

Universitatea din București
Facultatea de Matematică și Informatică
Calculatoare și Tehnologia Informației

BAZE DE DATE

COORDONATOR ȘTIINȚIFIC :
Prof. Vasile Silviu-Laurențiu

STUDENT:
Codreanu Radu-Ștefan
Grupa 261

BUCUREȘTI
2021

Universitatea din București
Facultatea de Matematică și Informatică
Calculatoare și Tehnologia Informației

Lanț de magazine cinematografice

COORDONATOR ȘTIINȚIFIC :
Prof. Vasile Silviu-Laurențiu

STUDENT:
Codreanu Radu-Ștefan
Grupa 261

BUCUREȘTI
2021

CUPRINS

1.	<i>Prezentarea modelului și a regulilor acestuia</i>	1
1.1.	<i>Descrierea proiectului și motivarea temei alese</i>	1
1.2.	<i>Regulile proiectului</i>	1
2.	<i>Diagrama entitate-relație</i>	3
2.1.	<i>Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților</i>	4
3.	<i>Diagrama conceptuală</i>	13
3.1.	<i>Descrierea constrângerilor de integritate</i>	14
3.2.	<i>Schemele relaționale</i>	17
4.	<i>Implementarea într-un sistem de gestiune a bazelor de date</i>	19
4.1.	<i>Crearea tabelelor + constrângeri (incluzând FK)</i>	19
4.2.	<i>Introducere date</i>	27

1. Prezentarea modelului și a regulilor acestuia

1.1. Descrierea proiectului și motivarea temei alese

Încă de mic am fost pasionat de lumea cinematografeiei, chiar dacă pe vremea aceea nu știam exact ce presupune această lume mai exact. Odată cu trecerea timpului am reușit să înțeleg mai bine cum un film ajunge atât pe marile ecrane din cinematografe cât și pe ecranele de la noi de acasă. Dacă ar fi vreodată să îmi deschid o afacere de nivel mai mare, aș începe cu un lanț de magazine cinematografice, magazine care vând produsele pe care le-am îndrăgit încă de mic, filme.

Magazinile prezentate sunt inventate de către mine, însă doresc să menționez că am avut ca sursă de inspirație provider-ul american „Blockbuster LLC” , care între anii 1990-2004 a dominat piața produselor home-media.

O cerință importantă la care orice om care vrea să își deschidă o afacere ar trebui să fie atent este realizarea unei baze de date concisă și funcțională, iar acest lucru voi încerca să realizez în acest proiect. Bazele de date sunt din ce în ce mai frecvente în timpul nostru, iar puține lucruri pot să existe fără a avea o bază de date în spatele lor.

1.2. Regulile proiectului

- Un magazin are în stoc mai multe filme
- Nu există magazine goale, fără niciun film
- Un film se poate regăsi în mai multe magazine
- Există filme care nu mai sunt în stoc
- Magazinele prezentate mai jos operează doar în România
- Un film se poate afla sub mai multe formaturi (DvD, blu-ray, VHS etc.)
- Un film poate avea mai mulți regizori, însă trebuie să aibă minim unul

- Există regizori care nu aparțin de niciun film
- Un film trebuie să aibă un singur distribuitor
- Un distribuitor poate avea în catalogul său mai multe filme
- Comanda este livrată de un singur curier
- Curierul respectiv poate să nu mai livreze
- Ca o comandă să fie plasată aceasta trebuie să conțină minim un film
- Pot exista filme care nu sunt comandate
- Un client poate plasa mai multe comenzi sau niciuna
- Un client trebuie să aibă o adresă
- Moneda pentru efectuarea tranzacțiilor este RON

