COLEGIUL NAȚIONAL “GHEORGHE LAZĂR”

LUCRARE PENTRU OBȚINEREA ATESTATULUI PROFESIONAL LA INFORMATICĂ

**ADMINISTRAREA UNUI CINEMATOGRAF**

Elev: Profesor coordonator:

Ciocan Alexandra-Diana Dumitru Silviu

MAI 2021

CUPRINS

[1. Introducere 1](#_Toc72960435)

[*I.* *Baze de date: definiție, utilitate* 1](#_Toc72960436)

[*II.* *Sisteme de gesiune a bazelor de date* 1](#_Toc72960437)

[*III.* *Etape în realizarea unei baze de date* 2](#_Toc72960438)

[*IV.* *Entitate, atribut, relație, tipuri de relații* 2](#_Toc72960439)

[*V.* *Modele de baze de date* 3](#_Toc72960440)

[*VI.* *Reprezentarea modelului conceptual (ERD)* 4](#_Toc72960441)

[*VII.* *Normalizarea* 4](#_Toc72960442)

[*VIII.* *Maparea* 5](#_Toc72960443)

[*IX.* *Algebra relațională* 6](#_Toc72960444)

[*X.* *Limbajul SQL* 6](#_Toc72960445)

[2. Tema proiectului 9](#_Toc72960446)

[3. Descrierea proiectului 9](#_Toc72960447)

[*I.* *Prezentarea scenariului* 9](#_Toc72960448)

[*II.* *Prezentarea entităților, a atributelor, a relațiilor* 10](#_Toc72960449)

[*III.* *Reprezentarea modelului conceptual* 12](#_Toc72960450)

[*IV.* *Crearea tabelelor* 13](#_Toc72960451)

[4. Interogări 23](#_Toc72960452)

[5. Bibliografie 27](#_Toc72960453)

# Introducere

## *Baze de date: definiție, utilitate*

În lumea contemporană, progresul tehnologic a condus la amplificarea activităților legate de stocarea, interogarea și administrarea colecțiilor de date. Asfel, în activitățile noastre cotidiene interacționam des cu baze de date, în sitații precum: plata facturilor, împrumutul unei cărți de la bibliotecă, cumpărăturile online, tranzacțiile bancare.

Datele sunt informații neprelucrate, stocate în formă brută, în timp ce informațiile sunt date prelucrate, care au fost corectate, organizate, sortate, relaționate. Termenul “Database” a apărut prima dată în 1964, în titlul unei conferințe din Santa Monica, SUA: “Development and Management of Computer Centered DataBase”. O bază de date, abreviată BD, reprezintă o colecție de date aflate în relație unele cu altele și organizate astfel încat să poata servi unui anumit scop, prelucrarea rapidă și concomitentă a datelor de către mai multe persoane fiind facilitată. Prelucrarea poate consta în operații precum adăugare, ștergere, modificare, interogare.

Cele mai comune operații cu baze de date sunt:

* a proiecta o bază de date = a-i stabili structura: elementele componente, caracteristicile lor, restricțiile pe care trebuie să le respecte, relațiile dintre ele
* a construi o bază de date = a introduce datele în baza de date, după proiectarea acesteia
* a administra o bază de date = a asigura accesul la date al utilizatorilor (în funcție de drepturile fiecăruia), coerența bazei de date(recuperarea informației atunci când au loc incidente), securitatea datelor stocate
* a interoga o bază de date = a extrage și a vizualiza datele care îndeplinesc anumite criterii
* a actualiza o bază de date = a modifica structura/informațiile stocate în baza de date

## *Sisteme de gesiune a bazelor de date*

Bazele de date sunt manipulate cu ajutorul sistemelor de gestiune a bazelor de date. Astfel, un sistem de gestiune a bazelor de date, abreviat SGBD, este un pachet software care permite proiectarea, construirea și administrarea bazelor de date dedicate rezolvării problemelor din domenii variate ale vieții reale. Acesta poate fi convențional (informațiile sunt stocate pe foi de hârtie) sau computerizat (informațiile sunt păstrate în fișiere).

Informaticienii pot clasifica sistemele de gestiune a bazelor de date în funcție de modelele de baze de date pe care le acceptă. Bazele de date relaționale au devenit dominante în anii 1980. Acestea modelează datele ca rânduri și coloane într-o serie de tabele, iar marea majoritate folosesc SQL pentru scrierea și interogarea datelor. În anii 2000, bazele de date non-relaționale au devenit populare, denumite NoSQL, deoarece folosesc diferite limbaje de interogare.

În cazul bazelor de date relaţionale mari şi foarte mari, s-au impus SGBD-uri precum Oracle, DB2 şi Informix. Acestea au la bază tehnologia client-server. Oracle este un sistem interactiv, computerizat, produs și comercializat de Oracle Corporation. Acesta este utilizat în mod obișnuit pentru rularea proceselor de tranzacție online, depozitarea datelor și sarcini de lucru mixte.

## *Etape în realizarea unei baze de date*

Realizarea unei baze de date se desfășoară în mai multe etape, cele mai importante fiind:

1. Analiza

În etapa de analiză se examinează sistematic și aprofundat aspectul lumii reale care trebuie transpus într-o bază de date, urmând să se stabilească gradul de generalitate, dimensiunea bazei de date, categoriile și numărul de viitori utilizatori. O bază de date se realizează la cererea proprietarului și trebuie să conducă la facilitarea activității din organizația respectivă.

1. Proiectarea

Etapa de proiectare se poate descompune în trei subetape: proiectarea bazei de date la nivel conceptual, la nivel logic și la nivel fizic. În această etapă se poate selecta sistemul de gestiune a bazei de date, se proiectează interfețele și aplicațiile de utilizare și administrare a bazei de date. Principalele scopuri ale acestei etape sunt reprezentarea datelor și a relațiilor dintre date, furnizarea unui model de date care să asigure orice tip de prelucrare a datelor și schițarea unui proiect structurat astfel încat să satisfacă parametrii de eficiență specificați.

1. Implementarea

Etapa de implementare constă în realizarea propriu-zisă a bazei de date: scrierea programelor, conversia și încărcarea datelor.

## *Entitate, atribut, relație, tipuri de relații*

O entitate este un obiect, un eveniment sau o activitate din lumea reală care are semnificație pentru afacerea modelată. În procesul de realizare a unei baze de date, prin entitate înțelegem mulțimea obiectelor de un anumit tip, iar prin instanță înțelegem un anume element, bine individualizat, unic, din mulțimea elementelor care formează o anumită entitate, în timp ce un atribut este o caracteristică a unei entitați. Atributele unei entități descriu, cuantifică, clasifică sau specifică. O relație într-o bază de date este o legătură logică între două sau mai multe entități. Acestea pot fi:

1-1 (one-to-one) = o relație între două entități E1 și E2 în care unei instanțe a E1 îi corespunde o singură intanță din E2 și reciproc

1-m (one-to-many) = o relație între două entități E1 și E2 în care unei instanțe a E1 îi pot corespunde mai multe intanțe din entitatea E2, dar unei intanțe din E2 nu îi poate corespunde decât cel mult o instanță din E1

m-m (many-to-many) = o relație între două entități E1 și E2 în care unei instanțe a E1 îi pot corespunde mai multe intanțe din entitatea E2, și, reciproc, unei instanțe a E2 îi pot corespunde mai multe intanțe din entitatea E1

## *Modele de baze de date*

Un model este un ansamblu format dintr-o colecție de concepte necesare pentru descrierea structurii bazei de date (a tipurilor de date și a relațiilor dintre ele și a restricțiilor) și un set de operații de bază (care să specifice modul de efectuare a extragerii și actualizării datelor din baza de date).

Cele mai cunoscute modele de baze de date sunt următoarele:

* modelul tabelar – toate datele sunt memorate sub forma unui singur tabel, un tablou bidimensional de date
* modelul ierarhic – datele sunt organizate sub forma unor structuri arborescente, există deci o rădăcină cu mai mulţi dependenţi, care la rândul lor pot avea alţi dependenţi
* modelul reţea – un model performant, dar complicat (o bază de date de tip reţea reprezintă o colecţie de noduri şi legături, fiecare nod putând fi legat de oricare altulș legăturile trebuie stabilite având tot timpul în minte interogările posibile şi acţiunile viitoare probabile)
* modelul relaţional – cel mai utilizat model de stocare a datelor, în care datele sunt organizate sub formă de tabele între care există diverse legături
* modelul obiectual – destinat să suporte modele de obiecte complexe; este oarecum asemănător reţelei, iar prin faptul că pentru accesare directă, stochează o hartă a ierarhiilor şi relaţiilor claselor de obiecte, are ascendent şi în modelul ierarhic
* modelele hibride – mixturi ale modelelor prezentate anterior, din care cel mai semnificativ este modelul relaţional-obiectual, obţinut prin extensii ale modelului de organizare tabelar şi izvorât din tendinţa spre universalitate a bazei de date (entităţi complexe şi de naturi diferite, evoluând în condiţii eterogene).

Considerat cel mai important eveniment din istoria bazelor de date, apariția modelului relațional are loc în iunie 1970. Într-un model relațional, datele sunt percepute de utilizatori ca niște tabele, iar operațiile disponibile pentru utilizatori sunt operații care generează noi tabele pe baza tabelelor vechi.

