

Baza de date pentru Disney+

Iștoc Simona Maria

Grupa 131

Cuprins

Exercițiul 1.....	3
Exercițiul 2.....	3
Exercițiul 3.....	4
Exercițiul 4.....	4
Exercițiul 5.....	6
Exercițiul 6.....	10
Exercițiul 7.....	11
Exercițiul 8.....	12
Exercițiul 9.....	12
Exercițiul 10.....	14
Exercițiul 11.....	14
Exercițiul 12.....	33
Exercițiul 13.....	37
Exercițiul 14.....	40
Exercițiul 15.....	41

Exercițiul 1: Descrierea modelului real, al utilității acestuia și a regulilor de funcționare

Disney+ este un serviciu de streaming deținut de Walt Disney Company care conține numeroase filme și seriale produse de aceasta companie precum și alte branduri cum ar fi Pixar, Marvel, Star Wars și National Geographic.

Baza de date cuprinde informații cu privire la gestionarea platformei Disney+, a abonaților, subscripțiile, filmele, seriarele, directorii și actorii.

Un utilizator își poate face un singur abonament și trebuie să aleagă una din metodele de plată (pe an, pe luna). Un utilizator poate urmări orice film și serial de pe platformă atât timp cât are un abonament și acesta este cu plata la zi.

Fiecare film/serial are un director și unul sau mai mulți actori. Fiecare actor joacă în unul sau mai multe filme/seriale. Fiecare actor are un rol într-un film/serial, iar fiecare film/serial are unul sau mai multe roluri.

Exercițiul 2: Prezentarea contrângerilor (restricții, reguli) impuse asupra modelului.

- Un utilizator poate face un singur abonament, pe baza unui email.
- Un abonament are un singur tip de plată (card) și poate fi plătită în diferite moduri (pe luna, pe an)
- Dacă data de expirare a plății este mai mică decât sysdate abonamentul este suspendat până la plată.
- Un utilizator poate adăuga o singură metodă de plată.
- Un abonament poate avea un profil sau până la maxim șapte profile.
- O producție poate fi un serial sau un film.
- O producție este regizată de unul sau mai mulți directori, dar un director poate regiza mai multe producții.
- O producție poate fi vizionată de mai multe profile, și un profil poate urmări mai multe producții.
- Un serial poate să aibă mai multe episoade, dar un episod are un singur serial.
- O producție are mai mulți actori, și un actor poate juca în mai multe producții.
- O producție are mai multe roluri, și un rol poate apărea în mai multe producții.
- Un actor poate avea mai multe roluri, și un rol poate fi jucat de mai mulți actori.

Exercițiul 3: Descrierea entităților, incluzând precizarea cheii primare

UTILIZATOR = persoană fizică care își poate crea un cont pe baza unui email. Entitatea depinde de plată și abonament. Cheia primară este utilizator_id.

PLATA = entitate în care ținem informațiile cardului cu care plătește utilizatorul. Cheia primară este plată_id.

ABONAMENT = pachet care conține filmele și seriile disponibile pe Disney+. Cheia primară este abonament_id.

PROFIL = metodă de partajare a unui abonament între mai multe persoane. Entitatea depinde de abonament_id. Cheia primară este profil_id.

PRODUCȚIE = colecție de conținut video creat pentru divertisment conținând filme și seriale. Cheia primară este producție_id.

FILM = producție cinematografică care spun o poveste completă într-o singură lucrare separate pe mai multe genuri. Cheia primară este productie_id.

SERIAL = producții episodice care spun o poveste pe parcursul mai multor episoade și sezoane cu durată și structură variabilă, de asemenea separate pe mai multe genuri. Cheia primară este productie_id.

EPISOD = parte individuală dintr-un serial. Cheia primară este episod_id

DIRECTOR = persoană fizică responsabilă pentru supravegherea și coordonarea tuturor aspectelor creative ale unei producții. Cheia primară este director_id

ACTOR = persoană fizică care interpretează roluri în producții. Cheia primară este actor_id

ROL = un personaj pe care un actor îl interpretează într-o producție. Cheia primară este rol_id

Exercițiul 4: Descrierea relațiilor, incluzând precizarea cardinalității acestora.

UTILIZATOR are ABONAMENT

- Un utilizator poate să aleagă un singur abonament
- Un abonament poate să fie ales de zero sau mai mulți utilizatori
- Cardinalitate maximă: M:1
- Cardinalitate minimă: 0:1

UTILIZATOR adaugă o PLATĂ

- Un utilizator poate să adauge o singură plată
- O plată poate fi adaugată de 1 sau mai mulți utilizatori
- Cardinalitate maximă: M:1
- Cardinalitate minimă: 1:1

ABONAMENT are un PROFIL

- Un utilizator poate să aibă 1 sau mai multe profile (maxim 7 profile)
- Un profil este asociat unui singur abonament
- Cardinalitate maximă: 1:1
- Cardinalitate minimă: M:1

PRODUȚIE are ACTORI

- O producție poate avea zero sau mai mulți actori
- Un actor poate juca în zero sau mai multe producții
- Cardinalitate maximă: M:0
- Cardinalitate minimă: M:0

PRODUȚIE are ROLURI

- O producție poate avea zero sau mai multe roluri
- Un rol poate apărea în zero sau mai multe producții
- Cardinalitate maximă: M:M
- Cardinalitate minimă: 0:0

ACTORI are ROLURI

- Un actor poate avea zero sau mai multe roluri
- Un rol poate fi jucat de zero sau mai mulți actori
- Cardinalitate maximă: M:M
- Cardinalitate minimă: 0:0

PROFIL vizionează PRODUȚIE

- Un profil poate viziona zero sau mai multe producții
- O producție poate fi vizionată de una sau mai multe producții

- Cardinalitate maximă: M:M
- Cardinalitate minimă: 0:0

DIRECTOR regizează PRODUCȚIE

- Un director regizează una sau mai multe producții
- O producție este regizată de unul sau mai mulți directori
- Cardinalitate maximă: M:M
- Cardinalitate minimă: 1:1

SERIAL are EPISOD

- Un serial are unul sau mai multe episoade
- Un episod face parte dintr-un serial
- Cardinalitate maxima: M:1
- Cardinalitate minimă: 1:1

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

- UTILIZATOR
 - Utilizator_id = NUMBER(6) – codul utilizatorului
 - Plata_id = cheie străină din tabela PLATĂ
 - Abonament_id = cheie straină din tabela ABONAMENT
 - Mail = VARCHAR2(50) – mail-ul cu care se înscrie un utilizator
 - Parola = VARCHAR2(50) – parola asociată mail-ului unui utilizator
 - Data_creare = DATE (valoare implicită = data curentă) – data creării unui abonament
 - Telefon = VARCHAR2 (16) – telefonul asociat unui utilizator

Utilizator_id	Plata_id	Abonament_id	Mail	Parola	Data_creare	Telefon
PK	FK, NOT NULL	FK, NOT NULL	NOT NULL, UNIQUE	NOT NULL	NOT NULL	-

- ABONAMENT
 - Abonament_id = NUMBER(6) - codul unui abonament, care va fi asociat unui utilizator
 - Tip_plată = VARCHAR2(10) – modul de plata care poate fi lunar, anual
 - Data_expirare_plată = DATE (+30 de zile pentru abonament lunar, +356 de zile pentru abonament annual)

Abonament_id	Tip_plata	Data_expirare_plată
PK	NOT NULL	NOT NULL

- PROFIL

- Profil_id= NUMBER(6) – codul unui profil, asociat unui abonament
- Poreclă = VARCHAR(20) – nume asociat unui profil
- Abonament_id = cheie străină din tabela ABONAMENT
- Data_creare = DATE (valoare implicită =data curentă) – data creării unui profil

Profil_id	Poreclă	Abonament_id	Data_creare
PK	NOT NULL	FK, NOT NULL	NOT NULL

- PLATĂ

- Plata_id = NUMBER(6) – cod unic platii, asociat unui utilizator
- Detinator = VARCHAR2(50) – numele scris pe spatele cardului
- Cod = NUMBER(16) – codul cardului
- Data_expirare = DATE – data de expirare a cardului
- CVV = NUMBER(3) – CVV-ul cardului

Plata_id	Detinator	Cod	Data_expirare	CVV
PK	NOT NULL	NOT NULL, UNIQUE	NOT NULL	NOT NULL

- PRODUCȚIE

- Producție_id= NUMBER(6) – cod unic unei producții
- Denumire = VARCHAR2(50) – titlul unei producții
- Gen = VARCHAR2(50) – genul unei producții
- Data_apariție = DATE – data apariției unei producții
- Director_id = cheie primară din DIRECTOR

Producție_id	Denumire	Gen	Data_apariție	Director_id
PK	NOT NULL	NOT NULL	NOT NULL	FK, NOT NULL

- FILM – subentitate a PRODUCȚIE

- Productie_id = NUMBER(6) – cod unic al filmului
- Durata = NUMBER(3) – durata a filmului în minute

Productie_id	Durata
PK(FK)	NOT NULL

- SERIAL – subentitate a PRODUCȚIE

- Productie_id = NUMBER(6) – cod unic al serialului
- Numar_sezoane = NUMBER(2) – numar de sezoane al serialului

Productie_id	Numar_sezoane
PK(FK)	NOT NULL

- EPISOD

- Episod_id = NUMBER(6) – cod unic al episodului
- Număr_episod = NUMBER(2) – numarul episodului
- Număr_sezon = NUMBER(2) – numarul sezonului
- Titlu = VARCHAR2(50) – titlul episodului
- Durata = NUMBER(3) – durata episodului în minute
- Productie_id = cheie străină din tabela SERIAL

Episod_id	Numar_episod	Numar_sezon	Titlu	Durata	Productie_id
PK	NOT NULL	NOT NULL	NOT NULL	NOT NULL	FK, NOT NULL

- ACTOR

- Actor_id = NUMBER(5) – cod unic al actorului
- Nume = VARCHAR2(50) – numele actorului
- Prenume = VARCHAR2(50) – prenumele actorului
- Data_nastere = DATE – data de nastere a actorului