2. Diagrama entitate-relație

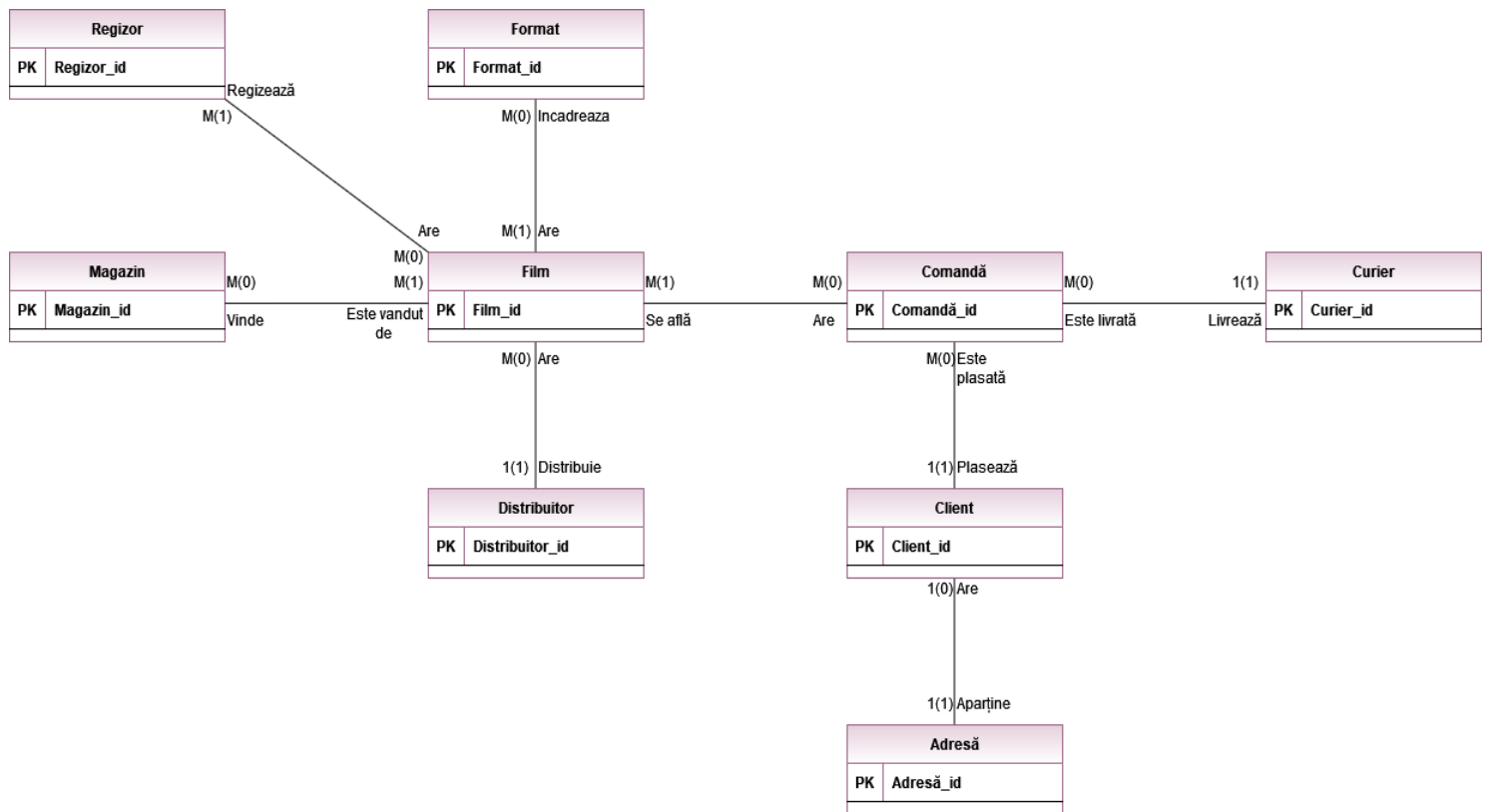


Figura 1. Diagrama entitate-relație

2.1. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

TABELUL MAGAZIN

◆Descriere entitate:

- Tabelul MAGAZIN stochează caracteristicile unui magazin, de exemplu numele, orașul în care se află, website-ul etc. Fiecare magazin are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele magazinului
- Oraș – reprezintă locația magazinului respectiv
- Număr_telefon – reprezintă una dintre metodele prin care se poate lua contact cu magazinul respectiv
- E-mail – reprezintă altă metodă de contact
- Website – reprezintă pagina de internet a magazinului

◆Chei:

- Magazin_id (PK) – identificator unic pentru fiecare magazin.

◆Relații și cardinalități:

- Cele două tabele, „Magazin” și „Film”, sunt într-o relație de tip „many-to-many”. Astfel, apare un tabel intermediar, numit „VindeFilm”. Un magazin vinde mai multe filme, iar cel puțin un film se regăsește în mai multe magazine. Nu există magazine fără filme, însă există filme care nu se mai află în stoc.

TABELUL FILM

◆Descriere entitate:

- Tabelul FILM stochează caracteristicile unui film, de exemplu numele, durata, pegi-ul, etc. Fiecare film are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele filmului
- Durată – reprezintă lungimea filmului (în minute)
- Pegi – reprezintă limita de vârstă pentru vizionare
- Limbă – reprezintă dublajul audio al filmului
- Gen – reprezintă categoria filmului (Acțiune, Horror etc.)
- Preț – reprezintă costul achiziționării filmului

◆Chei:

- Film_id (PK) – identificator unic pentru fiecare film
- Distribuitor_id (FK1) – reprezintă legătura cu tabela „Distribuitor”

◆Relații și cardinalități:

- Cele două entități, „Film” și „Regizor”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „RegizeazăFilm”. Un regizor poate să regizeze mai multe filme, iar un film are cel puțin un regizor. Pot să existe și regizori la început de drum fără niciun film în repertoriu.
- Cele două entități, „Film” și „Format”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „ÎncadreazăFilm”. Un format are mai multe filme, iar un film se încadrează în cel puțin un format. Filmele care nu se prezintă sub un format nu mai sunt în stoc.

- Cele două entități, „Film” și „Comandă”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „FilmComandă”. Un film se poate găsi în mai multe comenzi, iar o comandă trebuie să conțină cel puțin un film.
- Cele două entități, „Film” și „Distribuitor”, sunt într-o relație de tip „one-to-many”. Un film are un singur distribuitor (doar unul), iar un distribuitor are zero sau mai multe filme în catalogul său.

TABELUL REGIZOR

◆Descriere entitate:

- Tabelul „Regizor” reprezintă un tablou unde se vor stoca informații cu privire la datele personale și la metodele de contact ale unui regizor. Fiecare regizor are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele regizorului
- Prenume – reprezintă prenumele regizorului
- Data_nașterii – reprezintă data în care s-a născut regizorul
- E-mail – reprezintă o metodă de contact
- Studii_superioare – reprezintă facultatea absolvită a regizorului

◆Chei:

- Regizor_id (PK) – identificator unic pentru fiecare regizor

◆Relații și cardinalități:

- Cele două entități, „Film” și „Regizor”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „RegizeazăFilm” . Un regizor poate să regizeze mai multe filme, iar un film are cel puțin un regizor.

TABELUL FORMAT

◆Descriere entitate:

- Tabelul FORMAT stochează caracteristicile tipului de stochare al unui film. Fiecare format are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Tip – reprezintă metoda de stochare a filmului (DvD, VHS, etc.)

◆Chei:

- Format_id (PK) – identificator unic pentru fiecare format

◆Relații și cardinalități:

- Cele două entități, „Film” și „Format”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „ÎncadreazăFilm” . Un format are mai multe filme, iar un film se încadrează în cel puțin un format.

TABELUL DISTRIBUTOR

◆Descriere entitate:

- Tabelul DISTRIBUTOR stochează caracteristicile unui distribuitor, adică compania care trebuie să identifice tipul de spectatori pentru acel film și să se gândească la metode de a promova produsul respectiv. Exemplu caracteristici : numele, locația sediului, website-ul etc. Fiecare distribuitor are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele distribuitorului
- Sediul – reprezintă locația în care se află sediul distribuitorului (oraș)
- Data_fondării – reprezintă data în care firma a fost creată
- Proprietar – reprezintă părintele companiei
- Website – reprezintă zona cu informații precise despre distribuitor

◆Chei:

- Distribuitor_id (PK) – identificator unic pentru fiecare distribuitor

◆Relații și cardinalități:

- Cele două entități, „Film” și „Distribuitor”, sunt într-o relație de tip „one-to-many”. Un film are un singur distribuitor (doar unul), iar un distribuitor are zero sau mai multe filme în catalogul său.