## *Reprezentarea modelului conceptual (ERD)*

Primul pas în realizarea unei aplicaţii de baze de date este analiza datelor şi realizarea unei scheme conceptuale (model conceptual) ale acestora. În această etapă sunt analizate natura şi modul de utilizare a datelor. Sunt identificate datele care vor trebui memorate şi procesate, se împart în grupuri logice şi se identifică relaţiile care există între aceste grupuri.

O diagramă a relației entității (ERD - entity relationship diagram) arată relațiile dintre entitățile stocate într-o bază de date. Prin definirea entităților, atributelor lor și arătarea relațiilor dintre ele, o diagramă ER ilustrează structura logică a bazelor de date. Diagramele ERD sunt utilizate pentru a schița proiectarea unei baze de date.

## *Normalizarea*

Normalizarea este o tehnică de proiectare a bazelor de date prin care se elimină/se evită anumite anomalii şi inconsistenţe ale datelor. O bază de date bine proiectată nu permite ca datele să fie redundante, adică aceeaşi informaţie să se găsească în locuri diferite sau să se memoreze în baza de date informaţii care se pot deduce pe baza altor informaţii memorate în baza de date. Anomaliile care pot să apară la o bază de date nenormalizată sunt următoarele:

* anomalii la actualizarea datelor
* anomalii de inserare
* anomalii de ştergere

Edgar Codd a definit primele trei forme normale 1NF, 2NF şi 3NF. Ulterior, s-au mai definit formele normale 4NF, 5NF şi 6NF care însă sunt rar folosite în proiectarea bazelor de date.

* Prima formă normală (1NF)

O entitate se găseşte în prima formă normală dacă şi numai dacă nu există atribute cu valori multiple sau nu există atribute sau grupuri de atribute care se repetă. Cu alte cuvinte, toate atributele trebuie să fie atomice, adică să conţină o singură informaţie.

Dacă un atribut are valori multiple, sau un grup de atribute se repetă, atunci trebuie creată o entitate suplimentară, urmând să fie legată de entitatea originală printr-o relaţie de 1-m. În noua entitate vor fi introduse atributele sau grupurile de atribute care se repetă.

* A doua formă normală (2NF)

O entitate se găseşte în a doua formă normală dacă şi numai dacă se găseşte în prima formă normală şi în plus, orice atribut care nu face parte din UID (Unique IDentifier) va depinde de întregul UID, nu doar de o parte a acestuia.

O situaţie mai specială este în cazul relaţiilor barate, când trebuie ţinut seama că UID-ul unei entităţi este compus din atribute din entitatea respectivă, plus un atribut sau mai multe provenite din relaţia barată.

De asemenea, dacă o entitate se găseşte în prima formă normală şi UID-ul său este format dintr-un singur atribut, atunci ea se găseşte automat în a doua formă normală.

* A treia formă normală (3NF)

O entitate se găseşte în a treia formă normală dacă şi numai dacă se găseşte în a doua formă normală şi în plus niciun atribut care nu este parte a UID-ului nu depinde de un alt atribut non-UID. Cu alte cuvinte, nu se acceptă dependenţe tranzitive, adică un atribut să depindă de UID în mod indirect.

## *Maparea*

Transformarea modelului conceptual, a ERD-ului, în modelul fizic, adică în baza de date propriu-zisă, se numeşte mapare. Acest proces implică transformarea fiecărui element al ERD-ului.

Prima etapă a acestui proces constă în crearea tabelelor bazei de date. Astfel:

* fiecărei entităţi îi va corespunde câte un tabel, iar spre deosebire de entitate, un tabel va avea numele un substantiv la plural
* fiecare atribut al unei entităţi va deveni o coloană a tabelei (fiecare coloană va memora date de acelaşi tip)
* fiecare instanţă a unei entităţi se va transforma într-un rând (sau înregistrare) al tabelului corespunzător
* unicul identificator al entităţii devine cheia primară a tabelei (coloana sau combinaţia de coloane care identifică în mod unic toate liniile unui tabel se numeşte cheie primară)

În general, la maparea unei relaţii de tip 1-m vom introduce, în tabela corespunzătoare entităţii de pe partea many a relaţiei, cheia primară a entităţii de pe partea one a relaţiei. Câmpurile astfel introduse se vor numi chei străine (în engleză, foreign key). Aşadar, cheia străină a unei tabele este cheia primară din tabela referinţă și este întotdeauna introdusă în tabela corespunzătoare entităţii din partea many a relaţiei.

Pentru maparea relațiilor 1-1, dându-se două entităţi A şi B legate între ele prin acest tip de relație, putem include cheia primară a lui A în cadrul tabelei B, dar putem proceda la fel de bine şi invers, incluzând cheia primară a tabelei B în cadrul tabelei A, deoarece fiecărei instanţe a entităţii A îi corespunde cel mult o instanţă a entităţii B, dar şi invers, oricărei instanţe a entităţii B îi corespunde cel mult o instanţă a entităţii A. Decizia depinde de specificul afacerii modelate, uneori este convenabil să memorăm cheia străină în ambele părţi ale relaţiei.

Relaţiile barate se transformǎ în urma mapǎrii în cheie străină în tabela aflată în partea many a relaţiei, la fel ca la maparea oricărei relaţii one-to-many. Bara de pe relaţie exprimă faptul că acele coloane ce fac parte din cheia străină vor deveni parte a cheii primare a tabelei din partea many a relaţiei barate.

## *Algebra relațională*

Bazele de date relaţionale au fost dezvoltate având în vedere în primul rând utilizatorii finali. Acest model are la bază teoria matematică a relaţiilor, ceea ce a făcut posibilă tratarea algoritmică a proiectării bazelor de date şi problema normalizării datelor.

Modelul relaţional este un model simplu, bazat pe algebra relaţională, care a făcut posibilă dezvoltarea limbajelor relaţionale sub forma unui software specializat ce asistă procesul de implementare a bazelor de date. Astfel de limbaje sunt SQL-ul (Structured Query Language) şi QBE (Query By Example).

Algebra relaţională oferă un ansamblu de operatori permiţând operarea asupra conceptelormodelului relaţional. Rezultatul obţinut prin utilizarea operatorilor algebrei relaţionale este un tabel virtual . Având în vedere fundamentele matematice ale modelului relaţional, algebrarelaţională utilizează operatori clasici de manipulare a ansamblurilor de date (reuniune, intersecţie, diferenţă, produs cartezian) şi introduce operatori proprii bazelor de date (selecţie, proiecţie, compunere, diviziune).

## *Limbajul SQL*

În practica creării şi utilizării bazelor de date relaţionale s-a impus necesitatea existenţei unui limbaj standard care permite efectuarea acestor operaţii. Astfel, a apărut SQL - Structured Query Language. Limbajul este supervizat de comisia de standardizare ANSI (American National Standards Institute), motiv pentru care se mai numeşte şi ANSI SQL. SQL nu este un limbaj de firmă, el este implementat de o mulţime de SGBD-uri, de această dată consacrate, cum ar fi: Microsoft Access, Oracle, Microsoft SQL Server şi MySQL.

Limbajul SQL – Structured Query Language este un limbaj standard de interogare a bazelor de date. Implementat în majoritatea SGBD-urilor, dar nu numai. In ORACLE este implementat nucleul SQL cu anumite particularitati.SQL utilizează o sintaxă simplă, uşor de învăţat şi utilizat.

Limbajul SQL lucrează cu următoarele tipuri de date:

VARCHAR2(n) – șir de caractere cu lungimea variabilă, de maxim n caractere

CHAR(n) – șir de caractere de lungime fixă, având n caractere

NUMBER(p,s) – număr cu precizia p și scala s

LONG – șiruri de caractere cu lungime variabilă, nu pot ocupa mai mult de 2GB

DATE – date calendaristice valide

BFILE – un locatar către un fișier binar de dimensiune maximă 4GB, stocat înafara bazei de date

O expresie este formată din variabile, constante, operatori şi funcţii. Operatorii pot fi:

* Operatori aritmetici

Operatorii aritmetici permişi în SQL sunt cei patru operatori din matematică: adunare +, scădere -, înmulţire \*, împărţire /. Ordinea de efectuare a operaţiilor aritmetice este cea din matematică (mai întâi înmulţirea şi împărţirea şi apoi adunarea şi scăderea).

* Operatori alfanumerici

Există un singur operator alfanumeric şi anume operatorul de concatenare a două şiruri || (două bare verticale fără spaţii între ele).

De exemplu, expresia 'abc'||'xyz' are valoarea 'abcxyz'.

* Operatori de comparaţie

Pe lângă operatorii obişnuiţi de comparaţie: <, >, <=, >=, <> sau != (pentru diferit), =, SQL mai implementează următorii operatori speciali:

LIKE – despre care vom discuta puţin mai târziu în acest capitol;

Interogări simple. Sortarea datelor 115

BETWEEN – testează dacă o valoare se găseşte într-un interval definit de două valori. Astfel, expresia x BETWEEN a AND b este echivalentă cu expresia (x>=a) AND (x<=b)

IN – testează dacă o valoare aparţine unei mulţimi de valori specificate. De exemplu, expresia x IN (a,b,c) este echivalentă cu (x=a) OR (x=b) OR (x=c)

IS NULL şi IS NOT NULL – se folosesc pentru a testa dacă o expresie are valoarea NULL sau nu. Comparaţia cu NULL nu se poate face folosind operatorii obişnuiţi = şi respectiv <>.

