invalid    EXCEPTION;
    s_id              store_mng.store_id%TYPE;
    numarator        NUMBER(4) := 0;
BEGIN
    IF UPPER(s_nume) NOT LIKE 'THE VELVET STAGE%' THEN
        RAISE s_nume_invalid;
    END IF;

    IF rating < 1 OR rating > 10 THEN
        RAISE rating_invalid;
    END IF;

    SELECT store_id INTO s_id

```

```

FROM store_mng
WHERE store_name = s_num;

SELECT COUNT(*) INTO numarator
FROM store_mng s
JOIN store_inventory s_i ON (s.store_id = s_i.store_id)
JOIN film_mng f ON (s_i.film_id = f.film_id)
JOIN director_mng d ON (f.director_id = d.director_id)
WHERE s.store_id = s_id AND f.film_rating > rating;

RETURN numarator;

EXCEPTION
    WHEN s_num_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Numele
magazinului introdus este invalid!');
    WHEN rating_invalid THEN
        RAISE_APPLICATION_ERROR(-20001, 'Rating-ul introdus
este invalid!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20001, 'Alta eroare!');
END f_8_mng;

PROCEDURE p_9_mng (nume customer_mng.cust_last%TYPE,
                    prenume customer_mng.cust_first%TYPE,
                    language_mng.language_name%TYPE) IS
    c_id          customer_mng.cust_id%TYPE;
    l_id          language_mng.language_id%TYPE;
    TYPE vector IS VARRAY(50) OF film_mng.film_title%TYPE;
    f_titlu      vector;
BEGIN
    SELECT cust_id INTO c_id
    FROM customer_mng

```

```
WHERE UPPER(cust_last) = UPPER(ume) AND UPPER(cust_first) =
UPPER(prenume);
```

```
SELECT language_id INTO l_id
FROM language_mng
WHERE UPPER(language_name) = UPPER(limba);
```

```
SELECT f.film_title BULK COLLECT INTO f_titlu
FROM customer_mng c
JOIN rental_mng r ON (c.cust_id = r.cust_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 translated_to t ON (f.film_id = t.film_id)
WHERE c.cust_id = c_id AND t.language_id = l_id;
```

```
IF f_titlu.COUNT = 0 THEN
    DBMS_OUTPUT.PUT_LINE('Clientul introdus nu a rezervat niciun
film tradus in ' || LOWER(limba));
ELSE
    DBMS_OUTPUT.PUT_LINE('- Lista filme - ');
    FOR i IN 1..f_titlu.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE('Film: ' || f_titlu(i));
    END LOOP;
END IF;
```