TABELUL COMANDĂ

◆Descriere entitate:

- Tabelul COMANDĂ stochează caracteristicile unei comenzi care a fost plasată de către un client. Fiecare comandă are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Awb – reprezintă un cod unic al comenzii
- Status – reprezintă dacă comanda a fost livrată sau nu
- Dată_plasare – reprezintă data în care comanda a fost plasată
- Dată_livrare – reprezintă estimarea datei în care comanda va fi livrată

◆Chei:

- Comandă_id (PK) – identificator unic pentru fiecare comandă
- Curier_id (FK1) – reprezintă legătura cu tabela „Curier”
- Client_id (FK2) – reprezintă legătura cu tabela „Client”

◆Relații și cardinalități:

- Cele două entități, „Comandă” și „Curier”, sunt într-o relație de tip „one-to-many”. O comandă este livrată de un singur curier (doar unul), iar un curier poate livra zero sau mai multe comenzi.
- Cele două entități, „Comandă” și „Client”, sunt într-o relație de tip „one-to-many”. O comandă este plasată de un singur client (doar unul), iar un client poate plasa zero sau mai multe comenzi.
- Cele două entități, „Film” și „Comandă”, sunt într-o relație de tip „many-to-many”. Datorită acestei relații mai apare un tabel intermediar numit „FilmComandă”. Un film se poate găsi în mai multe comenzi, iar o comandă trebuie să conțină cel puțin un film.

TABELUL CURIER

◆Descriere entitate:

- Tabelul CURIER stochează informații despre curierii care transportă comenzile. Fiecare curier are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele șoferului
- Prenume – reprezintă prenumele șoferului
- Cost_transport – reprezintă prețul pentru livrarea coletului
- Firmă_curierat – reprezintă numele firmei de care aparține curierul
- Telefon – reprezintă numărul de telefon al șoferului

◆Chei:

- Curier_id (PK) – identificator unic pentru fiecare curier

◆Relații și cardinalități:

- Cele două entități, „Comandă” și „Curier”, sunt într-o relație de tip „one-to-many”. O comandă este livrată de un singur curier (doar unul), iar un curier poate livra zero sau mai multe comenzi.

TABELUL CLIENT

◆Descriere entitate:

- Tabelul CLIENT stochează informații despre clienții care au apelat la serviciile noastre. Fiecare client are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Nume – reprezintă numele clientului
- Prenume – reprezintă prenumele clientului
- E-mail – reprezintă adresa de mail a clientului
- Telefon – reprezintă numărul de telefon al clientului
- Data_nașterii – reprezintă ziua de naștere a clientului

◆Chei:

- Client_id (PK) – identificator unic pentru fiecare curier
- Adresa_id (FK1) – legătura cu tabela „Adresă”

◆Relații și cardinalități:

- Cele două entități, „Client” și „Adresă”, sunt într-o relație de tip „one-to-one”. Un client trebuie să aibă o adresă (strict una), iar o adresă aparține de un client. Există adrese care nu aparțin de niciun client.
- Cele două entități, „Comandă” și „Client”, sunt într-o relație de tip „one-to-many”. O comandă este plasată de un singur client (doar unul), iar un client poate plasa zero sau mai multe comenzi.

TABELUL ADRESĂ

◆Descriere entitate:

- Tabelul ADRESĂ stochează informații despre adresa clienților pentru finalizarea livrării. Fiecare adresă are un cod unic la care doar administratorul bazei de date are acces.

◆Descriere atribute:

- Județ – reprezintă numele țării pentru livrarea comenzii
- Oraș – reprezintă numele orașului
- Strada – reprezintă numele străzii
- Număr – reprezintă numărul locuinței clientului
- Bloc – reprezintă numele blocului
- Cod_poștal – reprezintă codul poștal adresei

◆Chei:

- Adresa_id (PK) – identificator unic pentru fiecare adresă

◆Relații și cardinalități:

- Cele două entități, „Client” și „Adresă”, sunt într-o relație de tip „one-to-one”. Un client trebuie să aibă o adresă (strict una), iar o adresă aparține de un client. Există adrese care nu aparțin de niciun client.