Actor_id	Nume	Prenume	Data_naștere
PK	NOT NULL	NOT NULL	-

- ROL

- Rol_id = NUMBER(5) – cod unic al rolului
- Nume_caracter = VARCHAR2(50) – numele caracterului
- Importanță = VARCHAR(50) – importanta personajului (principal, secundar, etc)
- Descriere = VARCHAR(200) – scurtă descriere a rolului

Rol_id	Nume_Caracter	Importanță	Descriere
PK	NOT NULL	NOT NULL	-

- DIRECTOR

- Director_id = NUMBER(5) – cod unic al directorului
- Nume = VARCHAR2(50) – numele directorului
- Prenume = VARCHAR2(50) – prenumele directorului
- Data_nastere = DATE – data de naștere a directorului

Director_id	Nume	Prenume	Data_naștere
PK	NOT NULL	NOT NULL	-

- ROL_PRODUCTIE_JUCAT

- Rol_productie_actor_id = NUMBER(7) – cod unic pentru rolul jucat de un actor într-un anumit film
- Productie_id = cheie străină din tabela FILM
- Rol_id = cheie străină din tabela ROL
- Actor_id = cheie străină din tabela ACTOR
- Timp_ecran = NUMBER(3) – durata totala (in minute) în care un actor apare într-o producție

Rol_productie_actor_id	Productie_id	Rol_id	Actor_id	Timp_ecran
PK	FK, NOT NULL	FK, NOT NULL	FK, NOT NULL	-

- PROFIL_PRODUCTIE

- Profil_productie_id = NUMBER(6) – cod unic pentru producțiile vizionate de un profil
- Profil_id = cheie străină din tabela PROFIL
- Producție_id = cheie străină din tabela PRODUCȚIE

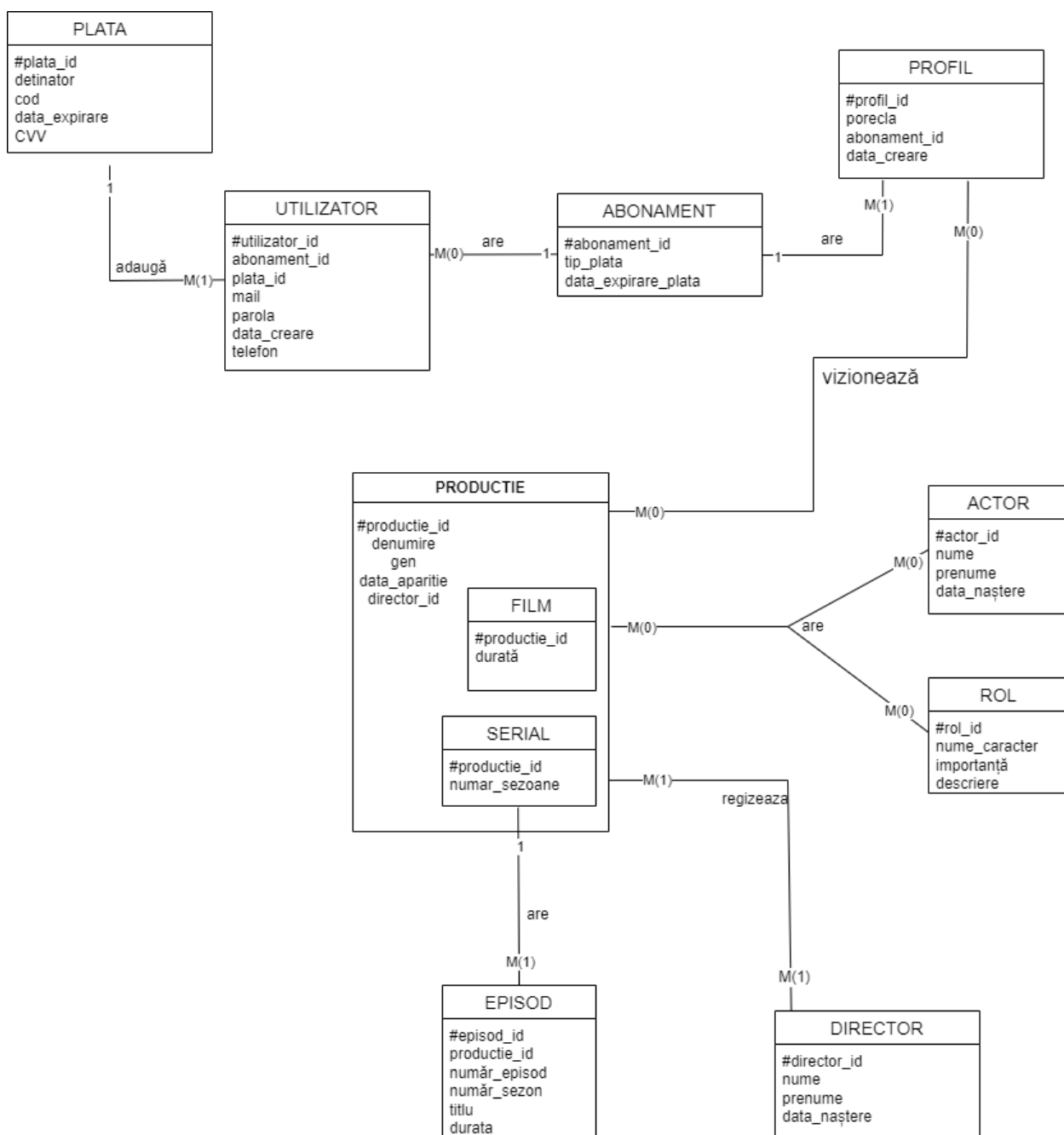
Profil_productie_id	Profil_id	Productie_id
PK	FK, NOT NULL	FK, NOT NULL

- PRODUCTIE_DIRECTOR

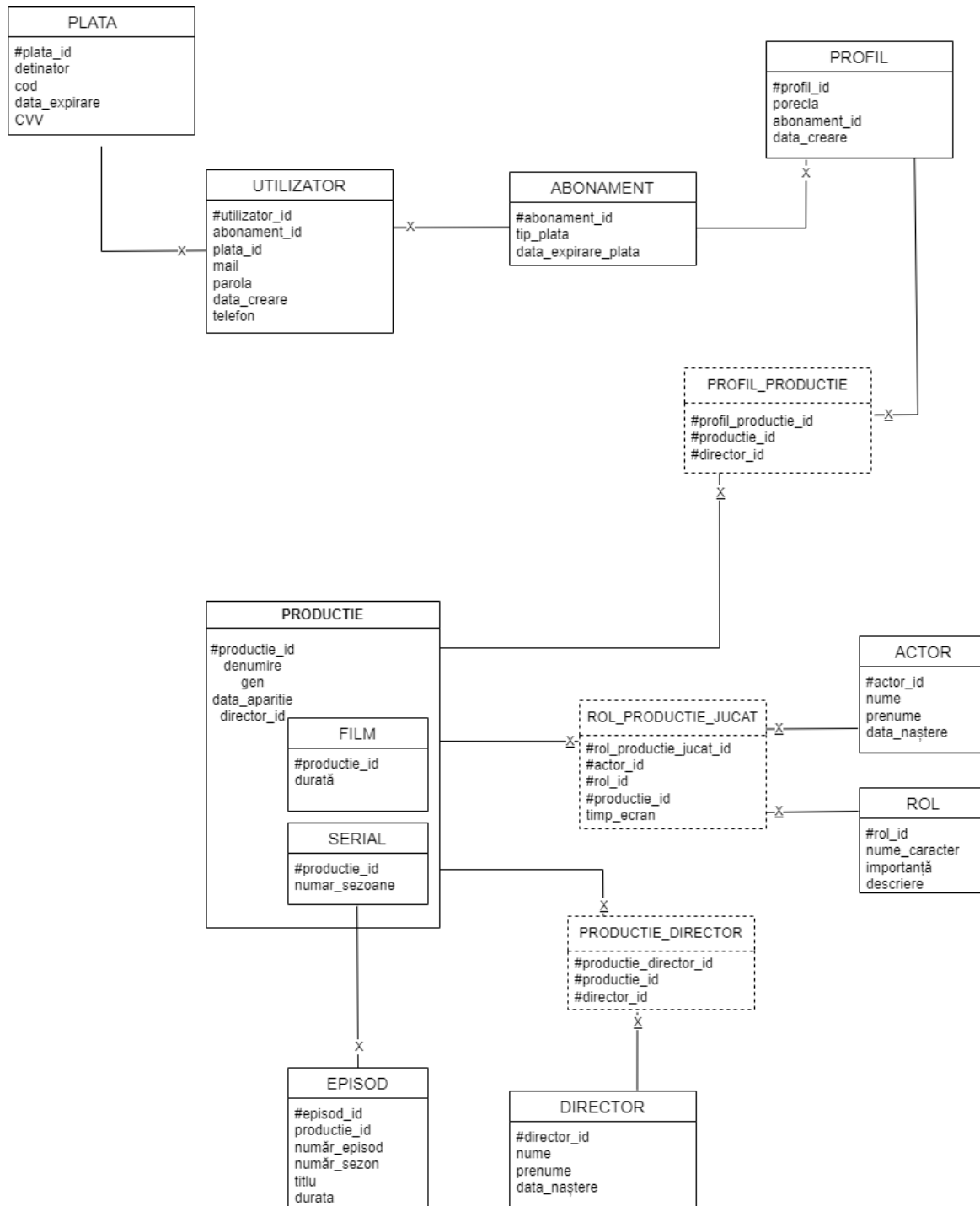
- Productie_director_id = NUMBER(6) – cod unic pentru directorul care a regizat o anumita producție
- Productie_id = cheie străină din tabela PRODUCȚIE
- Director_id = cheie străină din tabela DIRECTOR

Productie_director_id	Productie_id	Director_id
PK	FK, NOT NULL	FK, NOT NULL

Exercițiul 6: Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



Exercițiul 7: Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



Exercițiul 8: Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

PLATA(#plata_id, detinator, cod, data_expirare, CVV)

UTILIZATOR(#utilizator_id, #abonament_id, #plata_id, mail, parola, data_creare, telefon)

ABONAMENT(#abonament_id, tip_plata, data_expirare_plata)

PROFIL(#profil_id, porecla, abonament_id, data_creare)

PRODUCTIE(#productie_id, denumire, gen, data_aparitie, #director_id)

FILM(#productie_id, durată)

SERIAL(#productie_id, numar_sezoane)

ACTOR(#actor_id, nume, prenume, data_naștere)

ROL(#rol_id, nume_caracter, importanță, descriere)

EPISOD(#episod_id, #productie_id, număr_episod, număr_sezon, titlu, durata)

DIRECTOR(#director_id, nume, prenume, data_naștere)

PROFIL_PRODUCTIE(#profil_productie_id, #productie_id, #director_id)

PRODUCTIE_DIRECTOR(#productie_director_id, #productie_id, #director_id)

ROL_PRODUCTIE_JUCAT(#rol_productie_jucat_id, #actor_id, #rol_id, #productie_id, timp_ecran)

Exercițiul 9: Realizarea normalizării până la forma normală 3 (FN1-FN3).