```
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20001, 'Datele introduse nu
exista!');
    WHEN TOO_MANY_ROWS THEN
        RAISE_APPLICATION_ERROR(-20002, 'Exista mai multi cu
clienti cu acest nume!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20003, 'Alta eroare!');
```



END p\_9\_mng;

END p\_13\_mng;



The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL script in the main editor is as follows:

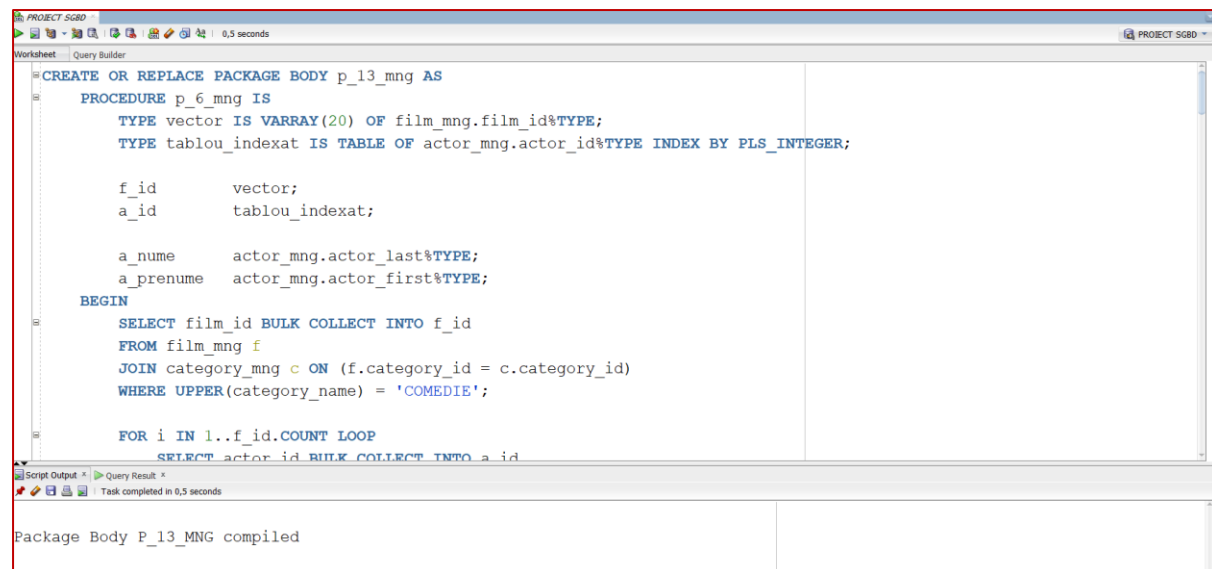
```
CREATE OR REPLACE PACKAGE p_13_mng AS
  PROCEDURE p_6_mng;

  PROCEDURE p_7_mng;

  FUNCTION f_8_mng(s_nume store_mng.store_name%TYPE,
                  rating film_mng.film_rating%TYPE)
  RETURN NUMBER;

  PROCEDURE p_9_mng (nume customer_mng.cust_last%TYPE,
                    prenume customer_mng.cust_first%TYPE,
                    limba language_mng.language_name%TYPE);
END p_13_mng;
```

The status bar at the bottom indicates 'Task completed in 0,179 seconds' and the message 'Package P\_13\_MNG compiled' is displayed in the 'Script Output' pane.



The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The SQL script in the main editor is as follows:

```
CREATE OR REPLACE PACKAGE BODY p_13_mng AS
  PROCEDURE p_6_mng IS
    TYPE vector IS VARRAY(20) OF film_mng.film_id%TYPE;
    TYPE tablou_indexat IS TABLE OF actor_mng.actor_id%TYPE INDEX BY PLS_INTEGER;

    f_id      vector;
    a_id      tablou_indexat;

    a_nume    actor_mng.actor_last%TYPE;
    a_prenume actor_mng.actor_first%TYPE;
  BEGIN
    SELECT film_id BULK COLLECT INTO f_id
    FROM film_mng f
    JOIN category_mng c ON (f.category_id = c.category_id)
    WHERE UPPER(category_name) = 'COMEDIE';

    FOR i IN 1..f_id.COUNT LOOP
      SELECT actor_id BULK COLLECT INTO a_id

```

The status bar at the bottom indicates 'Task completed in 0,5 seconds' and the message 'Package Body P\_13\_MNG compiled' is displayed in the 'Script Output' pane.

## 14. Pachet care include tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate.

Cerință: Creați un pachet care să conțină:

- ⊗ procedură care pentru un magazin din franciză, calculează media profitului vânzării unui DVD pe perioada de primăvara; dacă aceasta este mai mică decât 50, prețul achiziționării unui DVD va scădea cu 15%;
- ⊗ funcție care pentru un magazin din franciză, o lună și un an, returnează angajatul lunii respective, mai precis angajatul care a procesat cel mai mare număr de comenzi și închirieri pe acea lună;
- ⊗ procedură care pentru un magazin din franciză, afișează topul celor 5 filme ale căror DVD-uri au fost vândute și închiriate;
- ⊗ funcție care pentru un magazin din franciză, o lună și o categorie, calculează media profitului anual al vânzărilor, pentru acea categorie de film.