3. Diagrama conceptuală

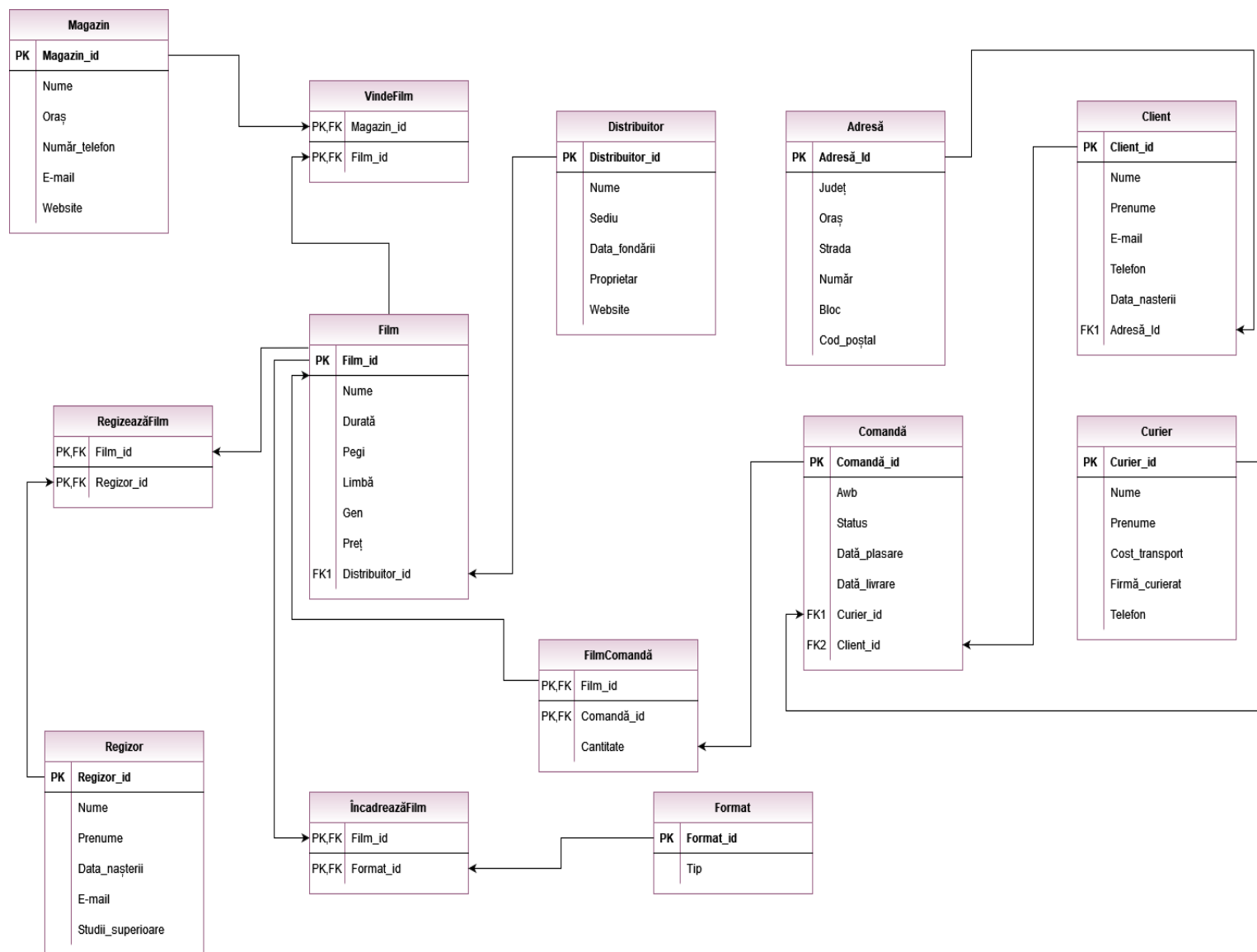


Figura 2. Diagramă conceptuală

3.1. Descrierea constrângerilor de integritate

MAGAZIN

- Nume, Oraș, Număr_telefon, E-mail, Website nu trebuie să fie NULL
- Magazin_id este PK
- Nume – conține doar litere – trebuie să aibă minim un caracter
- Oraș – conține doar litere
- Număr_telefon – conține doar cifre – trebuie să aibă 10 cifre –este unic
- E-mail – este unic
- Website – este unic

FILM

- Nume, Durată, Pegi, Limbă, Gen, Preț nu trebuie să fie NULL
- Film_id este PK
- Nume – trebuie să aibă minim un caracter
- Durată – conține doar cifre – este mai mare decât 30
- Limbă – conține doar litere
- Preț – conține doar cifre – este mai mare decât 0

REGIZOR

- Nume, Prenume, Data_nașterii, E-mail nu trebuie să fie NULL
- Regizor_id este PK
- Nume – conține doar litere

- Prenume – conține doar litere
- Email – trebuie să fie unic
- Studii_superioare este bool cu 2 opțiuni(D – Da, N – Nu)

FORMAT

- Tip nu trebuie să fie NULL
- Tip_id este PK
- Tip – nu trebuie să conțină cifre

DISTRIBUITOR

- Nume, Sediul, Data_fondării, Proprietar, Website trebuie să fie diferite de NULL
- Distribuitor_id este PK
- Nume – conține doar litere – este unic
- Proprietar – conține doar litere

COMANDĂ

- Awb, Status, Data_plasare, Data_livrare trebuie să fie diferit de NULL
- Comandă_id este PK
- Awb – este unic
- Status – conține doar litere
- Data_plasare – trebuie să fie mai mică decât data_livrare

CURIER

- Nume, Prenume, Cost_transport, Firmă_curierat, Telefon trebuie să fie diferit de NULL
- Curier_id este PK
- Nume – trebuie să fie unic – conține doar litere
- Prenume – trebuie să fie unic – conține doar litere
- Telefon – este unic – conține doar cifre

CLIENT

- Nume, Prenume, Email, Telefon, Data_nașterii trebuie să fie diferit de NULL
- Client_id este PK
- Nume – trebuie să fie unic – conține doar litere
- Prenume – trebuie să fie unic – conține doar litere
- Email – este unic

ADRESĂ

- Județ, Oraș, Strada, Număr, Cod_poștal nu trebuie să fie NULL
- Adresă_id este PK
- Județ – trebuie să fie alcătuite doar din litere
- Oraș – trebuie să fie alcătuite doar din litere
- Stradă – trebuie să fie alcătuite doar din litere
- Număr – conține doar cifre
- Cod_poștal – conține doar cifre

VINDEFILM

- Film_id este FK
- Magazin_id este FK
- Film_id și Magazin_id formează PK

REGIZEAZĂFILM

- Film_id este FK
- Regizor_id este FK
- Film_id și Regizor_id formează PK

ÎNCADREAZĂFILM

- Film_id este FK
- Format_id este FK
- Film_id și Format_id formează PK

FILMCOMANDĂ

- Film_id este FK
- Comandă_id este FK
- Film_id și Comandă_id formează PK
- Cantitate nu trebuie să fie NULL

3.2. Schemele relaționale

Schemele relaționale atașate diagramei conceptuale sunt:

- **MAGAZIN** (Magazin_id#, Nume, Oraș, Număr_telefon, E-mail, Website)
- **VindeFilm** (Magazin_id# <FK>, Film_id# <FK>)
- **FILM** (Film_id#, Distribuitor_id<FK> Nume, Durată, Pegi, Limbă, Gen, Preț)
- **REGIZOR**(Regizor_id#,Nume,Prenume, Data_nașterii, E-mail, Studii_superioare)
- **RegizeazăFilm** (Film_id# <FK>, Regizor_id# <FK>)
- **FORMAT** (Format_id#, Tip)
- **ÎncadreazăFilm** (Film_id# <FK>, Format_id# <FK>)
- **DISTRIBUITOR**(Distribuitor_id#,Nume,Sediu,Data_fondării,Proprietar,Website)
- **COMANDĂ**(Comandă_id#,Client_id<FK>,Curier_id<FK>,Awb,Status,Data_plasare, Data_livrare)
- **FilmComandă** (Film_id# <FK>, Comandă_id# <FK>, Cantitate)
- **CURIER** (Curier_id#, Nume, Prenume, Cost_transport, Firmă_curierat, Telefon)
- **CLIENT** (Client_id#, Adresă_id<FK>, Nume, Prenume, E-mail, Telefon, Data_nașterii)
- **ADRESĂ** (Adresă_id#, Județ, Oraș, Strada, Număr, Bloc, Cod_poștal)