FN1

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

În cadrul acestei baze de date, voi utiliza un exemplu pentru a prezenta aflarea acesteia în forma normală 1.

Voi lua entitățile ABONAMENT și PROFIL.

În cadrul unui abonament putem avea mai multe profile, iar diferite abonamente pot avea profile cu același nume.

ABONAMENT	PROFIL
Abonament1	Profil1, profil2, profil3
Abonament2	Profil1, profil4
Abonament3	Profil5

Pentru a selecta înregistrări pe baza câmpurilor care conțin mai multe valori este foarte dificil de implementat. Spre exemplu, pentru a cauta abonamentele care conțin profilul cu numele „Profil1” ar trebui să parcurgem fiecare sir de profile și să identificăm subsirul căutat.

ABONAMENT	PROFIL
Abonament1	Profil1
Abonament1	Profil2
Abonament1	Profil3
Abonament2	Profil1
Abonament2	Profil4
Abonament3	Profil5

FN2

O relație se află în FN2 dacă și numai dacă aceasta este deja în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară.

Pentru a ajunge în FN2 am rezolvat toate relațiile many-to-many cu ajutorul tabelor asociative.

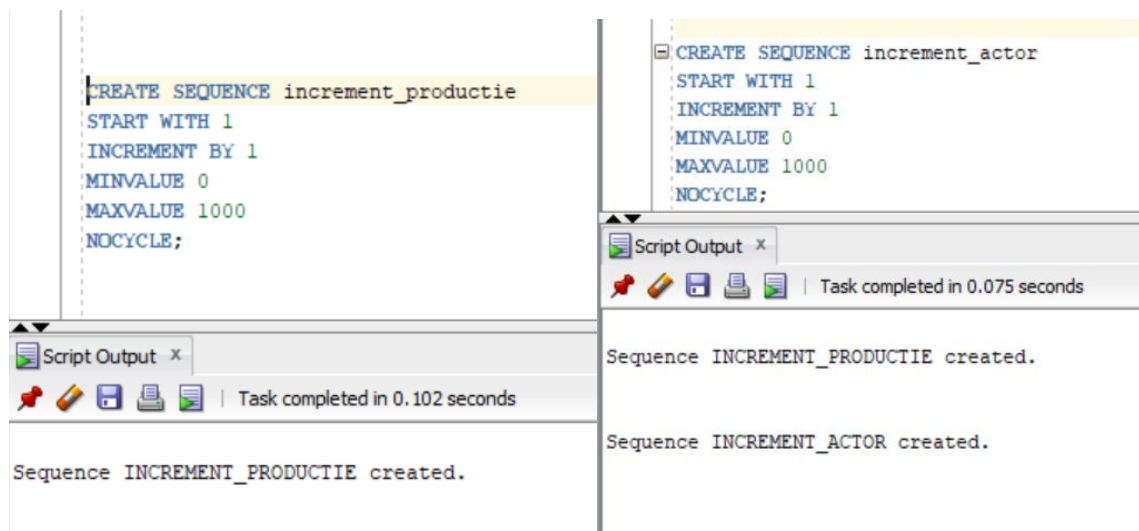
Între tabela ROL, ACTOR și PRODUCȚIE am adăugat ROL_PRODUCTIE_JUCAT care are cheile străine celor două și transformă relația many-to-many în one-to-many, analog pentru PRODUCȚIE și ABONAMENT, PROFIL și PRODUCȚIE, PRODUCȚIE și DIRECTOR.

FN3

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

Dacă initial timp_ecran se afla în tabela ACTOR, aceasta a fost mutată în tabela asociativă ROL_PRODUCTIE_JUCAT deoarece acest atribut depinde și de actor_id dar și de productie_id

Exercițiul 10: Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).



Exercițiul 11: Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative; maxim 30 de înregistrări în fiecare tabel).

PLATA

```
CREATE TABLE PLATA(
```

```
    plata_id NUMBER(6) PRIMARY KEY,
```

```
    detinator VARCHAR2(50) NOT NULL,
```

```
    cod NUMBER(16) NOT NULL UNIQUE,
```

```
    data_expirare DATE NOT NULL,
```

```
    CVV NUMBER(3) NOT NULL
```

```
);
```

```
INSERT INTO PLATA VALUES(1, 'JOHN SMITH', 1234567890123456,  
to_date('15.1.2024','dd.mm.yyyy'), 123);
```

```
INSERT INTO PLATA VALUES(2, 'ALICE JOHNSON', 2345678901234567,  
to_date('10.10.2025','dd.mm.yyyy'), 173);
```

```
INSERT INTO PLATA VALUES(3, 'MICHAEL BROWN', 3456789012345678,
to_date('20.12.2024','dd.mm.yyyy'), 333);
```

```
INSERT INTO PLATA VALUES(4, 'EMILY DAVIS', 4567890123456789,
to_date('14.7.2025','dd.mm.yyyy'), 909);
```

```
INSERT INTO PLATA VALUES(5, 'DAVID WILSON', 4254753564907767,
to_date('5.3.2025','dd.mm.yyyy'), 472);
```

```
INSERT INTO PLATA VALUES(6, 'SALAH LEE', 7345113144437373,
to_date('30.8.2024','dd.mm.yyyy'), 649);
```

```
CREATE TABLE PLATA(
    plata_id NUMBER(6) PRIMARY KEY,
    detinator VARCHAR2(50) NOT NULL,
    cod NUMBER(16) NOT NULL UNIQUE,
    data_expirare DATE NOT NULL,
    CVV NUMBER(3) NOT NULL
);
INSERT INTO PLATA VALUES(1, 'JOHN SMITH', 1234567890123456, to_date('15.1.2024','dd.mm.yyyy'), 123);
INSERT INTO PLATA VALUES(2, 'ALICE JOHNSON', 2345678901234567, to_date('10.10.2025','dd.mm.yyyy'), 173);
INSERT INTO PLATA VALUES(3, 'MICHAEL BROWN', 3456789012345678, to_date('20.12.2024','dd.mm.yyyy'), 333);
INSERT INTO PLATA VALUES(4, 'EMILY DAVIS', 4567890123456789, to_date('14.7.2025','dd.mm.yyyy'), 909);
INSERT INTO PLATA VALUES(5, 'DAVID WILSON', 4254753564907767, to_date('5.3.2025','dd.mm.yyyy'), 472);
INSERT INTO PLATA VALUES(6, 'SALAH LEE', 7345113144437373, to_date('30.8.2024','dd.mm.yyyy'), 649);
```

PLATA_ID	DETINATOR	COD	DATA_EXPIRARE	CVV
1	1 JOHN SMITH	1234567890123456	15-JAN-24	123
2	2 ALICE JOHNSON	2345678901234567	10-OCT-25	173
3	3 MICHAEL BROWN	3456789012345678	20-DEC-24	333
4	4 EMILY DAVIS	4567890123456789	14-JUL-25	909
5	5 DAVID WILSON	4254753564907767	05-MAR-25	472
6	6 SALAH LEE	7345113144437373	30-AUG-24	649

ABONAMENT

```
CREATE TABLE ABONAMENT(
```

```
    abonament_id NUMBER(6) PRIMARY KEY,
```

```
    tip_plata VARCHAR(10) NOT NULL,
```

```
    data_expirare_plata DATE NOT NULL
```

```
);
```

```
INSERT INTO ABONAMENT VALUES(10, 'lunar', to_date('13.2.2024', 'dd.mm.yyyy'));
```

```
INSERT INTO ABONAMENT VALUES(20, 'lunar', to_date('17.6.2024', 'dd.mm.yyyy'));
```

```
INSERT INTO ABONAMENT VALUES(30, 'anual', to_date('25.12.2025', 'dd.mm.yyyy'));
```

```
INSERT INTO ABONAMENT VALUES(40, 'anual', to_date('13.1.2025', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(50, 'lunar', to_date('25.6.2025', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(60, 'anual', to_date('13.2.2025', 'dd.mm.yyyy'));
```

```
CREATE TABLE ABONAMENT(
    abonament_id NUMBER(6) PRIMARY KEY,
    tip_plata VARCHAR(10) NOT NULL,
    data_expirare_plata DATE NOT NULL
);
INSERT INTO ABONAMENT VALUES(10, 'lunar', to_date('13.2.2024', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(20, 'lunar', to_date('17.6.2024', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(30, 'anual', to_date('25.12.2025', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(40, 'anual', to_date('13.1.2025', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(50, 'lunar', to_date('25.6.2025', 'dd.mm.yyyy'));
INSERT INTO ABONAMENT VALUES(60, 'anual', to_date('13.2.2025', 'dd.mm.yyyy'));
```

	ABONAMENT_ID	TIP_PLATA	DATA_EXPIRARE_PLATA
1	10	lunar	13-FEB-24
2	20	lunar	17-JUN-24
3	30	anual	25-DEC-25
4	40	anual	13-JAN-25
5	50	lunar	25-JUN-25
6	60	anual	13-FEB-25