```
CREATE OR REPLACE PACKAGE p_14_mng AS
```

```
    PROCEDURE p1_14_mng (s_id store_mng.store_id%TYPE);
```

```
    FUNCTION f1_14_mng (s_id store_mng.store_id%TYPE,
                        luna NUMBER,
                        an NUMBER)
```

```
    RETURN employee_mng.emp_id%TYPE;
```

```
    PROCEDURE p2_14_mng (s_id store_mng.store_id%TYPE);
```

```
    FUNCTION f2_14_mng (s_id store_mng.store_id%TYPE,
                        an NUMBER,
                        categorie category_mng.category_name%TYPE)
```

```
    RETURN NUMBER;
```

```
END p_14_mng;
```

```
/
```

```
CREATE OR REPLACE PACKAGE BODY p_14_mng AS
```

```
    PROCEDURE p1_14_mng (s_id store_mng.store_id%TYPE) IS
```

```

s_id_invalid    EXCEPTION;

o_profit        NUMBER(4);
o_numar         NUMBER(4);
m_profit        NUMBER(4);

TYPE vector IS VARRAY(30) OF film_mng.film_id%TYPE;
f_id            vector;
BEGIN
    IF s_id > 5 OR s_id < 1 THEN
        RAISE s_id_invalid;
    END IF;

    SELECT SUM(ord_price) INTO o_profit
    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_id
    AND EXTRACT (MONTH FROM ord_date) IN (3,4,5)
    AND EXTRACT (YEAR FROM ord_date) = EXTRACT (YEAR FROM
SYSDATE);

    SELECT COUNT(ord_id) INTO o_numar
    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_id
    AND EXTRACT (MONTH FROM ord_date) IN (3,4,5)
    AND EXTRACT (YEAR FROM ord_date) = EXTRACT (YEAR FROM
SYSDATE);

    IF NVL(o_numar,0) = 0 THEN
        DBMS_OUTPUT.PUT_LINE('Magazinul introdus nu a procesat nicio
comanda in primavara!');

```

```

ELSE
    m_profit := o_profit / o_numar;

    IF m_profit < 50 THEN
        SELECT film_id BULK COLLECT INTO f_id
        FROM store_inventory
        WHERE store_id = s_id;

        FOR i IN 1..f_id.COUNT LOOP
            UPDATE film_mng
            SET    film_order_price    =    film_order_price    -
(film_order_price * 15 / 100)
            WHERE film_id = f_id(i);
        END LOOP;

        DBMS_OUTPUT.PUT_LINE('Profit: ' || m_profit || ', update
realizat!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Profit: ' || m_profit || ', update
nerealizat!');
    END IF;
END IF;

EXCEPTION
    WHEN s_id_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Codul magazinului
introdus este invalid!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000, 'Alta eroare!');
END p1_14_mng;

FUNCTION f1_14_mng (s_id store_mng.store_id%TYPE,
                    luna NUMBER,

```

```

        an NUMBER)

RETURN employee_mng.emp_id%TYPE IS
    s_id_invalid  EXCEPTION;
    luna_invalida EXCEPTION;
    an_invalid    EXCEPTION;

    e_id          employee_mng.emp_id%TYPE;
BEGIN
    IF s_id > 5 OR s_id < 1 THEN
        RAISE s_id_invalid;
    END IF;

    IF luna > 12 OR luna < 1 THEN
        RAISE luna_invalida;
    END IF;

    IF an > EXTRACT (YEAR FROM SYSDATE) OR an < 2014 THEN
        RAISE an_invalid;
    END IF;

    SELECT e_cod INTO e_id
    FROM (SELECT e.emp_id e_cod, ((SELECT COUNT(*) FROM order_mng o
    WHERE o.emp_id = e.emp_id AND EXTRACT (MONTH FROM o.ord_date) = luna AND
    EXTRACT (YEAR FROM o.ord_date) = an) +
        (SELECT COUNT(*) FROM rental_mng r WHERE r.emp_id =
    e.emp_id AND EXTRACT (MONTH FROM r.rental_date) = luna AND EXTRACT (YEAR
    from r.rental_date) = an)) e_numar
    FROM employee_mng e
    JOIN works_in w_i ON (e.emp_id = w_i.emp_id)
    WHERE w_i.store_id = s_id
    ORDER BY e_numar DESC)
    WHERE ROWNUM = 1;