Există 12 constrângeri ON DELETE, câte una pentru fiecare constrângere de tip FOREIGN KEY. Acestea sunt:

- VindeFilm_Magazin_id_FK, ON DELETE CASCADE
 - dacă un magazin este șters, atunci, filmele din cadrul acestuia vor fi și ele șterse la rândul lor.
- VindeFilm_Film_id_FK, ON DELETE CASCADE
 - dacă un film este șters, magazinul va rămâne cu restul de filme, fără cel șters.
- RegizeazăFilm_Film_id_FK, ON DELETE CASCADE
 - dacă un film este șters, atunci, regizorul poate să regizeze alte filme.
- RegizeazăFilm_Regizor_id_FK, ON DELETE CASCADE
 - dacă un regizor este șters, filmul regizat de acesta va fi șters la rândul lui.
- ÎncadreazăFilm_Film_id_FK, ON DELETE CASCADE
 - dacă un film este șters, format-ul rămâne în continuare.
- ÎncadreazăFilm_Format_id_FK, ON DELETE CASCADE
 - dacă un format este șters, filmul respectiv poate rula pe mai multe formaturi.
- FilmComandă_Film_id_FK, ON DELETE CASCADE
 - dacă un film este șters, comanda va fi și ea ștearsă la rândul ei.
- FilmComandă_Comandă_id_FK, ON DELETE CASCADE
 - dacă o comanda va fi ștearsă, filmul va rămâne.
- Film_Distribuitoar_id_FK, ON DELETE SET NULL
 - dacă distribuitorul este șters, filmul nu va mai avea un distribuitor specific dar va rămâne în magazin.
- Client_Adresă_id_FK, ON DELETE CASCADE
 - dacă o adresă este ștearsă, atunci toți clienții care aveau această adresă vor fi șterși.
- Comandă_Curier_id_FK, ON DELETE SET NULL
 - dacă un curier este șters, comanda livrată de acesta va putea fi livrată de altă firmă.
- Comandă_Client_id_FK, ON DELETE CASCADE
 - dacă un client este șters, comanda plasată de acesta va fi și ea ștearsă la rândul ei

4. Implementarea într-un sistem de gestiune a bazelor de date

4.1. Crearea tabelelor + constrângeri (incluzând FK)

--CREARE ENTITATI

--TABELUL MAGAZIN

CREATE TABLE "Magazin"

(

Magazin_id number(4) CONSTRAINT Magazin_Magazin_id_PK PRIMARY KEY,

Nume varchar2(25) CONSTRAINT Magazin_Nume_c1 CHECK (LENGTH(Nume) >= 1)

CONSTRAINT Magazin_Nume_nn NOT NULL CONSTRAINT Magazin_Nume_c2 CHECK
(Nume NOT LIKE '%[^A-Z]%'),

Oras varchar2(15) CONSTRAINT Magazin_Oras_nn NOT NULL CONSTRAINT Magazin_Oras_c
CHECK (Oras NOT LIKE '%[^A-Z]%'),

Numar_telefon varchar2(11) CONSTRAINT Magazin_Telefon_u UNIQUE CONSTRAINT
Magazin_Telefon_c1 CHECK (Numar_telefon NOT LIKE '%[^0-9]%') CONSTRAINT
Magazin_Telefon_nn NOT NULL CONSTRAINT Magazin_telefon_c2 CHECK (LENGTH
(Numar_telefon)=10),

Email varchar2(25) CONSTRAINT Magazin_Email_nn NOT NULL CONSTRAINT
Magazin_Email_u UNIQUE,

Website varchar2(25) CONSTRAINT Magazin_Website_nn NOT NULL CONSTRAINT
Magazin_Website_u UNIQUE

);

--TABELUL DISTRIBUTOR

CREATE TABLE "Distribuitor"

(

Distribuitor_id number(4) CONSTRAINT Distr_Distr_id_PK PRIMARY KEY,

Nume varchar2(25) CONSTRAINT Distr_Nume_nn NOT NULL CONSTRAINT Distr_Nume_c
CHECK (Nume NOT LIKE '%[^A-Z]%'),

Sediu varchar2(20) CONSTRAINT Distr_Sediu_nn NOT NULL,

Data_fondarii date CONSTRAINT Distr_Data_fondarii_nn NOT NULL,

Proprietar varchar2(15) CONSTRAINT Distr_Proprietar_nn NOT NULL CONSTRAINT
Distr_Proprietar_c CHECK (Proprietar NOT LIKE '%[^A-Z]%'),

Website varchar2(25) CONSTRAINT Distr_Website_nn NOT NULL

);

-- Daca lasam Distribuitor la constraint-uri nu putea sa rulez din cauza lungimii

--TABELUL FILM

CREATE TABLE "Film"

(

Film_id number(4) CONSTRAINT Film_Film_id_PK PRIMARY KEY,

Nume varchar2(25) CONSTRAINT Film_Nume_nn NOT NULL CONSTRAINT Film_Nume_c
CHECK (LENGTH(Nume) >=1),

Durata number(3) CONSTRAINT Film_Durata_nn NOT NULL CONSTRAINT Film_Durata_c1
CHECK (Durata NOT LIKE '%[^0-9]%') CONSTRAINT Film_Durata_c2 CHECK (Durata >30),

```

Pegi varchar(10) CONSTRAINT Film_Pegi_nn NOT NULL,

Limba varchar(15) CONSTRAINT Film_Limba_nn NOT NULL CONSTRAINT Film_Limba_c
CHECK (Limba NOT LIKE '%[^A-Z]%),

Gen varchar(10) CONSTRAINT Film_Gen_nn NOT NULL,

Pret number(4) CONSTRAINT Film_Pret_nn NOT NULL CONSTRAINT Film_Pret_c CHECK
(Pret >0),

Distribuitoar_id number(4) CONSTRAINT Film_Distr_id_FK REFERENCES "Distribuitoar"
(Distribuitoar_id) ON DELETE SET NULL

);