PROFIL

```
CREATE TABLE PROFIL(
    profil_id NUMBER(6) PRIMARY KEY,
    abonament_id NUMBER(6) NOT NULL,
    porecla VARCHAR2(20) NOT NULL,
    data_creat DATE NOT NULL,
    FOREIGN KEY (abonament_id) REFERENCES ABONAMENT(abonament_id)
);
INSERT INTO PROFIL VALUES(11, 10, 'Spark', to_date('13.1.2023', 'dd.mm.yyyy'));
INSERT INTO PROFIL VALUES(22, 10, 'Luna', to_date('14.2.2024', 'dd.mm.yyyy'));
INSERT INTO PROFIL VALUES(33, 20, 'Rocket', to_date('16.1.2022', 'dd.mm.yyyy'));
INSERT INTO PROFIL VALUES(44, 30, 'Maverick', to_date('5.2.2024', 'dd.mm.yyyy'));
INSERT INTO PROFIL VALUES(55, 40, 'Raven', to_date('14.1.2023', 'dd.mm.yyyy'));
```


INSERT INTO PROFIL VALUES(66, 60, 'Scout', to_date('4.1.2024', 'dd.mm.yyyy'));

INSERT INTO PROFIL VALUES(77, 50, 'Scout', to_date('4.1.2024', 'dd.mm.yyyy'));

```
CREATE TABLE PROFIL(  
    profil_id NUMBER(6) PRIMARY KEY,  
    abonament_id NUMBER(6) NOT NULL,  
    porecla VARCHAR2(20) NOT NULL,  
    data_creat DATE NOT NULL,  
    FOREIGN KEY (abonament_id) REFERENCES ABONAMENT(abonament_id)  
);  
INSERT INTO PROFIL VALUES(11, 10, 'Spark', to_date('13.1.2023', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(22, 10, 'Luna', to_date('14.2.2024', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(33, 20, 'Rocket', to_date('16.1.2022', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(44, 30, 'Maverick', to_date('5.2.2024', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(55, 40, 'Raven', to_date('14.1.2023', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(66, 60, 'Scout', to_date('4.1.2024', 'dd.mm.yyyy'));  
INSERT INTO PROFIL VALUES(77, 50, 'Scout', to_date('4.1.2024', 'dd.mm.yyyy'));
```

	PROFIL_ID	ABONAMENT_ID	PORECLA	DATA_CREARE
1	11	10	Spark	13-JAN-23
2	22	10	Luna	14-FEB-24
3	33	20	Rocket	16-JAN-22
4	44	30	Maverick	05-FEB-24
5	55	40	Raven	14-JAN-23
6	66	60	Scout	04-JAN-24
7	77	50	Scout	04-JAN-24

UTILIZATOR

CREATE TABLE UTILIZATOR(
 utilizator_id NUMBER(6) PRIMARY KEY,
 abonament_id NUMBER(6) NOT NULL,
 plata_id NUMBER(6) NOT NULL,
 mail VARCHAR2(50) NOT NULL UNIQUE,
 parola VARCHAR(50) NOT NULL,
 data_creat DATE NOT NULL,
 telefon VARCHAR2(16),
 FOREIGN KEY (abonament_id) REFERENCES ABONAMENT(abonament_id),
 FOREIGN KEY (plata_id) REFERENCES PLATA(plata_id)
);

```
INSERT INTO UTILIZATOR VALUES(100, 10, 1, 'johnsmith@gmail.com', 'josmith123',
to_date('13.01.2023', 'dd.mm.yyyy'), 0762599250);
```

```
INSERT INTO UTILIZATOR VALUES(101, 20, 2, 'aljohnson@gmail.com', 'johnsonAls',
to_date('16.01.2024', 'dd.mm.yyyy'), 0742911211);
```

```
INSERT INTO UTILIZATOR VALUES(102, 30, 3, 'brownMICH@gmail.com', 'safePassword',
to_date('5.02.2024', 'dd.mm.yyyy'), 0798566123);
```

```
INSERT INTO UTILIZATOR VALUES(103, 40, 4, 'DaViSeml@gmail.com', 'EmilY',
to_date('14.01.2023', 'dd.mm.yyyy'), 0776544232);
```

```
INSERT INTO UTILIZATOR VALUES(104, 50, 5, 'willdavidson@gmail.com', 'q34trs67',
to_date('4.01.2024', 'dd.mm.yyyy'), 0768777565);
```

```
INSERT INTO UTILIZATOR(utilizator_id, abonament_id, plata_id, mail, parola, data_creat)
VALUES(105, 60, 6, 'LeeSrh@gmail.com', 'llovve56', to_date('4.01.2024', 'dd.mm.yyyy'));
```

```
CREATE TABLE UTILIZATOR(
    utilizator_id NUMBER(6) PRIMARY KEY,
    abonament_id NUMBER(6) NOT NULL,
    plata_id NUMBER(6) NOT NULL,
    mail VARCHAR2(50) NOT NULL UNIQUE,
    parola VARCHAR(50) NOT NULL,
    data_creat DATE NOT NULL,
    telefon VARCHAR2(16),
    FOREIGN KEY (abonament_id) REFERENCES ABONAMENT(abonament_id),
    FOREIGN KEY (plata_id) REFERENCES PLATA(plata_id)
);
INSERT INTO UTILIZATOR VALUES(100, 10, 1, 'johnsmith@gmail.com', 'josmith123', to_date('13.01.2023', 'dd.mm.yyyy'), 0762599250);
INSERT INTO UTILIZATOR VALUES(101, 20, 2, 'aljohnson@gmail.com', 'johnsonAls', to_date('16.01.2024', 'dd.mm.yyyy'), 0742911211);
INSERT INTO UTILIZATOR VALUES(102, 30, 3, 'brownMICH@gmail.com', 'safePassword', to_date('5.02.2024', 'dd.mm.yyyy'), 0798566123);
INSERT INTO UTILIZATOR VALUES(103, 40, 4, 'DaViSeml@gmail.com', 'EmilY', to_date('14.01.2023', 'dd.mm.yyyy'), 0776544232);
INSERT INTO UTILIZATOR VALUES(104, 50, 5, 'willdavidson@gmail.com', 'q34trs67', to_date('4.01.2024', 'dd.mm.yyyy'), 0768777565);
INSERT INTO UTILIZATOR(utilizator_id, abonament_id, plata_id, mail, parola, data_creat)
VALUES(105, 60, 6, 'LeeSrh@gmail.com', 'llovve56', to_date('4.01.2024', 'dd.mm.yyyy'));
```

SQL All Rows Fetched: 6 in 0.004 seconds							
	UTILIZATOR_ID	ABONAMENT_ID	PLATA_ID	MAIL	PAROLA	DATA_CREARE	TELEFON
1	100	10	1	johnsmith@gmail.com	josmith123	13-JAN-23	762599250
2	101	20	2	aljohnson@gmail.com	johnsonAls	16-JAN-24	742911211
3	102	30	3	brownMICH@gmail.com	safePassword	05-FEB-24	798566123
4	103	40	4	DaViSeml@gmail.com	EmilY	14-JAN-23	776544232
5	104	50	5	willdavidson@gmail.com	q34trs67	04-JAN-24	768777565
6	105	60	6	LeeSrh@gmail.com	llovve56	04-JAN-24	(null)

DIRECTOR

```
CREATE TABLE DIRECTOR(
    director_id NUMBER(5) PRIMARY KEY,
    nume VARCHAR2(50),
    prenume VARCHAR2(50),
```

data_nastere DATE

);

INSERT INTO DIRECTOR VALUES(101, 'Hart', 'Hanson', to_date('26.7.1957', 'dd.mm.yyyy'));

INSERT INTO DIRECTOR VALUES(202, 'Steven', 'Knight', to_date('1.4.1959', 'dd.mm.yyyy'));

INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(303, 'Adam', 'Horowitz');

INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(404, 'Paul', 'Scheuring');

INSERT INTO DIRECTOR VALUES(505, 'Dana', 'Terrace', to_date('8.12.1990', 'dd.mm.yyyy'));

INSERT INTO DIRECTOR VALUES(606, 'Kenneth', 'Branagh', to_date('10.12.1960', 'dd.mm.yyyy'));

INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(707, 'Mark', 'Mylod');

INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(808, 'Yargos ', 'Lanthimos');

INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(909, 'Edward', 'Kitsis');

```
CREATE TABLE DIRECTOR(  
    director_id NUMBER(5) PRIMARY KEY,  
    nume VARCHAR2(50),  
    prenume VARCHAR2(50),  
    data_nastere DATE  
);  
INSERT INTO DIRECTOR VALUES(101, 'Hart', 'Hanson', to_date('26.7.1957', 'dd.mm.yyyy'));  
INSERT INTO DIRECTOR VALUES(202, 'Steven', 'Knight', to_date('1.4.1959', 'dd.mm.yyyy'));  
INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(303, 'Adam', 'Horowitz');  
INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(404, 'Paul', 'Scheuring');  
INSERT INTO DIRECTOR VALUES(505, 'Dana', 'Terrace', to_date('8.12.1990', 'dd.mm.yyyy'));  
INSERT INTO DIRECTOR VALUES(606, 'Kenneth', 'Branagh', to_date('10.12.1960', 'dd.mm.yyyy'));  
INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(707, 'Mark', 'Mylod');  
INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(808, 'Yargos ', 'Lanthimos');  
INSERT INTO DIRECTOR (director_id, nume, prenume) VALUES(909, 'Edward', 'Kitsis');
```

SQL | All Rows Fetched: 9 in 0.003 seconds

	DIRECTOR_ID	NUME	PRENUME	DATA_NASTERE
1	101	Hart	Hanson	26-JUL-57
2	202	Steven	Knight	01-APR-59
3	505	Dana	Terrace	08-DEC-90
4	303	Adam	Horowitz	(null)
5	404	Paul	Scheuring	(null)
6	808	Yargos	Lanthimos	(null)
7	707	Mark	Mylod	(null)
8	606	Kenneth	Branagh	10-DEC-60
9	909	Edward	Kitsis	(null)