    RETURN e_id;

```

```

EXCEPTION
    WHEN s_id_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Codul magazinului
introdus este invalid!');
    WHEN luna_invalida THEN
        RAISE_APPLICATION_ERROR(-20000, 'Luna introdusa este
invalida!');
    WHEN an_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Anul introdus este
invalid!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000, 'Alta eroare!');
END f1_14_mng;

PROCEDURE p2_14_mng (s_id store_mng.store_id%TYPE) IS
    s_id_invalid  EXCEPTION;

    TYPE tablou_indexat IS TABLE OF film_mng.film_title%TYPE INDEX BY
PLS_INTEGER;

    f_titlu      tablou_indexat;

    s_numa      store_mng.store_name%TYPE;
    f_numar      NUMBER(4):= 0;
BEGIN
    IF s_id > 5 OR s_id < 1 THEN
        RAISE s_id_invalid;
    END IF;

    SELECT store_name INTO s_numa
    FROM store_mng
    WHERE store_id = s_id;

    SELECT f_titlu BULK COLLECT INTO f_titlu

```

```

FROM (SELECT f.film_title f_titlu, ((SELECT COUNT(*) FROM
order_consists_of o WHERE o.film_id = f.film_id) +
      (SELECT COUNT(*) FROM rental_consists_of r WHERE
r.film_id = f.film_id)) f_numar
      FROM film_mng f
      JOIN store_inventory s_i ON (f.film_id = s_i.film_id)
      WHERE s_i.store_id = s_id
      ORDER BY f_numar DESC)
WHERE ROWNUM < 6;

```

```

DBMS_OUTPUT.PUT_LINE(s_ume || ' - Top 5 filme -');
FOR i IN 1..f_titlu.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE(f_numar + 1 || ' ' || f_titlu(i));
    f_numar := f_numar + 1;
END LOOP;

```

```

EXCEPTION
    WHEN s_id_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Codul magazinului
introdus este invalid!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000, 'Alta eroare!');
END p2_14_mng;

```

```

FUNCTION f2_14_mng (s_id store_mng.store_id%TYPE,
                    an NUMBER,
                    categorie category_mng.category_name%TYPE)
RETURN NUMBER IS
    s_id_invalid    EXCEPTION;
    an_invalid      EXCEPTION;
    c_id            category_mng.category_id%TYPE;

    r_profit        NUMBER(4);

```

```

r_numar      NUMBER(4);
o_profit     NUMBER(4);
o_numar      NUMBER(4);
m_profit     NUMBER(4);

```

BEGIN

```

IF s_id > 5 OR s_id < 1 THEN
    RAISE s_id_invalid;
END IF;

```

```

IF an > EXTRACT (YEAR FROM SYSDATE) OR an < 2014 THEN
    RAISE an_invalid;
END IF;

```

```

SELECT category_id INTO c_id
FROM category_mng
WHERE UPPER(category_name) = UPPER(categorie);

```

```

SELECT SUM(r.rental_price) INTO r_profit
FROM rental_mng r
JOIN rental_consists_of r_c_o ON (r.rental_id = r_c_o.rental_id)
JOIN film_mng f ON (r_c_o.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE c.category_id = c_id AND EXTRACT (YEAR FROM r.rental_date)
= an;

```

```

SELECT COUNT(r.rental_id) INTO r_numar
FROM rental_mng r
JOIN rental_consists_of r_c_o ON (r.rental_id = r_c_o.rental_id)
JOIN film_mng f ON (r_c_o.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE c.category_id = c_id AND EXTRACT (YEAR FROM r.rental_date)
= an;

```