* Operatori logici

În ordinea priorităţii lor, aceştia sunt:

NOT – negaţia logică

AND – şi logic, expresia a AND b este adevărată dacă şi numai dacă ambii operanzi a şi b au valoarea adevărat.

OR – sau logic, expresia a OR b este adevărată dacă şi numai dacă cel puţin unul dintre operanzii a şi b au valoarea adevărat.

Comenzile SQL pot fi grupate în cinci categorii, după cum urmează:

* Limbajul de interogare permite regăsirea liniilor memorate în tabelele bazei de date. Vom scrie interogări folosind comanda SELECT.
* Limbajul de manipulare a datelor (DML - Data Manipulation Language) permite modificarea conţinutului tabelelor. Există următoarele comenzi DML:

INSERT - pentru adăugarea de noi linii într-o tabelă

UPDATE - pentru modificarea valorilor memorate într-o tabelă

DELETE - pentru ştergerea liniilor dintr-o tabelă.

* Limbajul de definire a datelor (DDL - Data Definition Language) vă permite să definiţi structura tabelelor care compun baza de date. Comenzile din această grupă sunt:

CREATE - vă permite să creaţi structurile bazei de date. De exemplu, CREATE TABLE este utilizată pentru crearea tabelelor, cu CREATE USER, puteţi crea utilizatorii bazei de date, etc.

ALTER - permite modificarea structurilor bazei de date. De exemplu, cu comanda ALTER TABLE puteţi modifica structura unei tabele.

DROP - puteţi şterge structuri ale bazei de date. De exemplu, pentru a şterge o tabelă, folosiţi comanda DROP TABLE.

RENAME - puteţi schimba numele unei tabele.

TRUNCATE - vă permite să ştergeţi întregul conţinut al unei tabele.

* Comenzi de control al tranzacţiilor (TC - Transaction Control):

COMMIT - vă permite să faceţi ca modificările asupra bazei de date să devină permanente.

ROLLBACK - permite renunţarea la ultimele modificări asupra bazei de date.

Interogări simple. Sortarea datelor 109

SAVEPOINT – vă permite să definiţi un "punct de salvare" la care să puteţi reveni, renunţând la modificările făcute după acel punct asupra bazei de date.

* Limbaj de control al datelor (DCL - Data Control Language) Permite definirea şi modificarea drepturilor utilizatorilor asupra bazei de date. Există două comenzi în această categorie:

GRANT - vă permite să acordaţi drepturi altor utilizatori asupra structurilor bazei voastre de date.

REVOKE - puteţi să anulaţi anumite drepturi utilizatorilor bazei de date.

# Tema proiectului

Ca temă a proiectului am ales administrarea unui cinematograf. Cinematograful este un loc destinat exclusiv proiecțiilor de filme și reprezintă pentru mulți oameni un loc preferat pentru petrecerea timpului liber. Chiar și pentru persoanele care nu se consideră cinefile, vizionarea unui film este activitatea perfectă pentru relaxare. În ultima perioadă serviciile de streaming au devenit tot mai populare și multe persoane aleg să vizioneze filme din confortul propriei locuințe, însă cinematograful rămâne locul care oferă o atmosferă mult mai favorabilă și o experiență mult mai captivantă și autentică de vizionare. Astfel, fiind una dintre persoanele care apreciază profund arta cinematografică și preferă experiența cinematografică, am decis ca aceasta să fie tema proiectului.

# Descrierea proiectului

## *Prezentarea scenariului*

Trebuie proiectată baza de date a unui cinematograf. Baza de date trebuie să conțină informații despre filmele proiectate, clienți, achizițiile acestora și angajați, care se împart în casieri și manageri, fiecare manager coordonând mai mulți angajați. Fiecare film este difuzat după un program în una din sălile cinematografului. Trebuie ținută evidența clară a biletelor achiziționate.

## *Prezentarea entităților, a atributelor, a relațiilor*

După analiza scenariului, am identificat următoarele entități, relevante pentru realizarea unei baze de date coerente:

CLIENT – toate persoanele care se înscriu în baza de date, despre care se știu lucruri precum preferințele legate de filme, fie că au facut vreodată o achiziție sau urmează să facă

ACHIZIȚIE – indică toate achizițiile făcute de clienți

CASIER – prezintă toți angajații care se ocupă de vânzarea biletelor și detalii despre aceștia precum datele personale (nume, prenume, CNP), dar și programul după care lucrează aceștia

MANAGER – toți angajații care se ocupă cu coordonarea casierilor

BILET – biletele, atat vândute cât și disponibile, indentificabile în funcție de un cod

FILM – reprezintă toate filmele care se proiectează la cinematograf și diverse detalii cu ajutorul cărora se poate deduce audiența țintă pentru fiecare (gen, regizor, producător)

LINIE PROGRAM – indică orele la care se difuzează anumite filme în anumite săli

SALĂ – entitate care privește toate sălile disponibile în cinematograf, impreună cu detalii despre ele precum capacitate (număr de locuri) și unde pot fi găsite (etaj)

Fiecare entitate identificată prezintă urmatoarele atribute:

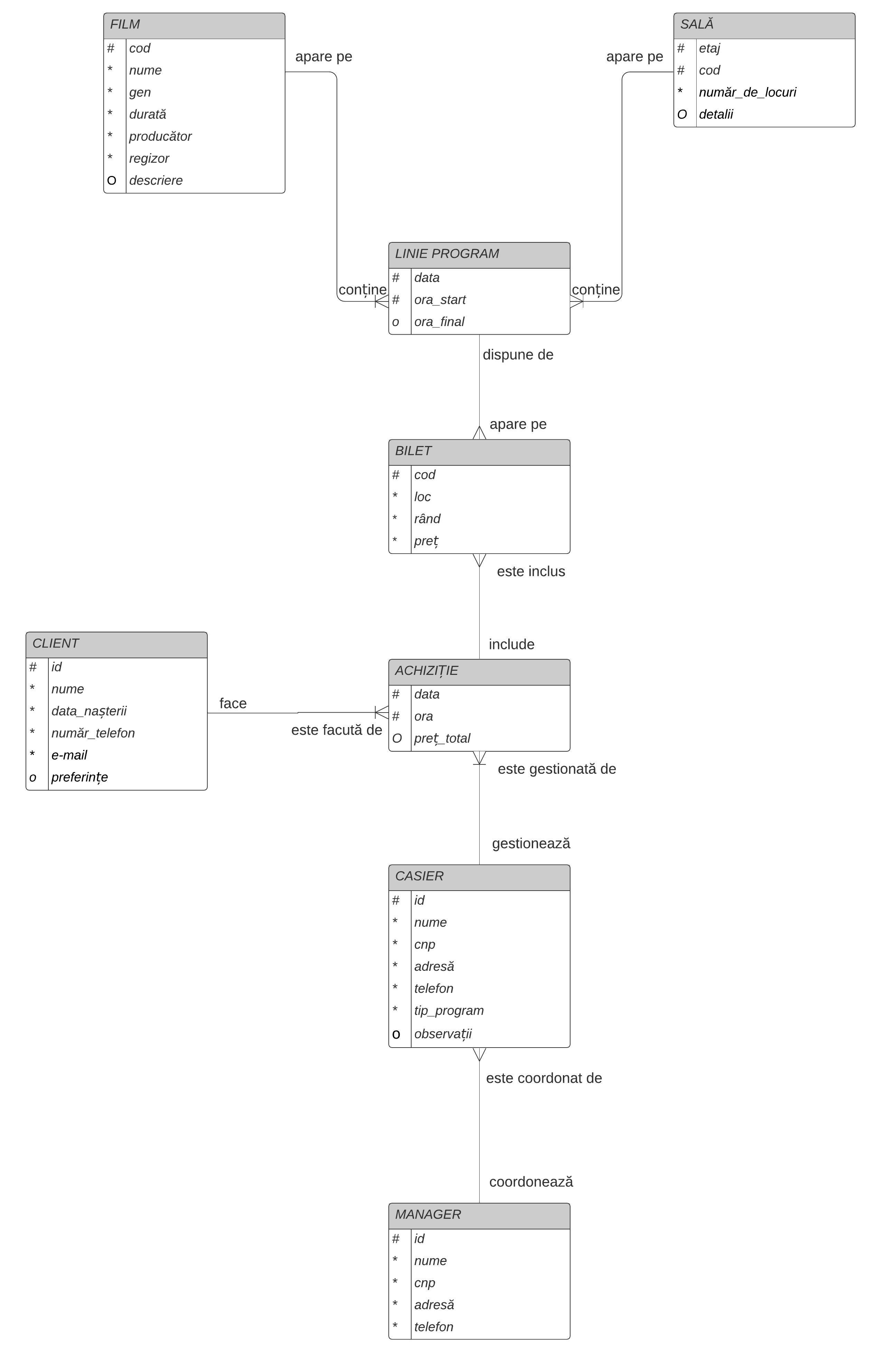
|  |  |
| --- | --- |
| CLIENT | #id  \*nume  \*data\_nașterii  \*număr\_telefon  \*e-mail  O preferințe |
| ACHIZIȚIE | #data  #ora  \*preț\_total |
| CASIER | #id  \*nume  \*cnp  \*adresă  \*telefon  \*tip\_program  O observații |
| MANAGER | #id  \*nume  \*cnp  \*adresă  \*telefon |
| BILET | #cod  \*loc  \*rând  \*preț |
| FILM | #data  #ora\_start  O ora\_final |
| LINIE PROGRAM | #cod  \*nume  \*gen  \*durată  \*producător  \*regizor |
| SALĂ | #etaj  #cod  \*număr\_de\_locuri  O detalii |