ACTOR

CREATE TABLE ACTOR(

actor_id NUMBER(5) PRIMARY KEY,

nome VARCHAR2(50) NOT NULL,

prename VARCHAR2(50) NOT NULL,

data_natare DATE

);

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Cillian', 'Murphy',
to_date('25.5.1976', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'David', 'Boreanaz',
to_date('16.5.1969', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Lana', 'Parrilla', to_date('15.7.1977',
'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Josh', 'Dallas', to_date('18.12.1978',
'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Dominic', 'Purcell',
to_date('17.2.1970', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Robles', 'Sarah-Nicole',
to_date('20.12.1991', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Kenneth', 'Branagh',
to_date('10.12.1960', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Taylor-Joy', 'Any',
to_date('16.4.1996', 'dd.mm.yyyy'));

INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Emma', 'Stone', to_date('6.11.1988',
'dd.mm.yyyy'));

```

CREATE TABLE ACTOR(
  actor_id NUMBER(5) PRIMARY KEY,
  nume VARCHAR2(50) NOT NULL,
  prenume VARCHAR2(50) NOT NULL,
  data_nater DATE
);
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Cillian', 'Murphy', to_date('25.5.1976', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'David', 'Boreanaz', to_date('16.5.1969', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Lana', 'Parrilla', to_date('15.7.1977', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Josh', 'Dallas', to_date('18.12.1978', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Dominic', 'Purcell', to_date('17.2.1970', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Robles', 'Sarah-Nicole', to_date('20.12.1991', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Kenneth', 'Branagh', to_date('10.12.1960', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Taylor-Joy', 'Any', to_date('16.4.1996', 'dd.mm.yyyy'));
INSERT INTO ACTOR VALUES(increment_actor.nextval, 'Emma', 'Stone', to_date('6.11.1988', 'dd.mm.yyyy'));

```

SQL | All Rows Fetched: 9 in 0.006 seconds

	ACTOR_ID	NUME	PRENUME	DATA_NATERE
1	1	Cillian	Murphy	25-MAY-76
2	2	David	Boreanaz	16-MAY-69
3	3	Lana	Parrilla	15-JUL-77
4	4	Josh	Dallas	18-DEC-78
5	5	Dominic	Purcell	17-FEB-70
6	6	Robles	Sarah-Nicole	20-DEC-91
7	7	Kenneth	Branagh	10-DEC-60
8	8	Taylor-Joy	Any	16-APR-96
9	9	Emma	Stone	06-NOV-88

ROL

CREATE TABLE ROL(

rol_id NUMBER(5) PRIMARY KEY,

nume_caracter VARCHAR2(50) NOT NULL,

importanta VARCHAR2(50) NOT NULL,

descriere VARCHAR2(200)

);

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(1, 'Thomas Shelby', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(2, 'Seeley Booth', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(3, 'Evil Queen', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(4, 'Prince Charming', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(5, 'Lincoln Burrows', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(6, 'Luz Noceda', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(7, 'Hercule Poirot', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(8, 'Margot', 'principal');

INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(9, 'Bella Baxter', 'principal');

```
CREATE TABLE ROL(  
    rol_id NUMBER(5) PRIMARY KEY,  
    nume_caracter VARCHAR2(50) NOT NULL,  
    importanta VARCHAR2(50) NOT NULL,  
    descriere VARCHAR2(200)  
);  
  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(1, 'Thomas Shelby', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(2, 'Seeley Booth', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(3, 'Evil Queen', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(4, 'Prince Charming', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(5, 'Lincoln Burrows', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(6, 'Luz Noceda', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(7, 'Hercule Poirot', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(8, 'Margot', 'principal');  
INSERT INTO ROL(rol_id, nume_caracter, importanta) VALUES(9, 'Bella Baxter', 'principal');
```

Script Output: SQL | All Rows Fetched: 9 in 0.008 seconds

ROL_ID	NUME_CARACTER	IMPORTANTA	DESCRIERE
1	1 Thomas Shelby	principal	(null)
2	2 Seeley Booth	principal	(null)
3	3 Evil Queen	principal	(null)
4	4 Prince Charming	principal	(null)
5	5 Lincoln Burrows	principal	(null)
6	6 Luz Noceda	principal	(null)
7	7 Hercule Poirot	principal	(null)
8	8 Margot	principal	(null)
9	9 Bella Baxter	principal	(null)

PRODUCTIE

CREATE TABLE PRODUCTIE(

productie_id NUMBER(6) PRIMARY KEY,

denumire VARCHAR2(50) NOT NULL,

gen VARCHAR2(50) NOT NULL,

```

data_aparitie DATE NOT NULL,

director_id NUMBER(5) NOT NULL,

FOREIGN KEY (director_id) REFERENCES DIRECTOR(director_id)

);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Peaky Blinders', 'drama',
to_date('12.12.2013', 'dd.mm.yyyy'), 202);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Death On The Nile',
'politist', to_date('11.2.2022', 'dd.mm.yyyy'), 606);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Bones', 'drama',
to_date('13.12.2005', 'dd.mm.yyyy'), 101);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'The Menu', 'comedie',
to_date('18.11.2022', 'dd.mm.yyyy'), 707);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Prison Break', 'drama',
to_date('29.8.2005', 'dd.mm.yyyy'), 404);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Owl House', 'aventura',
to_date('10.1.2020', 'dd.mm.yyyy'), 505);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'A Haunting In Venice',
'politist', to_date('15.9.2023', 'dd.mm.yyyy'), 606);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Once Upon A Time',
'drama', to_date('23.10.2011', 'dd.mm.yyyy'), 303);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Poor Things', 'comedie',
to_date('8.12.2023', 'dd.mm.yyyy'), 808);

```

```

CREATE TABLE PRODUCTIE (
    productie_id NUMBER(6) PRIMARY KEY,
    denumire VARCHAR2(50) NOT NULL,
    gen VARCHAR2(50) NOT NULL,
    data_aparitie DATE NOT NULL,
    director_id NUMBER(5) NOT NULL,
    FOREIGN KEY (director_id) REFERENCES DIRECTOR(director_id)
);

INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Peaky Blinders', 'drama', to_date('12.12.2013', 'dd.mm.yyyy'), 202);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Death On The Nile', 'politist', to_date('11.2.2022', 'dd.mm.yyyy'), 606);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Bones', 'drama', to_date('13.12.2005', 'dd.mm.yyyy'), 101);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'The Menu', 'comedie', to_date('18.11.2022', 'dd.mm.yyyy'), 707);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Prison Break', 'drama', to_date('29.8.2005', 'dd.mm.yyyy'), 404);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Owl House', 'aventura', to_date('10.1.2020', 'dd.mm.yyyy'), 505);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'A Haunting In Venice', 'politist', to_date('15.9.2023', 'dd.mm.yyyy'), 606);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Once Upon A Time', 'drama', to_date('23.10.2011', 'dd.mm.yyyy'), 303);
INSERT INTO PRODUCTIE VALUES(increment_productie.nextval, 'Poor Things', 'comedie', to_date('8.12.2023', 'dd.mm.yyyy'), 808);

```

SQL All Rows Fetched: 9 in 0.01 seconds					
	PRODUCTIE_ID	DENUMIRE	GEN	DATA_APARITIE	DIRECTOR_ID
1	1	Peaky Blinders	drama	12-DEC-13	202
2	2	Death On The Nile	politist	11-FEB-22	606
3	3	Bones	drama	13-DEC-05	101
4	4	The Menu	comedie	18-NOV-22	707
5	5	Prison Break	drama	29-AUG-05	404
6	6	Owl House	aventura	10-JAN-20	505
7	7	A Haunting In Venice	politist	15-SEP-23	606
8	8	Once Upon A Time	drama	23-OCT-11	303
9	9	Poor Things	comedie	08-DEC-23	808

FILM

CREATE TABLE FILM(

 productie_id NUMBER(6) PRIMARY KEY,

 durata NUMBER(3) NOT NULL,

 FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)

);

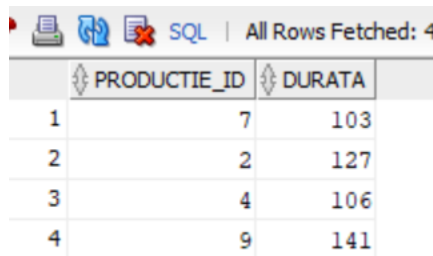
INSERT INTO FILM VALUES(7, 103);

INSERT INTO FILM VALUES(2, 127);

INSERT INTO FILM VALUES(4, 106);

INSERT INTO FILM VALUES(9, 141);

```
CREATE TABLE FILM(
    productie_id NUMBER(6) PRIMARY KEY,
    durata NUMBER(3) NOT NULL,
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);
INSERT INTO FILM VALUES(7, 103);
INSERT INTO FILM VALUES(2, 127);
INSERT INTO FILM VALUES(4, 106);
INSERT INTO FILM VALUES(9, 141);
```