```

SELECT SUM(o.ord_price) INTO o_profit

```



```

FROM order_mng o
JOIN order_consists_of o_c_o ON (o.ord_id = o_c_o.ord_id)
JOIN film_mng f ON (o_c_o.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE c.category_id = c_id AND EXTRACT (YEAR FROM o.ord_date) =
an;

```

```

SELECT COUNT(o.ord_id) INTO o_numar
FROM order_mng o
JOIN order_consists_of o_c_o ON (o.ord_id = o_c_o.ord_id)
JOIN film_mng f ON (o_c_o.film_id = f.film_id)
JOIN category_mng c ON (f.category_id = c.category_id)
WHERE c.category_id = c_id AND EXTRACT (YEAR FROM o.ord_date) =
an;

```

```

m_profit := (NVL(r_profit, 0) + NVL(o_profit, 0)) / (NVL(r_numar, 0) +
NVL(o_numar, 0));
RETURN m_profit;

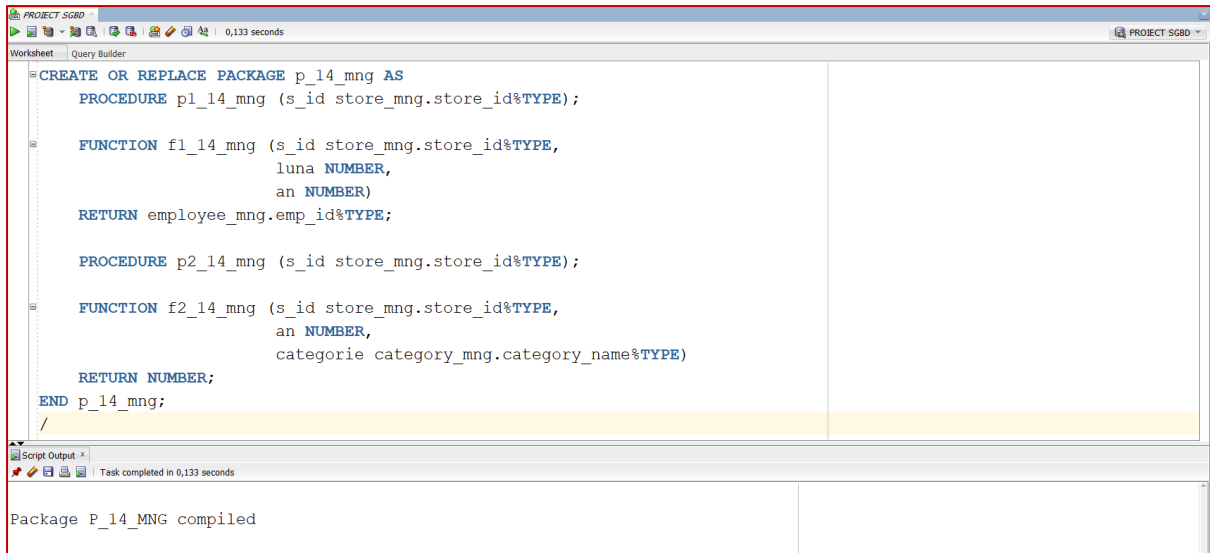
```

```

EXCEPTION
    WHEN s_id_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Codul magazinului
introdus este invalid!');
    WHEN an_invalid THEN
        RAISE_APPLICATION_ERROR(-20000, 'Anul introdus este
invalid!');
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Categoria introdusa
este invalida!');
    WHEN OTHERS THEN
        RAISE_APPLICATION_ERROR(-20000, 'Alta eroare!');