```

--TABELUL INTERMEDIAR VINDEFILM

```

CREATE TABLE "VindeFilm"

(

Magazin_id number(4) CONSTRAINT VindeFilm_Magazin_id_FK REFERENCES "Magazin"
(Magazin_id) ON DELETE CASCADE,

Film_id number(4) CONSTRAINT VindeFilm_Film_id_FK REFERENCES "Film" (Film_id) ON
DELETE CASCADE,

CONSTRAINT VindeFilm_PK PRIMARY KEY (Magazin_id, Film_id)

);

```


--TABELUL REGIZOR

CREATE TABLE "Regizor"

(

Regizor_id number(4) CONSTRAINT Regizor_Regizor_id_PK PRIMARY KEY,

Nume varchar2(20) CONSTRAINT Regizor_Nume_nn NOT NULL CONSTRAINT

Regizor_Nume_c CHECK (Nume NOT LIKE '%[^A-Z]%'),

Prenume varchar2(20) CONSTRAINT Regizor_Prenume_nn NOT NULL CONSTRAINT

Regizor_Prenume_c CHECK (Prenume NOT LIKE '%[^A-Z]%'),

Data_nasterii date CONSTRAINT Regizor_Data_nasterii_nn NOT NULL,

Email varchar2(25) CONSTRAINT Regizor_Email_nn NOT NULL CONSTRAINT

Regizor_Email_u UNIQUE,

Studii_superioare char(1) CONSTRAINT Regizor_Studii_superioare_c CHECK (Studii_superioare
IN ('D', 'N'))

);

--TABELUL REGIZEAZAFILM

CREATE TABLE "RegizeazaFilm"

(

Film_id number(4) CONSTRAINT RegizeazaFilm_Film_id_FK REFERENCES Film (Film_id) ON
DELETE CASCADE,

Regizor_id number(4) CONSTRAINT RegizeazaFilm_Regizor_id_FK REFERENCES Regizor
(Regizor_id) ON DELETE CASCADE,

CONSTRAINT RegizeazaFilm_PK PRIMARY KEY (Film_id, Regizor_id)

);

--TABELUL FORMAT

CREATE TABLE "Format"

(

Format_id number(4) CONSTRAINT Format_Format_id_PK PRIMARY KEY,

Tip varchar2 (10) CONSTRAINT Format_Tip_nn NOT NULL CONSTRAINT Format_Tip_c
CHECK (Tip NOT LIKE '%[^A-Z]%')

);

--TABELUL INCADREAZAFILM

CREATE TABLE "IncadreazaFilm"

(

Film_id number(4) CONSTRAINT IncadreazaFilm_Film_id_FK REFERENCES Film (Film_id) ON
DELETE CASCADE,

Format_id number(4) CONSTRAINT IncadreazaFilm_Format_id_FK REFERENCES "Format"
(Format_id) ON DELETE CASCADE,

CONSTRAINT IncadreazaFilm_PK PRIMARY KEY (Film_id, Format_id)

);

--TABELUL COMANDA

CREATE TABLE "Comanda"

(

Comanda_id number(4) CONSTRAINT Comanda_Comanda_id_PK PRIMARY KEY,

Awb varchar2 (50) CONSTRAINT Comanda_Awb_nn NOT NULL CONSTRAINT

Comanda_Awb_u UNIQUE,

Status varchar2 (15) CONSTRAINT Comanda_Status_nn NOT NULL CONSTRAINT

Comanda_Status_c CHECK (Status NOT LIKE '%[^A-Z]%),

Data_plasare date CONSTRAINT Comanda_Data_plasare_nn NOT NULL,

Data_livrare date CONSTRAINT Comanda_Data_livrare_nn NOT NULL,

CONSTRAINT Comanda_Data_plasare_c CHECK (Data_plasare < Data_livrare),

Curier_id number(4) CONSTRAINT Comanda_Curier_id_FK REFERENCES "Curier" (Curier_id)
ON DELETE SET NULL,

Client_id number(4) CONSTRAINT Comanda_Client_id_FK REFERENCES "Client" (Client_id)
ON DELETE CASCADE

);

--TABELUL CURIER

CREATE TABLE "Curier"

(

Curier_id number(4) CONSTRAINT Curier_Curier_id_PK PRIMARY KEY,

Nume varchar2(20) CONSTRAINT Curier_Nume_nn NOT NULL CONSTRAINT Curier_Nume_c
CHECK (Nume NOT LIKE '%[^A-Z]%) CONSTRAINT Curier_Nume_u UNIQUE,

```

Prenume varchar2(20) CONSTRAINT Curier_Prenume_nn NOT NULL CONSTRAINT
Curier_Prenume_c CHECK (Prenume NOT LIKE '%[^A-Z]%' ) CONSTRAINT Curier_Prenume_u
UNIQUE,

Cost_transport number(10) CONSTRAINT Curier_Cost_transport_nn NOT NULL,

Firma_curierat varchar2(20) CONSTRAINT Curier_Firma_curierat_nn NOT NULL,

Telefon varchar2(11) CONSTRAINT Curier_Telefon_u UNIQUE CONSTRAINT
Curier_Telefon_c1 CHECK (Telefon NOT LIKE '%[^0-9]%' ) CONSTRAINT Curier_Telefon_nn
NOT NULL

);

```

--TABELUL ADRESA

```

CREATE TABLE "Adresa"

(

Adresa_id number(4) CONSTRAINT Adresa_Adresa_id_PK PRIMARY KEY,

Judet varchar(15) CONSTRAINT Adresa_Judet_nn NOT NULL CONSTRAINT Adresa_Judet_c
CHECK (Judet NOT LIKE '%[^A-Z]%' ),

Oras varchar(15) CONSTRAINT Adresa_Oras_nn NOT NULL CONSTRAINT Adresa_Oras_c
CHECK (Oras NOT LIKE '%[^A-Z]%' ),

Strada varchar(15) CONSTRAINT Adresa_Strada_nn NOT NULL CONSTRAINT Adresa_Strada_c
CHECK (Strada NOT LIKE '%[^A-Z]%' ),

Numar number(5) CONSTRAINT Adresa_Numar_nn NOT NULL CONSTRAINT Adresa_Numar_c
CHECK (Numar NOT LIKE '%[^0-9]%' ),

"Bloc" varchar(5),

Cod_postal number(20) CONSTRAINT Adresa_Cod_postal_nn NOT NULL CONSTRAINT
Adresa_Cod_postal_c CHECK (Cod_postal NOT LIKE '%[^0-9]%' )

);