Între aceste entități se stabilesc următoarele relații:

|  |  |  |
| --- | --- | --- |
| Entități | Tip de relație | Descriere |
| FILM – LINIE PROGRAM | 1-M | Un film proiectat în cadrul cinematografului este proiectat în mod repetitiv pe o perioadă de timp, deci va apărea pe mai multe linii de program |
| SALĂ – LINIE PROGRAM | 1-M | O sală este spațiul de proiectare pentru mai multe filme în perioade orare diferite, deci apare pe mai multe linii de program |
| LINIE PROGRAM – BILET | 1-M | Linia de program indică perioada de proiectare a unui film într-o sală cu numeroase locuri, deci pentru fiecare linie de program există un număr mai mare de bilete |
| BILET – ACHIZIȚIE | 1-M | O achiziție poate consta în mai multe bilete cumpărate, însă fiecare bilet poate fi achiziționat o singură dată |
| CLIENT – ACHIZIȚIE | 1-M | Un client poate efectua mai multe achiziții pe parcursul timpului, însă o achiziție individuală se face pe numele unui singur client |
| CASIER - ACHIZIȚIE | 1-M | Un casier gestionează numeroase achiziții, însă pentru gestionarea unei achiziții este nevoie de un singur casier |
| CASIER – MANAGER | 1-M | Fiecare casier e coordonat de un manager, iar fiecare manager coordonează mai mulți casieri |

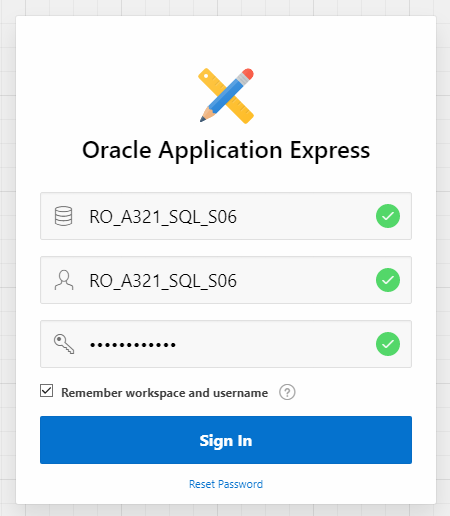
## *Reprezentarea modelului conceptual*

În continuare, am folosit platforma [www.lucidchart.com](http://www.lucidchart.com) pentru a realiza diagrama ERD:



## *Crearea tabelelor*

Pentru crearea tabelelor a fost nevoie de conexiune la Internet pentru a putea accesa adresa <https://iacademy.oracle.com>.



* Crearea tabelului CLIENTI, corespunzător entității CLIENT

CREATE TABLE "CLIENTI"

( "ID" NUMBER,

"NUME" VARCHAR2(30) NOT NULL ENABLE,

"DATA\_NASTERII" DATE NOT NULL ENABLE,

"NUMAR\_TELEFON" VARCHAR2(10) NOT NULL ENABLE,

"EMAIL" VARCHAR2(30) NOT NULL ENABLE,

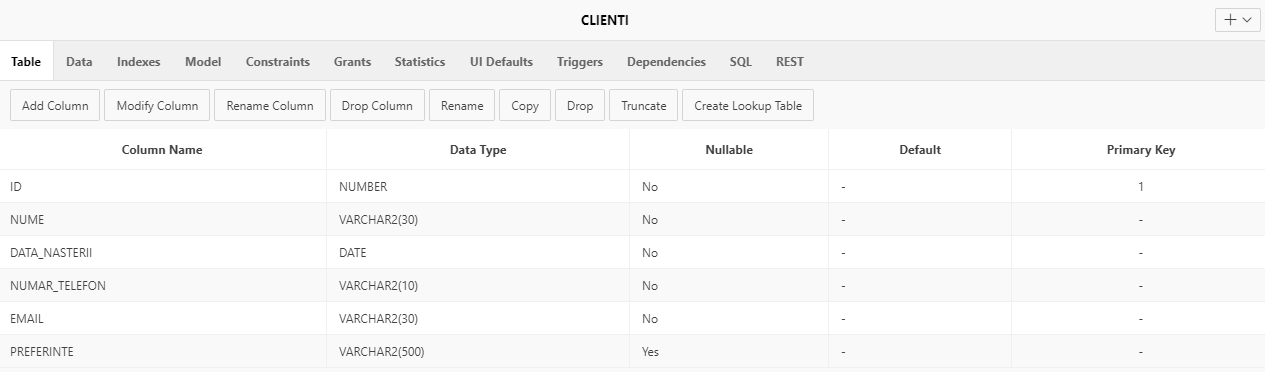
"PREFERINTE" VARCHAR2(500),

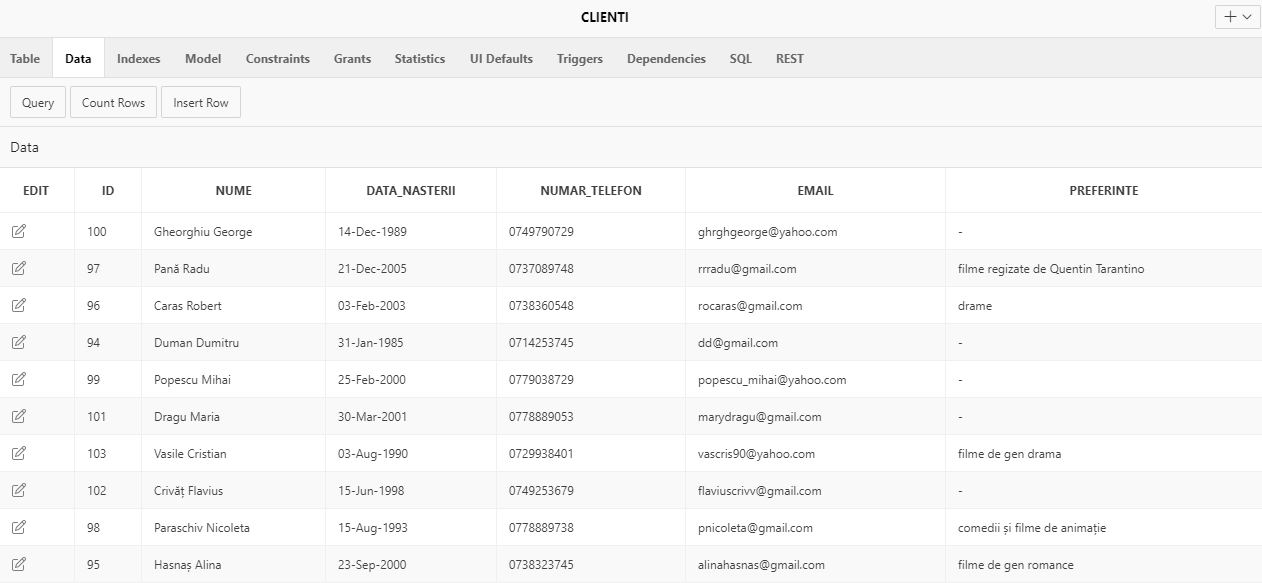
PRIMARY KEY ("ID")

USING INDEX ENABLE

)

/





* Crearea tabelului ACHIZITII, corespunzător entității ACHIZIȚIE

CREATE TABLE "ACHIZITII"

( "DATA" DATE NOT NULL ENABLE,

"ORA" VARCHAR2(5) NOT NULL ENABLE,

"PRET\_TOTAL" NUMBER NOT NULL ENABLE,

"ID\_CASIER" NUMBER NOT NULL ENABLE,

"ID\_CLIENT" NUMBER NOT NULL ENABLE,

PRIMARY KEY ("DATA", "ORA")

USING INDEX ENABLE

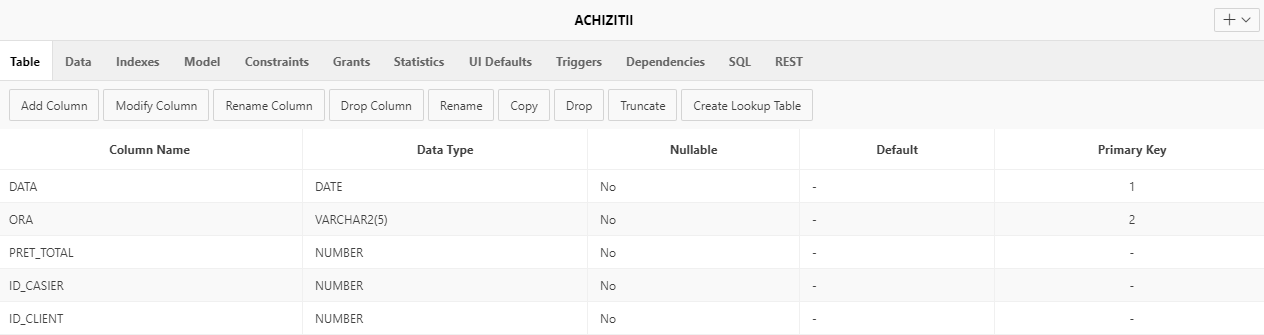
)