END f2_14_mng;
END p_14_mng;

```



The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main window displays the following SQL code:

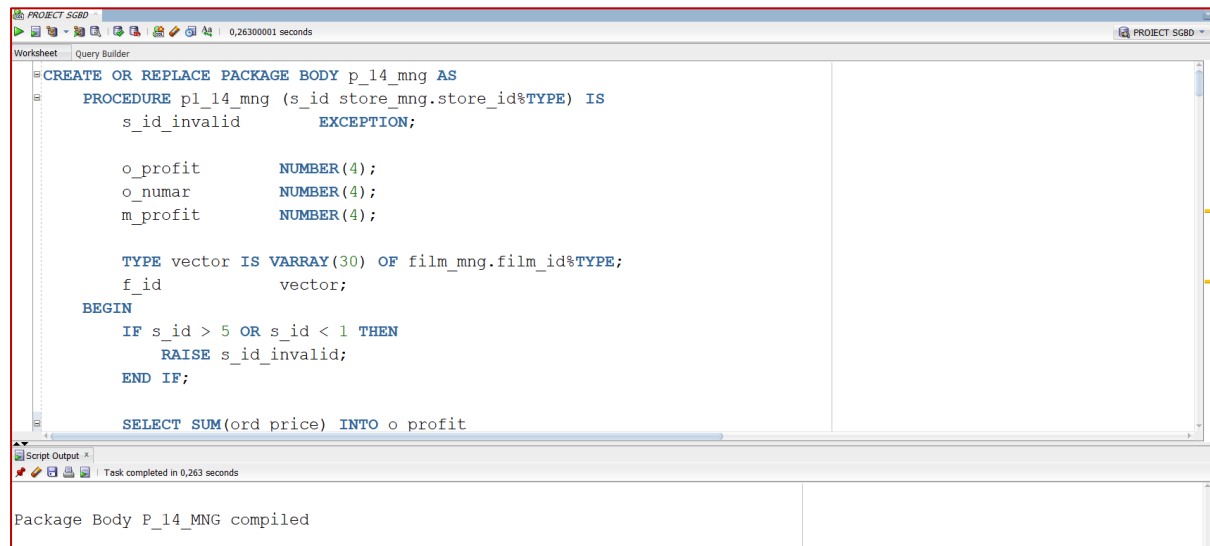
```
CREATE OR REPLACE PACKAGE p_14_mng AS
  PROCEDURE p1_14_mng (s_id store_mng.store_id%TYPE);

  FUNCTION f1_14_mng (s_id store_mng.store_id%TYPE,
                    luna NUMBER,
                    an NUMBER)
  RETURN employee_mng.emp_id%TYPE;

  PROCEDURE p2_14_mng (s_id store_mng.store_id%TYPE);

  FUNCTION f2_14_mng (s_id store_mng.store_id%TYPE,
                    an NUMBER,
                    categorie category_mng.category_name%TYPE)
  RETURN NUMBER;
END p_14_mng;
/
```

The 'Script Output' window at the bottom shows the message: 'Package P\_14\_MNG compiled'. The status bar indicates 'Task completed in 0,133 seconds'.



The screenshot shows the SQL Developer interface with the 'Query Builder' tab active. The main window displays the following SQL code:

```
CREATE OR REPLACE PACKAGE BODY p_14_mng AS
  PROCEDURE p1_14_mng (s_id store_mng.store_id%TYPE) IS
    s_id_invalid          EXCEPTION;

    o_profit              NUMBER(4);
    o_numar               NUMBER(4);
    m_profit              NUMBER(4);

    TYPE vector IS VARRAY(30) OF film_mng.film_id%TYPE;
    f_id                 vector;
  BEGIN
    IF s_id > 5 OR s_id < 1 THEN
      RAISE s_id_invalid;
    END IF;

    SELECT SUM(ord price) INTO o profit
```

The 'Script Output' window at the bottom shows the message: 'Package Body P\_14\_MNG compiled'. The status bar indicates 'Task completed in 0,263 seconds'.

Apel – p\_14\_mng:

BEGIN

p\_14\_mng.p1\_14\_mng(1);

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE(p\_14\_mng.f1\_14\_mng(1, 3, 2019));

DBMS\_OUTPUT.NEW\_LINE;

```
p_14_mng.p2_14_mng(1);  
DBMS_OUTPUT.NEW_LINE;
```

```
DBMS_OUTPUT.PUT_LINE(p_14_mng.f2_14_mng(1, 2019, 'COMEDIE'));
```

```
END;
```

```
/
```