```

--TABELUL CLIENT

CREATE TABLE "Client"

(

Client_id number(4) CONSTRAINT Client_Client_id_PK PRIMARY KEY,

Nume varchar2(20) CONSTRAINT Client_Nume_nn NOT NULL CONSTRAINT Client_Nume_c
CHECK (Nume NOT LIKE '%[^A-Z]%) CONSTRAINT Client_Nume_u UNIQUE,

Prenume varchar2(20) CONSTRAINT Client_Prenume_nn NOT NULL CONSTRAINT
Client_Prenume_c CHECK (Prenume NOT LIKE '%[^A-Z]%) CONSTRAINT Client_Prenume_u
UNIQUE,

Email varchar(20) CONSTRAINT Client_Email_nn NOT NULL CONSTRAINT Client_Email_u
UNIQUE,

Telefon varchar(11) CONSTRAINT Client_Telefon_nn NOT NULL,

Data_nasterii date CONSTRAINT Client_Data_nasterii_nn NOT NULL,

Adresa_id number(4) CONSTRAINT Client_Adresa_id_FK REFERENCES "Adresa" (Adresa_id)
ON DELETE CASCADE

);

--TABELUL FILMCOMANDA

CREATE TABLE "FilmComanda"

(

Film_id number(4) CONSTRAINT FilmComanda_Film_id_FK REFERENCES "Film" (Film_id) ON
DELETE CASCADE,

```

Comanda_id number(4) CONSTRAINT FilmComanda_Comanda_id_FK REFERENCES
"Comanda" (Comanda_id) ON DELETE CASCADE,

CONSTRAINT FilmComanda_PK PRIMARY KEY (Comanda_id, Film_id),

Cantitate number(5) CONSTRAINT FilmComanda_Cantitate_nn NOT NULL

);

```

4.2. Introducere date

--INSERARE DATE IN TABELE

--1-MAGAZIN

```

INSERT INTO "Magazin"

```

```

VALUES (1000,'Blockbuster','Bucuresti','0712453687','blockbuster@gmail.com','www.blockbuster.ro');

```

```

INSERT INTO "Magazin"

```

```

VALUES

```

```

(1001,'CinemaTown','Targoviste','0712458587','cinematown@gmail.com','www.cinematown.ro');

```

```

INSERT INTO "Magazin"

```

```

VALUES

```

```

(1002,'MiracleMovie','Constanta','0710528587','miraclemovie@gmail.com','www.miraclemovie.ro');

```

```

INSERT INTO "Magazin"

```

```

VALUES (1003,'IMOVIE','Cluj-Napoca','0712973487','imovie@gmail.com','www.imovie.ro');

```

```

INSERT INTO "Magazin"

```

```

VALUES (1004,'Super-Movie','Timisoara','0717948587','supermovie@gmail.com','www.supermovie.ro');

```

```

INSERT INTO "Magazin"

```

```

VALUES (1005,'Cinefil','Iasi','0774358587','cinefil@gmail.com','www.cinefil.ro');

```

commit;

--2-DISTRIBUTOR

INSERT INTO "Distributeur"

VALUES (2000,'Sony Pictures','Toronto','15-01-1967','Sony','www.sonypictures.com');

INSERT INTO "Distributeur"

VALUES (2001,'Disney Pictures','Paris','12-11-1921','Walt Disney','www.disneypictures.com');

INSERT INTO "Distributeur"

VALUES (2002,'20th Century Fox','Los Angeles','31-05-1935','Walt Disney','www.20thfox.com');

INSERT INTO "Distributeur"

VALUES (2003,'Paramount Pictures','Hollywood','08-05-1912','ViacomCBS','www.paramountpictures.com');

INSERT INTO "Distributeur"

VALUES (2004,'A24','New York City','20-08-2012','A24 Films','www.a24.com');

INSERT INTO "Distributeur"

VALUES (2005,'New Line Productions','Burbank','18-06-1967','New Line Cinema','www.newlineprod.com');

commit;

--3-FILM

INSERT INTO "Film"

VALUES (3000, 'Spider-Man 2', 185, 'PG-8', 'Engleza', 'Superhero', 20, 2000);

INSERT INTO "Film"

VALUES (3001, 'Regele Leu', 128, 'PG-3', 'Romana', 'Aventura', 25, 2001);

INSERT INTO "Film"

VALUES (3002, 'Deadpool', 145, 'R-18', 'Engleza', 'Actiune', 30, 2002);

INSERT INTO "Film"

VALUES (3003, 'Scream', 110, 'NC-17', 'Engleza', 'Horror', 15, 2003);

INSERT INTO "Film"

VALUES (3004, 'Ex Machina', 158, 'PG-13', 'Germana', 'Sci-Fi', 20, 2004);

INSERT INTO "Film"

VALUES (3005, 'The Lord of the Rings', 210, 'PG-13', 'Romana', 'Fantasy', 35, 2005);

commit;

--4-VINDEFILM

INSERT INTO "VindeFilm"

VALUES (1000, 3000);

INSERT INTO "VindeFilm"

VALUES (1001, 3001);

INSERT INTO "VindeFilm"

VALUES (1002, 3002);

INSERT INTO "VindeFilm"

VALUES (1003, 3003);

INSERT INTO "VindeFilm"

VALUES (1004, 3004);

INSERT INTO "VindeFilm"

VALUES (1005, 3005);

commit;

--5-REGIZOR

INSERT INTO "Regizor"

VALUES (4000, 'Raimi','Sam', '23-10-1959', 'samraimi@gmail.com','D');

INSERT INTO "Regizor"

VALUES (4001, 'Roger','Allers', '29-06-1949', 'rogerallers@gmail.com','N');

INSERT INTO "Regizor"

VALUES (4002, 'Minkoff','Robert', '11-08-1962', 'rpbertyminkoff@yahoo.com','N');

INSERT INTO "Regizor"