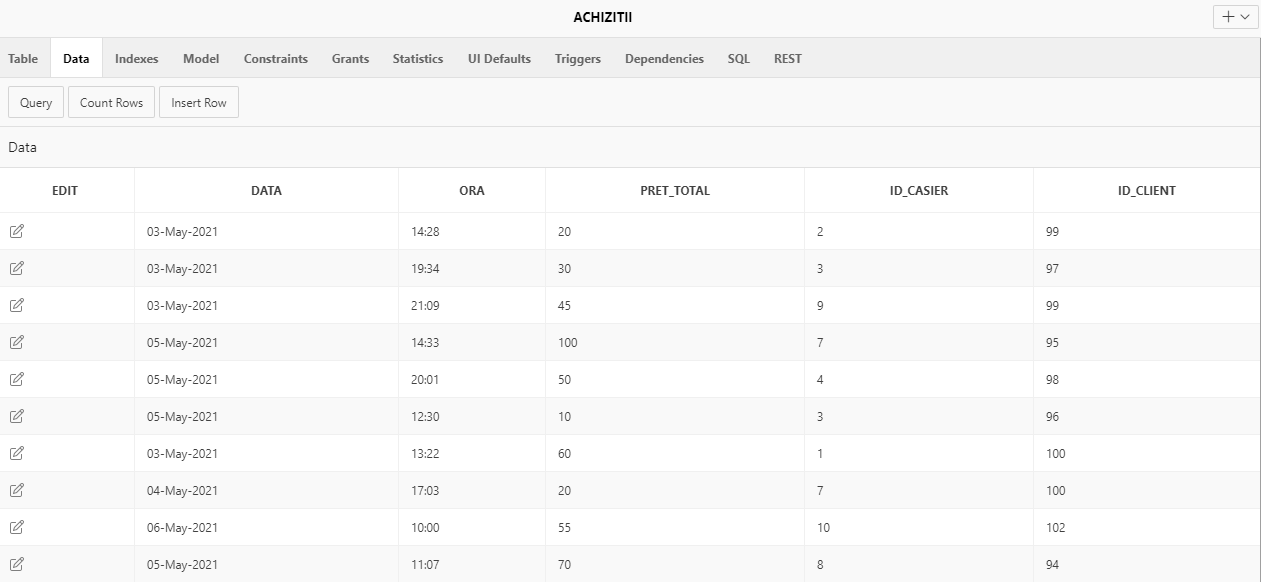
/

ALTER TABLE "ACHIZITII" ADD FOREIGN KEY ("ID\_CLIENT")

REFERENCES "CLIENTI" ("ID") ENABLE

/





* Crearea tabelului CASIERI, corespunzător entității CASIER

CREATE TABLE "CASIERI"

( "ID" NUMBER,

"ID\_MANAGER" NUMBER NOT NULL ENABLE,

"NUME" VARCHAR2(30) NOT NULL ENABLE,

"CNP" VARCHAR2(13) NOT NULL ENABLE,

"ADRESA" VARCHAR2(150) NOT NULL ENABLE,

"TELEFON" VARCHAR2(10) NOT NULL ENABLE,

"TIP\_PROGRAM" VARCHAR2(10) NOT NULL ENABLE,

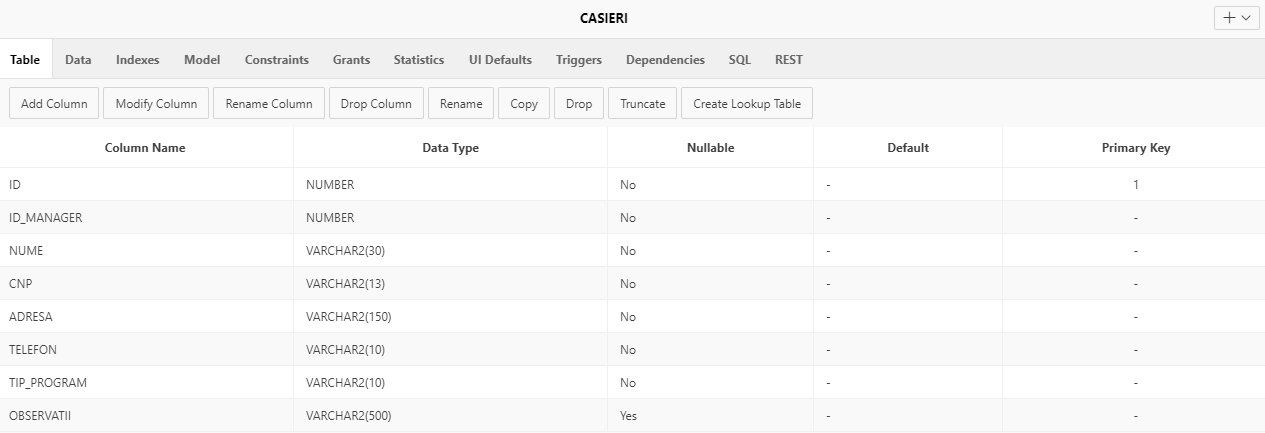
"OBSERVATII" VARCHAR2(500),

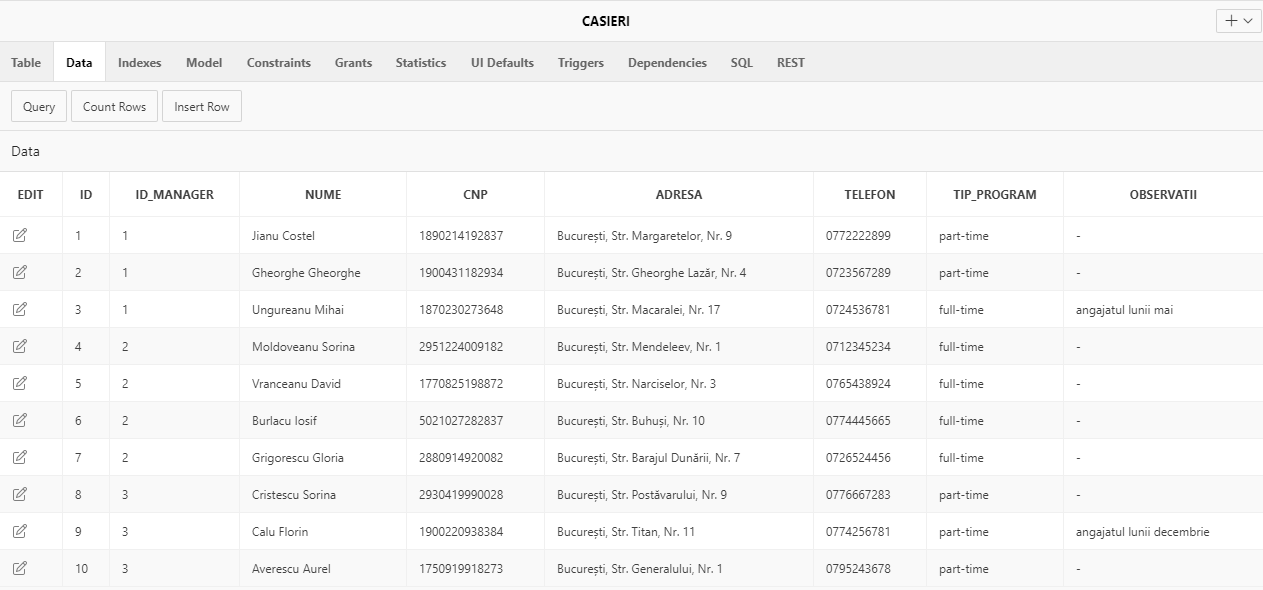
PRIMARY KEY ("ID")

USING INDEX ENABLE

)

/





* Crearea tabelului MANAGERI, corespunzător entității MANAGER

CREATE TABLE "MANAGERI"