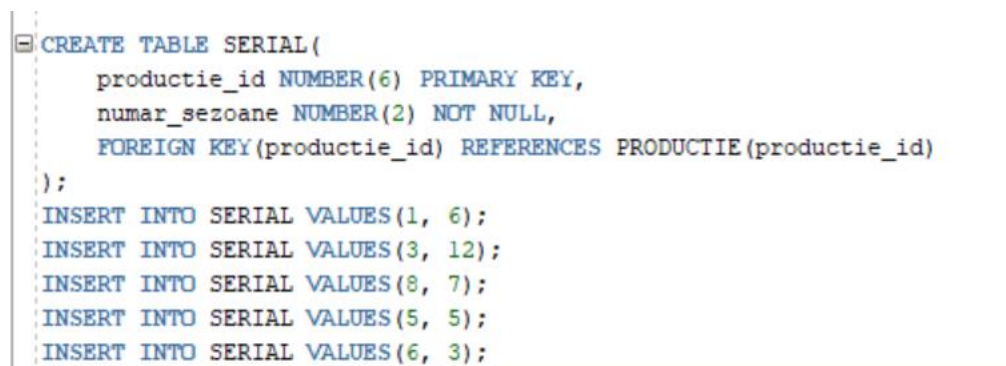



	PRODUCTIE_ID	DURATA
1	7	103
2	2	127
3	4	106
4	9	141

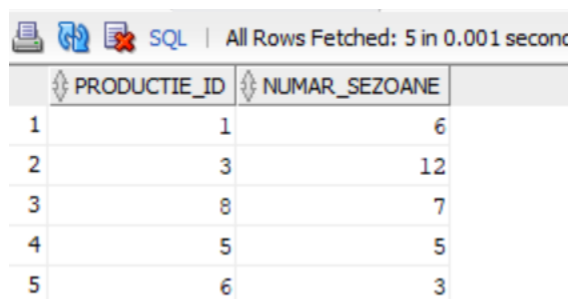
SERIAL

```
CREATE TABLE SERIAL(
    productie_id NUMBER(6) PRIMARY KEY,
    numar_sezoane NUMBER(2) NOT NULL,
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);
```

```
INSERT INTO SERIAL VALUES(1, 6);
INSERT INTO SERIAL VALUES(3, 12);
INSERT INTO SERIAL VALUES(8, 7);
INSERT INTO SERIAL VALUES(5, 5);
INSERT INTO SERIAL VALUES(6, 3);
```



```
CREATE TABLE SERIAL(
    productie_id NUMBER(6) PRIMARY KEY,
    numar_sezoane NUMBER(2) NOT NULL,
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);
INSERT INTO SERIAL VALUES(1, 6);
INSERT INTO SERIAL VALUES(3, 12);
INSERT INTO SERIAL VALUES(8, 7);
INSERT INTO SERIAL VALUES(5, 5);
INSERT INTO SERIAL VALUES(6, 3);
```



	PRODUCTIE_ID	NUMAR_SEZOANE
1	1	6
2	3	12
3	8	7
4	5	5
5	6	3

EPISOD

```
CREATE TABLE EPISOD(  
    episod_id NUMBER(6) PRIMARY KEY,  
    productie_id NUMBER(6) NOT NULL,  
    numar_episod NUMBER(2) NOT NULL,  
    numar_sezon NUMBER(2) NOT NULL,  
    titlu VARCHAR2(50) NOT NULL,  
    durata NUMBER(3) NOT NULL,  
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)  
);  
  
INSERT INTO EPISOD VALUES(1, 1, 1, 1, 'Episod 1', 58);  
INSERT INTO EPISOD VALUES(2, 1, 2, 1, 'Episod 2', 58);  
INSERT INTO EPISOD VALUES(3, 3, 5, 7, 'The Twist int the Twister', 42);  
INSERT INTO EPISOD VALUES(4, 3, 6, 7, 'The Crack In The Code', 45);  
INSERT INTO EPISOD VALUES(5, 3, 11, 12, 'The Day in the Life', 46);  
INSERT INTO EPISOD VALUES(6, 8, 1, 1, 'Pilot', 42);  
INSERT INTO EPISOD VALUES(7, 8, 2, 1, 'The Thing You Love Most', 41);  
INSERT INTO EPISOD VALUES(8, 8, 3, 1, 'Snow Falls', 42);  
INSERT INTO EPISOD VALUES(9, 8, 4, 1, 'The Price Of Gold', 45);  
INSERT INTO EPISOD VALUES(10, 5, 1, 1, 'Pilot', 42);  
INSERT INTO EPISOD VALUES(11, 5, 2, 1, 'Allen', 44);  
INSERT INTO EPISOD VALUES(12, 5, 3, 1, 'Cell Test', 43);
```

```
CREATE TABLE EPISOD(  
    episod_id NUMBER(6) PRIMARY KEY,  
    productie_id NUMBER(6) NOT NULL,  
    numar_episod NUMBER(2) NOT NULL,  
    numar_sezon NUMBER(2) NOT NULL,  
    titlu VARCHAR2(50) NOT NULL,  
    durata NUMBER(3) NOT NULL,  
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)  
);
```

```

INSERT INTO EPISOD VALUES(1, 1, 1, 1, 'Episod 1', 58);
INSERT INTO EPISOD VALUES(2, 1, 2, 1, 'Episod 2', 58);
INSERT INTO EPISOD VALUES(3, 3, 5, 7, 'The Twist int the Twister', 42);
INSERT INTO EPISOD VALUES(4, 3, 6, 7, 'The Crack In The Code', 45);
INSERT INTO EPISOD VALUES(5, 3, 11, 12, 'The Day in the Life', 46);
INSERT INTO EPISOD VALUES(6, 8, 1, 1, 'Pilot', 42);
INSERT INTO EPISOD VALUES(7, 8, 2, 1, 'The Thing You Love Most', 41);
INSERT INTO EPISOD VALUES(8, 8, 3, 1, 'Snow Falls', 42);
INSERT INTO EPISOD VALUES(9, 8, 4, 1, 'The Price Of Gold', 45);
INSERT INTO EPISOD VALUES(10, 5, 1, 1, 'Pilot', 42);
INSERT INTO EPISOD VALUES(11, 5, 2, 1, 'Allen', 44);
INSERT INTO EPISOD VALUES(12, 5, 3, 1, 'Cell Test', 43);

```

SQL | All Rows fetched: 12 in 0.004 seconds

EPISOD_ID	PRODUCTIE_ID	NUMAR_EPISOD	NUMAR_SEZON	TITLU	DURATA
1	1	1	1	1 Episod 1	58
2	2	1	2	1 Episod 2	58
3	3	3	5	7 The Twist int the Twister	42
4	4	3	6	7 The Crack In The Code	45
5	5	3	11	12 The Day in the Life	46
6	6	8	1	1 Pilot	42
7	7	8	2	1 The Thing You Love Most	41
8	8	8	3	1 Snow Falls	42
9	9	8	4	1 The Price Of Gold	45
10	10	5	1	1 Pilot	42
11	11	5	2	1 Allen	44
12	12	5	3	1 Cell Test	43

PROFIL_PRODUCTIE

```

CREATE TABLE PROFIL_PRODUCTIE(
    profil_productie_id NUMBER(6) PRIMARY KEY,
    profil_id NUMBER(6) NOT NULL,
    productie_id NUMBER(6) NOT NULL,
    FOREIGN KEY(profil_id) REFERENCES PROFIL(profil_id),
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);

```

```

INSERT INTO PROFIL_PRODUCTIE VALUES(111, 11, 1);
INSERT INTO PROFIL_PRODUCTIE VALUES(112, 11, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(113, 11, 7);

```

```

INSERT INTO PROFIL_PRODUCTIE VALUES(114, 22, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(115, 33, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(116, 44, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(117, 44, 3);
INSERT INTO PROFIL_PRODUCTIE VALUES(118, 55, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(119, 66, 6);
INSERT INTO PROFIL_PRODUCTIE VALUES(120, 11, 6);
INSERT INTO PROFIL_PRODUCTIE VALUES(121, 55, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(122, 11, 4);
INSERT INTO PROFIL_PRODUCTIE VALUES(123, 11, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(124, 22, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(125, 22, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(126, 66, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(127, 66, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(128, 66, 4);

```

```

CREATE TABLE PROFIL_PRODUCTIE (
    profil_productie_id NUMBER(6) PRIMARY KEY,
    profil_id NUMBER(6) NOT NULL,
    productie_id NUMBER(6) NOT NULL,
    FOREIGN KEY(profil_id) REFERENCES PROFIL(profil_id),
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);

```

```

INSERT INTO PROFIL_PRODUCTIE VALUES(111, 11, 1);
INSERT INTO PROFIL_PRODUCTIE VALUES(112, 11, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(113, 11, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(114, 22, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(115, 33, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(116, 44, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(117, 44, 3);
INSERT INTO PROFIL_PRODUCTIE VALUES(118, 55, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(119, 66, 6);
INSERT INTO PROFIL_PRODUCTIE VALUES(120, 11, 6);
INSERT INTO PROFIL_PRODUCTIE VALUES(121, 55, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(122, 11, 4);
INSERT INTO PROFIL_PRODUCTIE VALUES(123, 11, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(124, 22, 9);
INSERT INTO PROFIL_PRODUCTIE VALUES(125, 22, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(126, 66, 7);
INSERT INTO PROFIL_PRODUCTIE VALUES(127, 66, 2);
INSERT INTO PROFIL_PRODUCTIE VALUES(128, 66, 4);

```

SQL | All Rows Fetched: 18 in 0.003 seconds

	PROFIL_PRODUCTIE_ID	PROFIL_ID	PRODUCTIE_ID
3	113	11	7
4	114	22	7
5	115	33	7
6	116	44	9
7	117	44	3
8	118	55	9
9	119	66	6
10	120	11	6
11	121	55	7
12	122	11	4
13	123	11	9
14	124	22	9
15	125	22	2
16	126	66	7
17	127	66	2
18	128	66	4

PRODUCTIE DIRECTOR

```

CREATE TABLE PRODUCTIE_DIRECTOR(
    productie_director_id NUMBER(6) PRIMARY KEY,

```

```

productie_id NUMBER(6) NOT NULL,
director_id NUMBER(5) NOT NULL,
FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id),
FOREIGN KEY(director_id) REFERENCES DIRECTOR(director_id)
);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(1, 1, 202);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(2, 2, 606);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(3, 3, 101);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(4, 4, 707);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(5, 5, 404);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(6, 6, 505);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(7, 7, 606);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(8, 8, 303);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(9, 9, 808);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(10, 8, 909);

```

```

CREATE TABLE PRODUCTIE_DIRECTOR(
    productie_director_id NUMBER(6) PRIMARY KEY,
    productie_id NUMBER(6) NOT NULL,
    director_id NUMBER(5) NOT NULL,
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id),
    FOREIGN KEY(director_id) REFERENCES DIRECTOR(director_id)
);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(1, 1, 202);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(2, 2, 606);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(3, 3, 101);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(4, 4, 707);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(5, 5, 404);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(6, 6, 505);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(7, 7, 606);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(8, 8, 303);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(9, 9, 808);
INSERT INTO PRODUCTIE_DIRECTOR VALUES(10, 8, 909);