VALUES (4003, 'Miller','Tim', '10-10-1964', 'tmiller@gmail.com','N');

INSERT INTO "Regizor"

VALUES (4004, 'Craven','Wesley', '02-08-1939', 'wescraven@yahoo.com','D');

INSERT INTO "Regizor"

VALUES (4005, 'Garland','Alexander', '26-05-1970', 'alexgarland@yahoo.com','D');

```
INSERT INTO "Regizor"  
  
VALUES (4006, 'Jackson','Peter', '31-10-1961', 'petejck@yahoo.com','D');  
  
commit;
```

--6-REGIZEAZAFILM

```
INSERT INTO "RegizeazaFilm"  
  
VALUES (3000, 4000);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3001, 4001);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3001, 4002);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3002, 4003);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3003, 4004);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3004, 4005);  
  
INSERT INTO "RegizeazaFilm"  
  
VALUES (3005, 4006);  
  
commit;
```

--7-FORMAT

INSERT INTO "Format"

VALUES (5000, 'Dvd');

INSERT INTO "Format"

VALUES (5001, 'blu-ray');

INSERT INTO "Format"

VALUES (5002, 'VHS');

INSERT INTO "Format"

VALUES (5003, 'Videotape');

INSERT INTO "Format"

VALUES (5004, 'Cassette');

INSERT INTO "Format"

VALUES (5005, 'Betamax');

INSERT INTO "Format"

VALUES (5006, 'Video CD');

commit;

--8-INCADREAZAFILM

INSERT INTO "IncadreazaFilm"

VALUES (3000, 5000);

INSERT INTO "IncadreazaFilm"

VALUES (3000, 5001);

INSERT INTO "IncadreazaFilm"

VALUES (3001, 5002);

INSERT INTO "IncadreazaFilm"

VALUES (3002, 5003);

INSERT INTO "IncadreazaFilm"

VALUES (3003, 5004);

INSERT INTO "IncadreazaFilm"

VALUES (3003, 5005);

INSERT INTO "IncadreazaFilm"

VALUES (3004, 5001);

INSERT INTO "IncadreazaFilm"

VALUES (3004, 5006);

INSERT INTO "IncadreazaFilm"

VALUES (3005, 5002);

INSERT INTO "IncadreazaFilm"

VALUES (3005, 5003);

INSERT INTO "IncadreazaFilm"

VALUES (3005, 5004);

commit;

--9-CURIER

INSERT INTO "Curier"

VALUES (6000, 'Dumitrascu', 'Marian', '15', 'FedEx', '0726495703');

INSERT INTO "Curier"

VALUES (6001, 'Condurache', 'George', '10', 'Cargus', '0726432754');

INSERT INTO "Curier"

VALUES (6002, 'Cristache', 'Andrei', '20', 'Fan Courier', '0724856103');

INSERT INTO "Curier"

VALUES (6003, 'Negut', 'Radu', '10', 'SameDay', '0723215773');

INSERT INTO "Curier"

VALUES (6004, 'Matache', 'Rares', '30', 'NemoExperss', '0726437733');

commit;

--10-ADRESA

INSERT INTO "Adresa"

VALUES (7000, 'Dambovita', 'Targoviste', 'Cal.Domeneasca', 13, 'null', 130167);

INSERT INTO "Adresa"

```
VALUES (7001, 'Ilfov','Bucuresti', 'Str.Jiului', 45,'null',111357);

INSERT INTO "Adresa"

VALUES (7002, 'Cluj','Cluj-Napoca', 'Str.Prutului', 10,'E3',400058);

INSERT INTO "Adresa"

VALUES (7003, 'Timis','Timisoara', 'Str.Corvinilor', 143,'null',300081);

INSERT INTO "Adresa"

VALUES (7004, 'Dolj','Craiova', 'Cal.Voievozilor', 49,'A4',200122);

INSERT INTO "Adresa"

VALUES (7005, 'Constanta','Constanta', 'Str.Sighetului', 32,'null',900227);
```

```
commit;
```

```
--11-CLIENT
```

```
INSERT INTO "Client"

VALUES (8000, 'Codreanu','Radu', 'raducod@gmail.com', '0735947589','19-12-2000',7000);

INSERT INTO "Client"

VALUES (8001, 'Dovincescu','Adrian', 'dovadrian@gmail.com', '0705768589','29-04-1980',7001);

INSERT INTO "Client"

VALUES (8002, 'Rotaru','Mircea', 'mircear@gmail.com', '0732547569','14-11-2002',7002);

INSERT INTO "Client"

VALUES (8003, 'Dumitrache','Aurel', 'aurel11@yahoo.com', '0796447540','10-07-1992',7003);

INSERT INTO "Client"

VALUES (8004, 'Manea','Luca', 'luca231@gmail.com', '0745943581','30-03-1998',7004);
```

```
commit;
```

```
--12-COMANDA
```

```
INSERT INTO "Comanda"
```

```
VALUES (9000, '2181010250175', 'Nelivrat', '13-01-2022','19-01-2022',6000, 8000);
```

```
INSERT INTO "Comanda"
```

```
VALUES (9001, '1183210240435', 'Nelivrat', '15-01-2022','21-01-2022',6001, 8001);
```

```
INSERT INTO "Comanda"
```

```
VALUES (9002, '32143014355435', 'Livrat', '10-11-2021','12-11-2021',6002, 8002);
```

```
INSERT INTO "Comanda"
```

```
VALUES (9003, '613201435576', 'Nelivrat', '16-01-2022','22-01-2022',6003, 8003);
```

```
INSERT INTO "Comanda"
```

```
VALUES (9004, '7132010450438', 'Livrat', '01-06-2021','05-06-2021',6004, 8004);
```

```
INSERT INTO "Comanda"
```

```
VALUES (9005, '1133265450438', 'Livrat', '23-09-2021','29-09-2021',6003, 8004);
```

```
commit;
```

```
--13-FILMCOMANDA
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3000, 9000, 2);
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3001, 9001, 3);
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3002, 9002, 1);
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3003, 9003, 5);
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3004, 9004, 10);
```

```
INSERT INTO "FilmComanda"
```

```
VALUES (3005, 9005, 4);
```

```
commit;
```