( "ID" NUMBER,

"NUME" VARCHAR2(30) NOT NULL ENABLE,

"CNP" VARCHAR2(13) NOT NULL ENABLE,

"ADRESA" VARCHAR2(150) NOT NULL ENABLE,

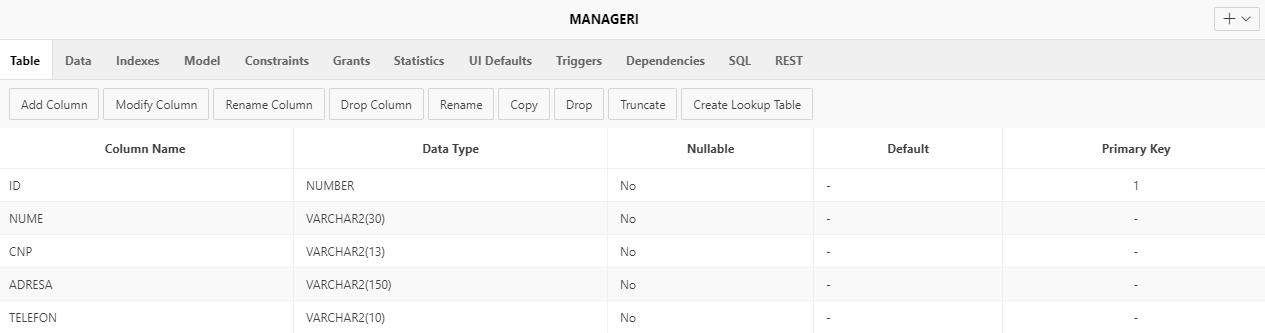
"TELEFON" VARCHAR2(10) NOT NULL ENABLE,

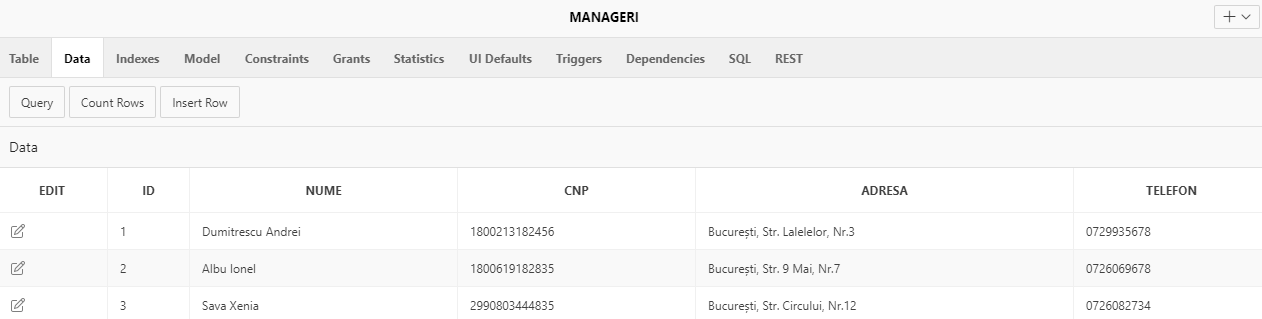
PRIMARY KEY ("ID")

USING INDEX ENABLE

)

/





* Crearea tabelului BILETE, corespunzător entității BILET

CREATE TABLE "BILETE"

( "COD" NUMBER,

"LOC" NUMBER NOT NULL ENABLE,

"RAND" NUMBER NOT NULL ENABLE,

"PRET" NUMBER NOT NULL ENABLE,

"COD\_SALA" NUMBER NOT NULL ENABLE,

"DATA\_PROGRAM" DATE NOT NULL ENABLE,

"ORA\_PROGRAM" VARCHAR2(5) NOT NULL ENABLE,

"COD\_FILM" NUMBER NOT NULL ENABLE,

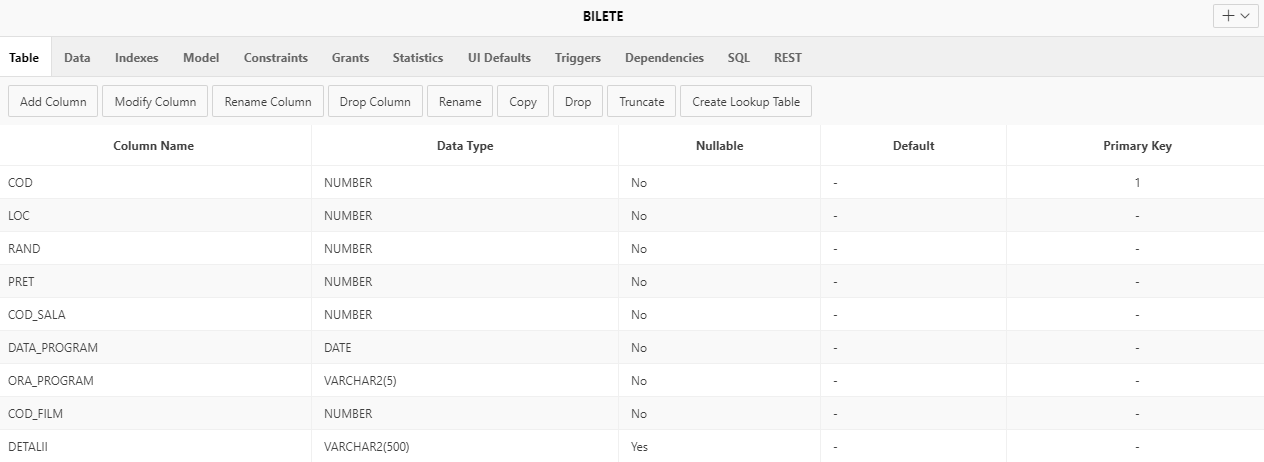
"DETALII" VARCHAR2(500),

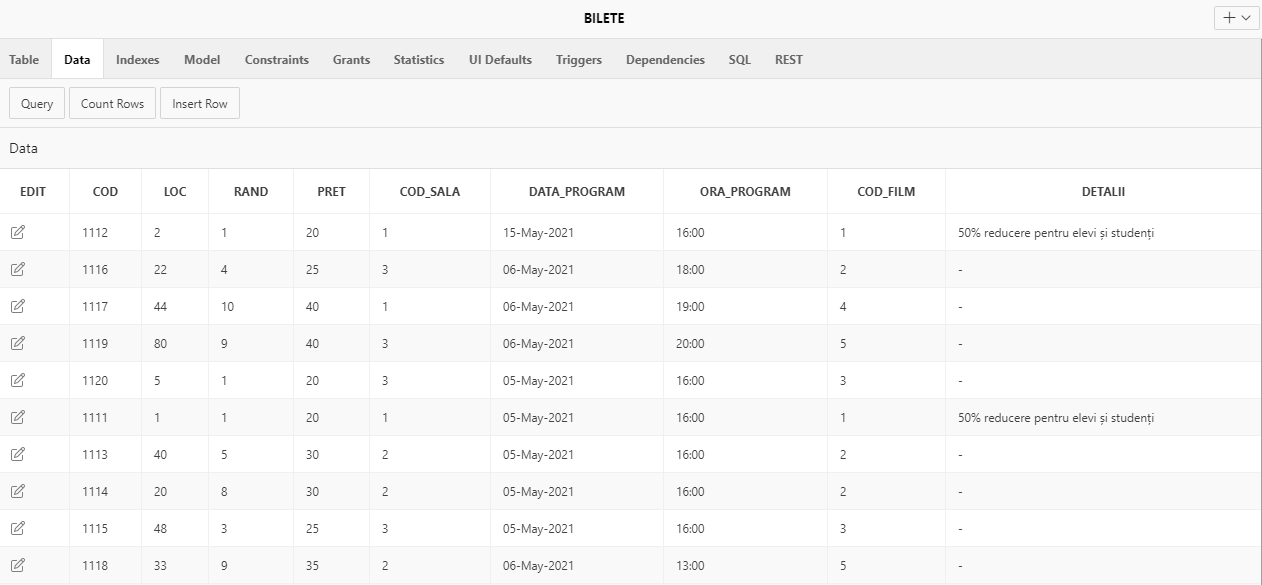
PRIMARY KEY ("COD")

USING INDEX ENABLE

)

/





* Crearea tabelului FILME, corespunzător entității FILM

CREATE TABLE "FILME"