```

script output | Query Result

SQL | All Rows Fetched: 10 in 0.003 seconds

	PRODUCTIE_DIRECTOR_ID	PRODUCTIE_ID	DIRECTOR_ID
1	1	1	202
2	2	2	606
3	3	3	101
4	4	4	707
5	5	5	404
6	6	6	505
7	7	7	606
8	8	8	303
9	9	9	808
10	10	8	909

ROL_PRODUCTIE_JUCAT

```
CREATE TABLE ROL_PRODUCTIE_JUCAT(
    rol_productie_jucat NUMBER(7) PRIMARY KEY,
    actor_id NUMBER(5) NOT NULL,
    rol_id NUMBER(5) NOT NULL,
    productie_id NUMBER(6) NOT NULL,
    timp_ecran NUMBER(3),
    FOREIGN KEY(actor_id) REFERENCES ACTOR(actor_id),
    FOREIGN KEY(rol_id) REFERENCES ROL(rol_id),
    FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(1, 1, 1, 1);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(2, 2, 2, 3);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(3, 3, 3, 8);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(4, 4, 4, 8);
```



```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(5, 5, 5, 5);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(6, 6, 6, 6, 359);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(7, 7, 7, 7, 87);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id)
VALUES(8, 7, 7, 2);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(9, 8, 8, 4, 74);
```

```
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(10, 9, 9, 9, 112);
```

```
CREATE TABLE ROL_PRODUCTIE_JUCAT(
  rol_productie_jucat NUMBER(7) PRIMARY KEY,
  actor_id NUMBER(5) NOT NULL,
  rol_id NUMBER(5) NOT NULL,
  productie_id NUMBER(6) NOT NULL,
  timp_ecran NUMBER(3),
  FOREIGN KEY(actor_id) REFERENCES ACTOR(actor_id),
  FOREIGN KEY(rol_id) REFERENCES ROL(rol_id),
  FOREIGN KEY(productie_id) REFERENCES PRODUCTIE(productie_id)
);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(1, 1, 1, 1);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(2, 2, 2, 3);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(3, 3, 3, 8);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(4, 4, 4, 8);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(5, 5, 5, 5);
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(6, 6, 6, 6, 359);
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(7, 7, 7, 7, 87);
INSERT INTO ROL_PRODUCTIE_JUCAT(rol_productie_jucat, actor_id, rol_id, productie_id) VALUES(8, 7, 7, 2);
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(9, 8, 8, 4, 74);
INSERT INTO ROL_PRODUCTIE_JUCAT VALUES(10, 9, 9, 9, 112);
```

Script Output | Query Results | All Rows Fetched: 10 in 0.003 seconds

	ROL_PRODUCTIE_JUCAT	ACTOR_ID	ROL_ID	PRODUCTIE_ID	TIMP_ECRAN
1	1	1	1	1	(null)
2	2	2	2	3	(null)
3	3	3	3	8	(null)
4	4	4	4	8	(null)
5	5	5	5	5	(null)
6	6	6	6	6	359
7	7	7	7	7	87
8	8	7	7	2	(null)
9	9	8	8	4	74
10	10	9	9	9	112

Exercițiul 12: Formulați în limbaj natural și implementați 5 cereri SQL complexe.

1. Ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

```
SELECT u.parola as "Email",
       DECODE(to_char(a.data_expirare_plata, 'yyyy'), 2024, 'Anul acesta', 2025, 'Anul
viitor', 'N/A') as "An expirare plata",
       NVL(u.telefon, u.mail) as "Contact"
FROM utilizator u LEFT JOIN abonament a
      ON a.abonament_id=u.abonament_id
ORDER BY u.mail;
```



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

```
SELECT u.parola as "Email",
       DECODE(to_char(a.data_expirare_plata, 'yyyy'), 2024, 'Anul acesta', 2025, 'Anul viitor', 'N/A') as "An expirare plata",
       NVL(u.telefon, u.mail) as "Contact"
FROM utilizator u LEFT JOIN abonament a
      ON a.abonament_id=u.abonament_id
ORDER BY u.mail;
```

The results window shows 6 rows of data:

	Email	An expirare plata	Contact
1	Emily	Anul viitor	776544232
2	llovve56	Anul viitor	LeeSrh@gmail.com
3	johnsonAls	Anul acesta	742911211
4	safePassword	Anul viitor	798566123
5	josmith123	Anul acesta	762599250
6	q34trs67	Anul viitor	768777565

2. Grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING) în care intervin cel puțin 3 tabele (în cadrul aceleiași cereri)

```
select d.director_id,
       concat(concat(d.num, ' '), d.prenume) as "Nume director",
       count(p.productie_id) as "Numar productii"
from director d join productie_director pd
      on d.director_id=pd.director_id
      join productie p
      on p.productie_id=pd.productie_id
where p.data_aparitie>=add_months(sysdate, -60)
group by d.director_id, d.num, d.prenume
having count(p.productie_id)>1
      and (select count(*)
            from productie_director pd2
            where pd2.director_id=d.director_id)>1
```

order by "Numar productii" desc;

```
select d.director_id,  
       concat(concat(d.num, ' '), d.prenume) as "Nume director",  
       count(p.productie_id) as "Numar productii"  
from director d join productie_director pd  
  on d.director_id=pd.director_id  
 join productie p  
  on p.productie_id=pd.productie_id  
where p.data_aparitiei >= add_months(sysdate, -60)  
group by d.director_id, d.num, d.prenume  
having count(p.productie_id) >= 1  
       and (select count(*)  
            from productie_director pd2  
            where pd2.director_id=d.director_id) >= 1  
order by "Numar productii" desc;
```

SQL | All Rows Fetched: 4 in 0.011 seconds

DIRECTOR_ID	Nume director	Numar productii
1	606 Kenneth Branagh	2
2	707 Mark Mylod	1
3	808 Yorgos Lanthimos	1
4	505 Dana Terrace	1


3. O functie cu sir de caractere, o functie cu date calendaristice +subcereri sincronizare pe 3 tabele

```
select  
  concat(concat(d.num, ' '), d.prenume) as "Nume actor",  
  p.denumire as "Denumire",  
  (select sysdate - p.data_aparitiei from dual) as "Date"  
from director d  
join productie_director pd on d.director_id = pd.director_id  
join productie p on p.productie_id = pd.productie_id  
order by "Date";
```

```
--
select
    concat(concat(d.num, ' '), d.prenume) as "Nume actor",
    p.denumire as "Denumire",
    (select sysdate - p.data_aparitie from dual) as "Date"
from director d
join productie_director pd on d.director_id = pd.director_id
join productie p on p.productie_id = pd.productie_id
order by "Date";
```

Script Output x

Query Result x

 | All Rows Fetched: 10 in 0.017 seconds

	Nume actor	Denumire	Date
1	Yargos Lanthimos	Poor Things	175.576990740740740740740740740741
2	Kenneth Branagh	A Haunting In Venice	259.576990740740740740740740740741
3	Mark Mylod	The Menu	560.576990740740740740740740740741
4	Kenneth Branagh	Death On The Nile	840.576990740740740740740740740741
5	Dana Terrace	Owl House	1603.576990740740740740740740740741
6	Steven Knight	Peaky Blinders	3823.576990740740740740740740740741
7	Edward Kitsis	Once Upon A Time	4604.576990740740740740740740740741
8	Adam Horowitz	Once Upon A Time	4604.576990740740740740740740740741
9	Hart Hanson	Bones	6744.576990740740740740740740740741
10	Paul Scheuring	Prison Break	6850.576990740740740740740740740741

4. 1 bloc de cerere WITH +CASE

```
with doar_filme as (
select p.productie_id, p.denumire, p.data_aparitie, p.director_id, f.durata
from productie p inner join film f
on p.productie_id=f.productie_id)
```

```
select df.denumire as "Denumire",
    d.num||' '||d.prenume as "Director",
    case
        when df.durata<110 then 'Short'
        when df.durata between 110 and 130 then 'Medium'
        else 'Long'
    end as "Durata"
from doar_filme df join director d
on df.director_id=d.director_id;
```

```

with doar_filme as (
select p.productie_id, p.denumire, p.data_aparitiei, p.director_id, f.durata
from productie p inner join film f
on p.productie_id=f.productie_id)

select df.denumire as "Denumire",
       d.nume||' '||d.prenume as "Director",
       case
         when df.durata<110 then 'Short'
         when df.durata between 110 and 130 then 'Medium'
         else 'Long'
       end as "Durata"
from doar_filme df join director d
on df.director_id=d.director_id;

```

SQL | All Rows Fetched: 4 in 0.063 seconds

	Denumire	Director	Durata
1	Poor Things	Yargos Lanthimos	Long
2	The Menu	Mark Mylod	Short
3	Death On The Nile	Kenneth Branagh	Medium
4	A Haunting In Venice	Kenneth Branagh	Short

5. O functie pe siruri de caractere + o functie pe date calendaristice+o cerere nesincronizata in FROM

```

with toate_productiile as(
select p.productie_id, p.denumire, p.gen, p.data_aparitiei, f.durata, s.numar_sezoane
from productie p left join film f
on p.productie_id=f.productie_id
left join serial s
on s.productie_id=p.productie_id)

select t.productie_id as "Id",
       lower(t.denumire) as "Titlu",
       t.gen as "Gen",
       t.durata as "Durata",
       t.numar_sezoane as "Numar sezoane",
       extract(year from t.data_aparitiei) as "An aparitiei",
       pp.nr_profile
from toate_productiile t
join (select productie_id,
       count(*) as nr_profile

```

```
--2.0 functie pe data calendaristica+o functie pe sir de caracter
with toate_productiile as(
    select p.productie_id, p.denumire, p.gen, p.data_aparitie, f.durata, s.numar_sezoane
    from productie p left join film f
        on p.productie_id=f.productie_id
    left join serial s
        on s.productie_id=p.productie_id)

select t.productie_id as "Id",
    lower(t.denumire) as "Titlu",
    t.gen as "Gen",
    t.durata as "Durata",
    t.numar_sezoane as "Numar sezoane",
    extract(year from t.data_aparitie) as "An aparitie",
    pp.nr_profile
from toate_productiile t
join (select productie_id,
    count(*) as nr_profile
    from profil_productie
    group by productie_id) pp
on t.productie_id=pp.productie_id
order by t.gen;
```

SQL | All Rows Fetched: 7 in 0.042 seconds

	Id	Titlu	Gen	Durata	Numar sezoane	An aparitie	NR_PROFILE
1	6	owl house	aventura	(null)	3	2020	2
2	4	the menu	comedie	106	(null)	2022	2
3	9	poor things	comedie	141	(null)	2023	4
4	1	peaky blinders	drama	(null)	6	2013	1
5	3	bones	drama	(null)	12	2005	1
6	7	a haunting in venice	politist	103	(null)	2023	5
7	2	death on the nile	politist	127	(null)	2022	3

- Ștergerea episoadelor cele mai lungi
delete from episod
where durata=(select max(durata) from episod);

```
--stergerea directotului cel mai batran
delete from episod
where durata=(select max(durata) from episod);
```

Script Output x Query Result x

Task completed in 0.035 seconds

*Action:

2 rows deleted.