( "COD" NUMBER,

"NUME" VARCHAR2(50) NOT NULL ENABLE,

"GEN" VARCHAR2(20) NOT NULL ENABLE,

"DURATA" VARCHAR2(10) NOT NULL ENABLE,

"PRODUCATOR" VARCHAR2(20) NOT NULL ENABLE,

"REGIZOR" VARCHAR2(20) NOT NULL ENABLE,

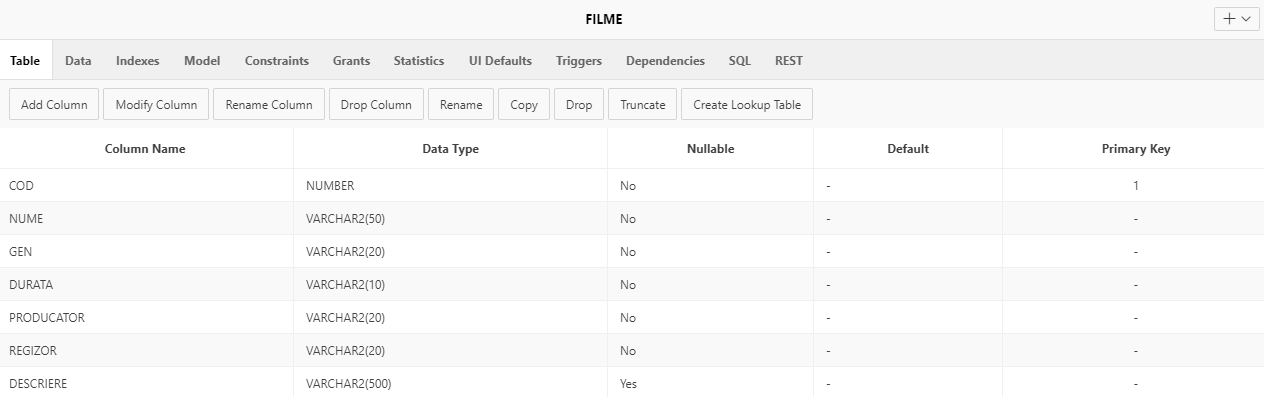
"DESCRIERE" VARCHAR2(500),

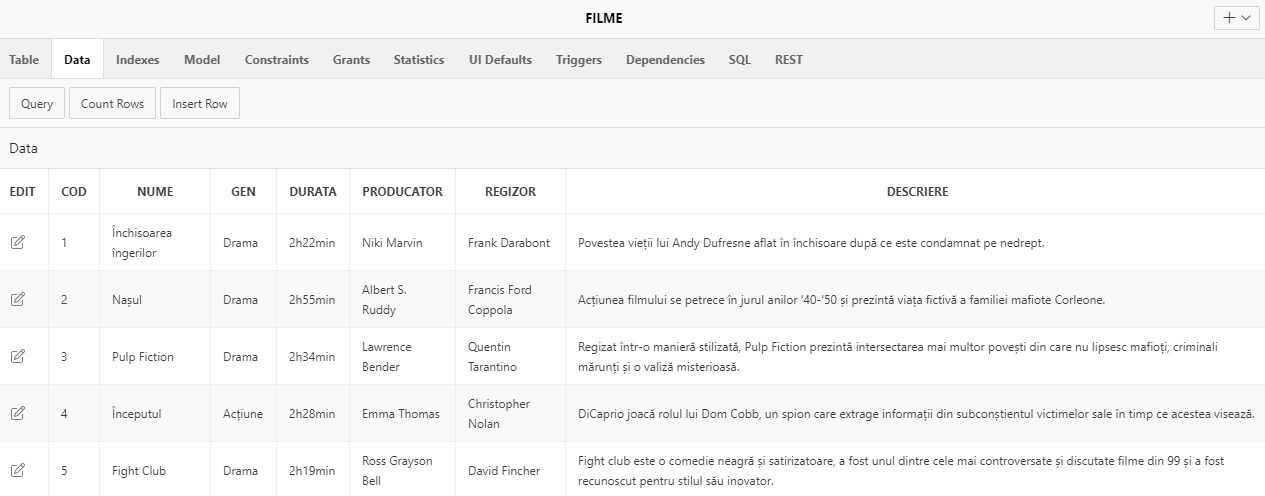
PRIMARY KEY ("COD")

USING INDEX ENABLE

)

/





* Crearea tabelului LINII\_PROGRAM, corespunzător entității LINIE PROGRAM

CREATE TABLE "LINII\_PROGRAM"

( "DATA" DATE NOT NULL ENABLE,

"ORA\_START" VARCHAR2(5) NOT NULL ENABLE,

"COD\_FILM" NUMBER NOT NULL ENABLE,

"ETAJ\_SALA" NUMBER NOT NULL ENABLE,

"COD\_SALA" NUMBER NOT NULL ENABLE,

"ORA\_FINAL" VARCHAR2(5) NOT NULL ENABLE,

PRIMARY KEY ("DATA", "ORA\_START", "COD\_FILM", "ETAJ\_SALA", "COD\_SALA")

USING INDEX ENABLE

)

/

ALTER TABLE "LINII\_PROGRAM" ADD FOREIGN KEY ("ETAJ\_SALA", "COD\_SALA")

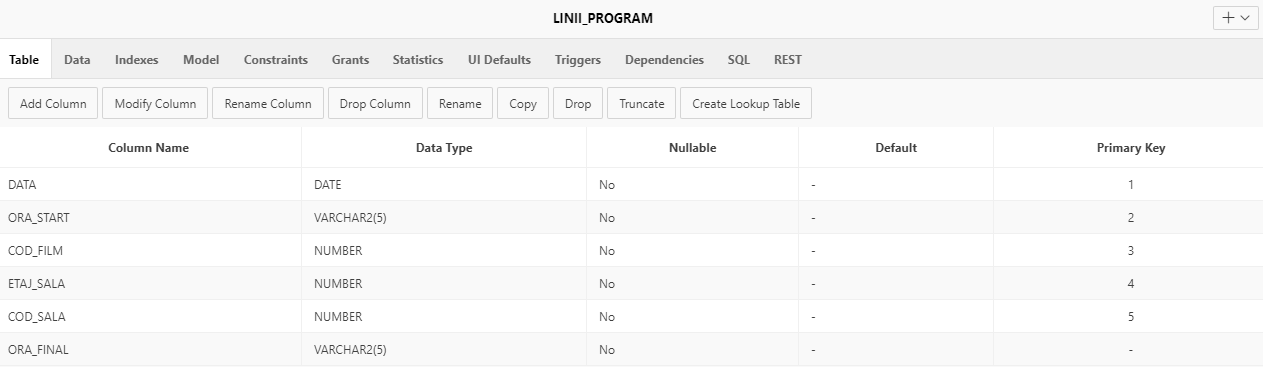
REFERENCES "SALI" ("ETAJ", "COD") ENABLE

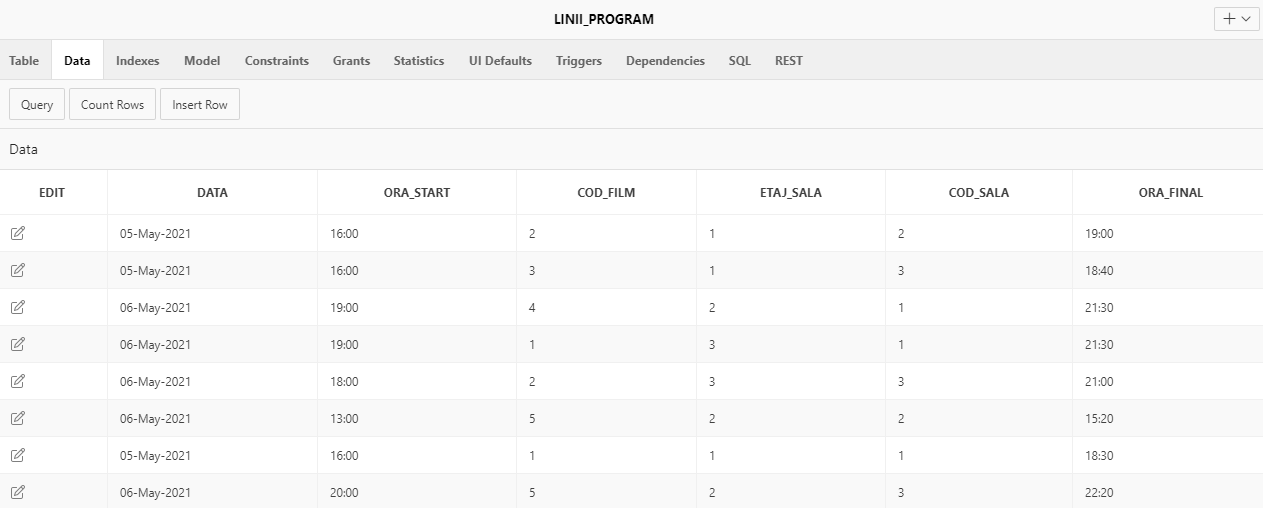
/

ALTER TABLE "LINII\_PROGRAM" ADD FOREIGN KEY ("COD\_FILM")

REFERENCES "FILME" ("COD") ENABLE

/





* Crearea tabelului SALI, corespunzător entității SALĂ

CREATE TABLE "SALI"