SQL | All Rows Fetched: 10 in 0.001 seconds

	EPISOD_ID	PRODUCTIE_ID	NUMAR_EPISOD	NUMAR_SEZON	TITLU	DURATA
1	3	3	5	7	The Twist int the Twister	42
2	4	3	6	7	The Crack In The Code	45
3	5	3	11	12	The Day in the Life	46
4	6	8	1	1	Pilot	42
5	7	8	2	1	The Thing You Love Most	41
6	8	8	3	1	Snow Falls	42
7	9	8	4	1	The Price Of Gold	45
8	10	5	1	1	Pilot	42
9	11	5	2	1	Allen	44
10	12	5	3	1	Cell Test	43

- Daca timp_ecran este null, il setam la 5 minute
update rol_productie_jucat
set timp_ecran=5
where timp_ecran is null;

```
--daca timp_ecran este null setam la 5 minute
update rol_productie_jucat
set timp_ecran=5
where timp_ecran is null;
```

Script Output x Query Result x

Task completed in 0.025 seconds

6 rows updated.

SQL All Rows Fetched: 10 in 0.006 seconds					
	ROL_PRODUCTIE_JUCAT	ACTOR_ID	ROL_ID	PRODUCTIE_ID	TIMP_ECRAN
1	1	1	1	1	5
2	2	2	2	3	5
3	3	3	3	8	5
4	4	4	4	8	5
5	5	5	5	5	5
6	6	6	6	6	359
7	7	7	7	7	87
8	8	7	7	2	5
9	9	8	8	4	74
10	10	9	9	9	112

- Update la utilizator, punem parola „PAROLA NOUA” daca anul crearii este minim
update utilizator
set parola='PAROLA NOUA'
where extract(year from data_creare)=
(select min(extract(year from data_creare)) from utilizator);

```

update utilizator
set parola='PAROLA NOUA'
where extract(year from data_creare)=
(select min(extract(year from data_creare)) from utilizator);

```

Script Output x Query Result x Query Result 1 x

Task completed in 0.043 seconds

2 rows updated.

	UTILIZATOR_ID	ABONAMENT_ID	PLATA_ID	MAIL	PAROLA	DATA_CREARE	TELEFON
1	100	10	1	johnsmith@gmail.com	PAROLA NOUA	13-JAN-23	762599250
2	101	20	2	aljohnson@gmail.com	johnsonAls	16-JAN-24	742911211
3	102	30	3	brownMICH@gmail.com	safePassword	05-FEB-24	798566123
4	103	40	4	DaViSem1@gmail.com	PAROLA NOUA	14-JAN-23	776544232
5	104	50	5	willdavidson@gmail.com	q34trs67	04-JAN-24	768777565
6	105	60	6	LeeSrh@gmail.com	1lovve56	04-JAN-24	(null)

Exercițiul 14: Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă

```
with complex_view as(
select distinct p.denumire, a.num||' '||a.prenume, r.num_caracter, d.num||' '||d.prenume
from productie p
join rol_productie_jucat rpj on (p.productie_id=rpj.productie_id)
join actor a on(a.actor_id=rpj.actor_id)
join rol r on (r.rol_id=rpj.rol_id)
join productie_director pd on (p.productie_id=pd.productie_id)
join director d on (d.director_id=pd.director_id)
)
```

	DENUMIRE	A.NUM ' ' A.PRENUME	NUME_CARACTER	D.NUM ' ' D.PRENUME
1	Bones	David Boreanaz	Seeley Booth	Hart Hanson
2	Once Upon A Time	Lana Parrilla	Evil Queen	Edward Kitsis
3	Owl House	Robles Sarah-Nicole	Luz Noceda	Dana Terrace
4	Once Upon A Time	Josh Dallas	Prince Charming	Adam Horowitz
5	Death On The Nile	Kenneth Branagh	Hercule Poirot	Kenneth Branagh
6	The Menu	Taylor-Joy Any	Margot	Mark Mylod
7	Poor Things	Emma Stone	Bella Baxter	Yargos Lanthimos
8	A Haunting In Venice	Kenneth Branagh	Hercule Poirot	Kenneth Branagh
9	Once Upon A Time	Josh Dallas	Prince Charming	Edward Kitsis
10	Prison Break	Dominic Purcell	Lincoln Burrows	Paul Scheuring
11	Once Upon A Time	Lana Parrilla	Evil Queen	Adam Horowitz
12	Peaky Blinders	Cillian Murphy	Thomas Shelby	Steven Knight

```
--operatie permisă
select * from complex_view;
```

```
--operatie nepermisă
update complex_view
set denumire='DA'
where nume_caracter like '%a%';
```


Exercițiul 15: Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

OUTER-JOIN – afisarea tuturor actorilor, rolurile si productiile in care joaca

SELECT

```
a.actor_id,  
a.num AS "Nume actor",  
r.rol_id,  
r.num_caracter AS "nume rol",  
p.productie_id,  
p.denumire AS "nume productie"
```

FROM actor a


FULL OUTER JOIN rol_productie_jucat rpj ON a.actor_id = rpj.actor_id

FULL OUTER JOIN productie p ON rpj.productie_id = p.productie_id

FULL OUTER JOIN rol r ON rpj.rol_id = r.rol_id;

```
--outer_join  
SELECT  
    a.actor_id,  
    a.num AS "Nume actor",  
    r.rol_id,  
    r.num_caracter AS "nume rol",  
    p.productie_id,  
    p.denumire AS "nume productie"  
FROM actor a  
FULL OUTER JOIN rol_productie_jucat rpj ON a.actor_id = rpj.actor_id  
FULL OUTER JOIN productie p ON rpj.productie_id = p.productie_id  
FULL OUTER JOIN rol r ON rpj.rol_id = r.rol_id;
```

Script Output x | Query Result x | Query Result 1 x

 | All Rows Fetched: 10 in 0.068 seconds

	ACTOR_ID	Nume actor	ROL_ID	nume rol	PRODUCTIE_ID	nume productie
1	1	Cillian	1	Thomas Shelby	1	Peaky Blinders
2	2	David	2	Seeley Booth	3	Bones
3	3	Lana	3	Evil Queen	8	Once Upon A Time
4	4	Josh	4	Prince Charming	8	Once Upon A Time
5	5	Dominic	5	Lincoln Burrows	5	Prison Break
6	6	Robles	6	Luz Noceda	6	Owl House
7	7	Kenneth	7	Hercule Poirot	2	Death On The Nile
8	7	Kenneth	7	Hercule Poirot	7	A Haunting In Venice
9	8	Taylor-Joy	8	Margot	4	The Menu
10	9	Emma	9	Bella Baxter	9	Poor Things

TOP-N – afisarea tuturor productiilor cu a in nume

select denumire, director_id

from (select *

from productie

where lower(denumire) like '%a%'

order by denumire

)

where rownum<=3

order by rownum desc;

```
--top-n
select denumire, director_id
from (select *
      from productie
      where lower(denumire) like '%a%'
      order by denumire
    )
where rownum<=3
order by rownum desc;
```

Script Output x Query Result x Query Result 1 x		
SQL All Rows Fetched: 3 in 0.065 seconds		
DENUMIRE	DIRECTOR_ID	
1 A Haunting In Venice	606	
2 Death On The Nile	606	
3 Once Upon A Time	303	

DIVISION – afisarea tuturor actorilor care nu joaca in „Once upon a time” și care nu au caracterul cu “evil” în nume

```
select a.actor_id,  
       a.num||' '||a.prenume  
from actor a  
where not exists(select 1 from rol_productie_jucat rpj  
                join rol r on rpj.rol_id=r.rol_id  
                join productie p on rpj.productie_id=p.productie_id  
                where rpj.actor_id=a.actor_id  
                and lower(p.denumire)='once upon a time'  
                )  
and not exists( select 1 from rol_productie_jucat rpj  
                join rol r on r.rol_id=rpj.rol_id  
                where rpj.actor_id=a.actor_id  
                and lower(r.num_caracter)='evil');
```

```
select a.actor_id,  
       a.num||' '||a.prenume  
from actor a  
where not exists(select 1 from rol_productie_jucat rpj  
                join rol r on rpj.rol_id=r.rol_id  
                join productie p on rpj.productie_id=p.productie_id  
                where rpj.actor_id=a.actor_id  
                and lower(p.denumire)='once upon a time'  
                )  
and not exists( select 1 from rol_productie_jucat rpj  
                join rol r on r.rol_id=rpj.rol_id  
                where rpj.actor_id=a.actor_id  
                and lower(r.num_caracter)='evil');
```

Script Output x | Query Result x | Query Result 1 x

SQL | All Rows Fetched: 7 in 0.016 seconds

	ACTOR_ID	A.NUME ' A.PRENUME
1	6	Robles Sarah-Nicole
2	1	Cillian Murphy
3	7	Kenneth Branagh
4	2	David Boreanaz
5	8	Taylor-Joy Any
6	5	Dominic Purcell
7	9	Emma Stone