( "ETAJ" NUMBER NOT NULL ENABLE,

"COD" NUMBER NOT NULL ENABLE,

"NUMAR\_DE\_LOCURI" NUMBER NOT NULL ENABLE,

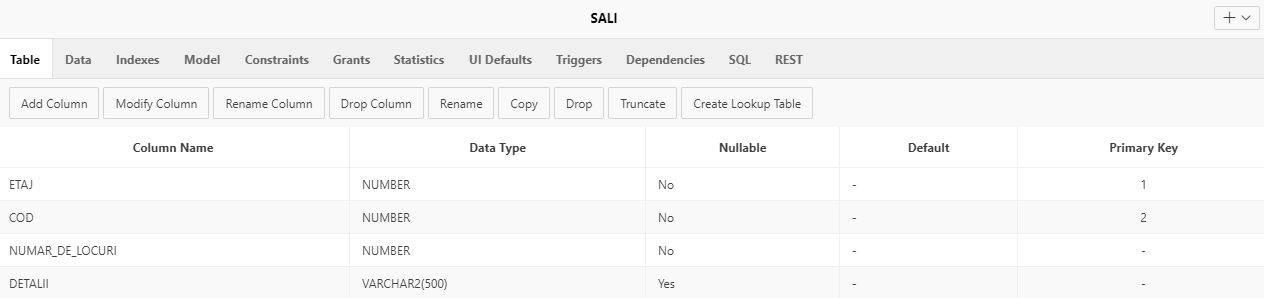
"DETALII" VARCHAR2(500),

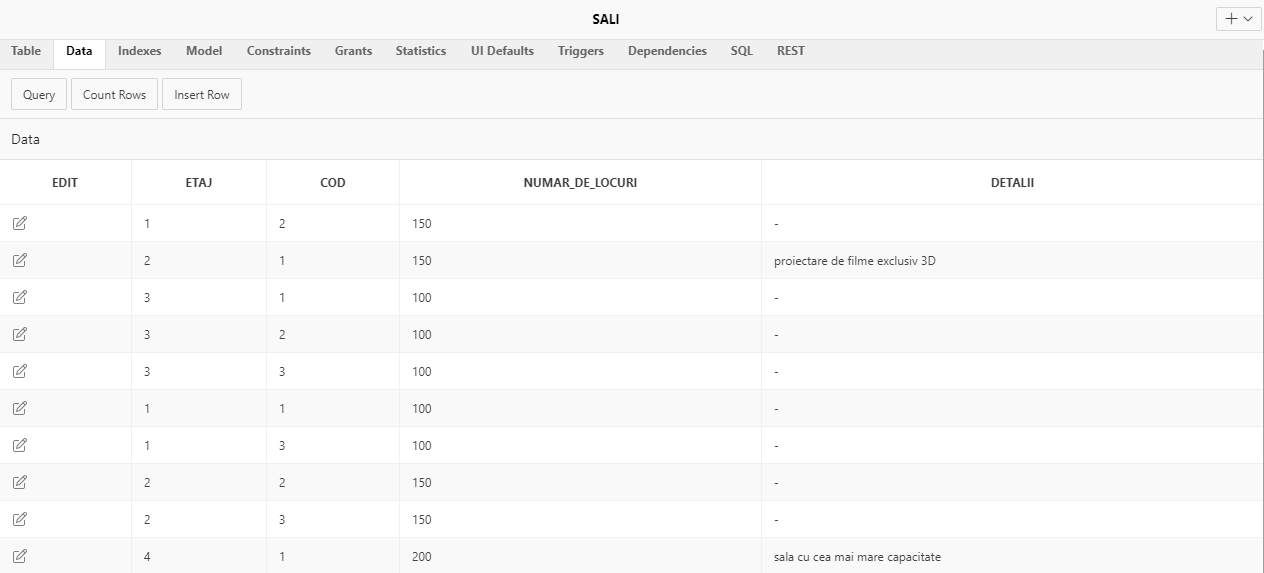
PRIMARY KEY ("ETAJ", "COD")

USING INDEX ENABLE

)

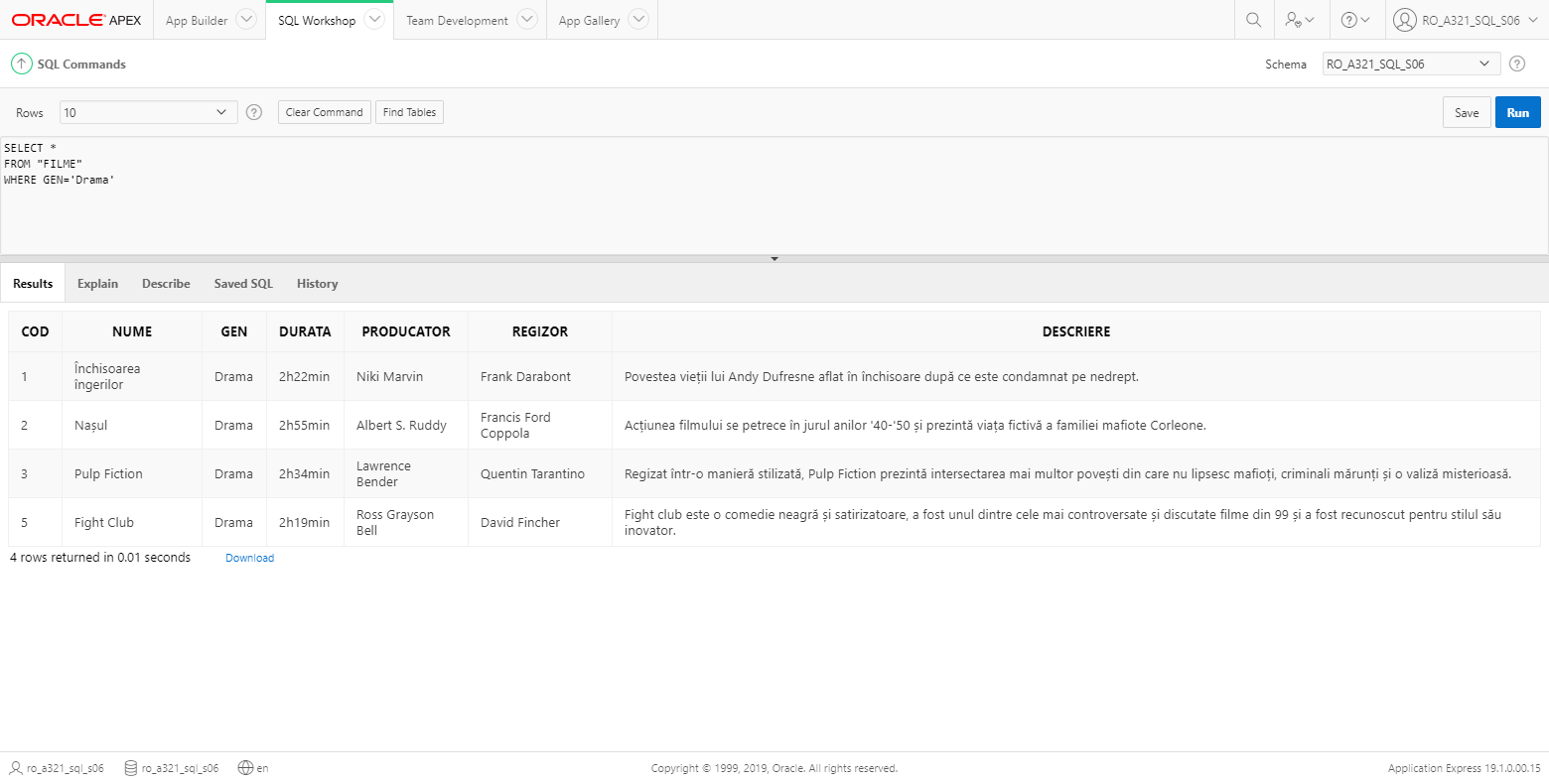
/



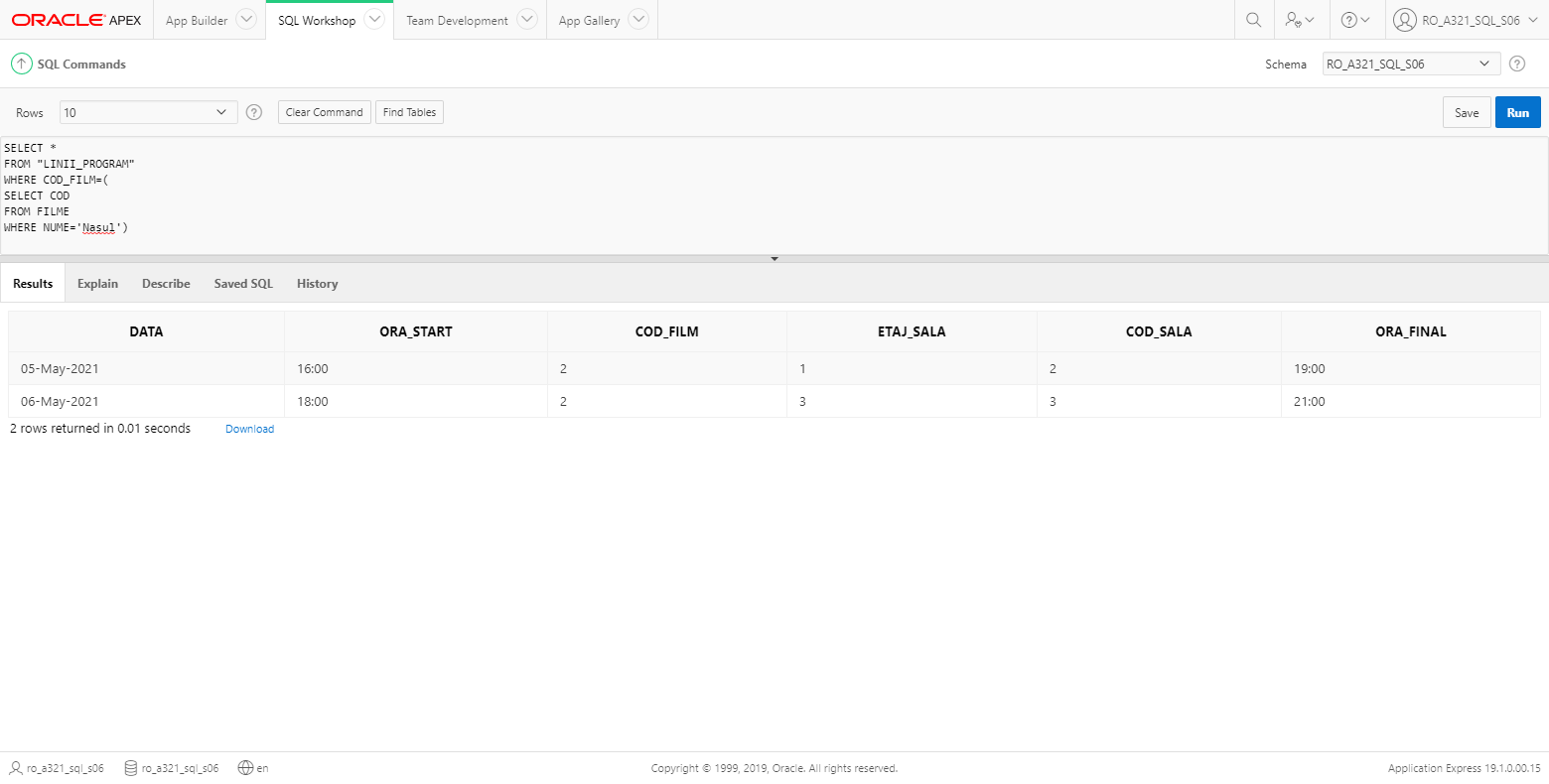


# Interogări

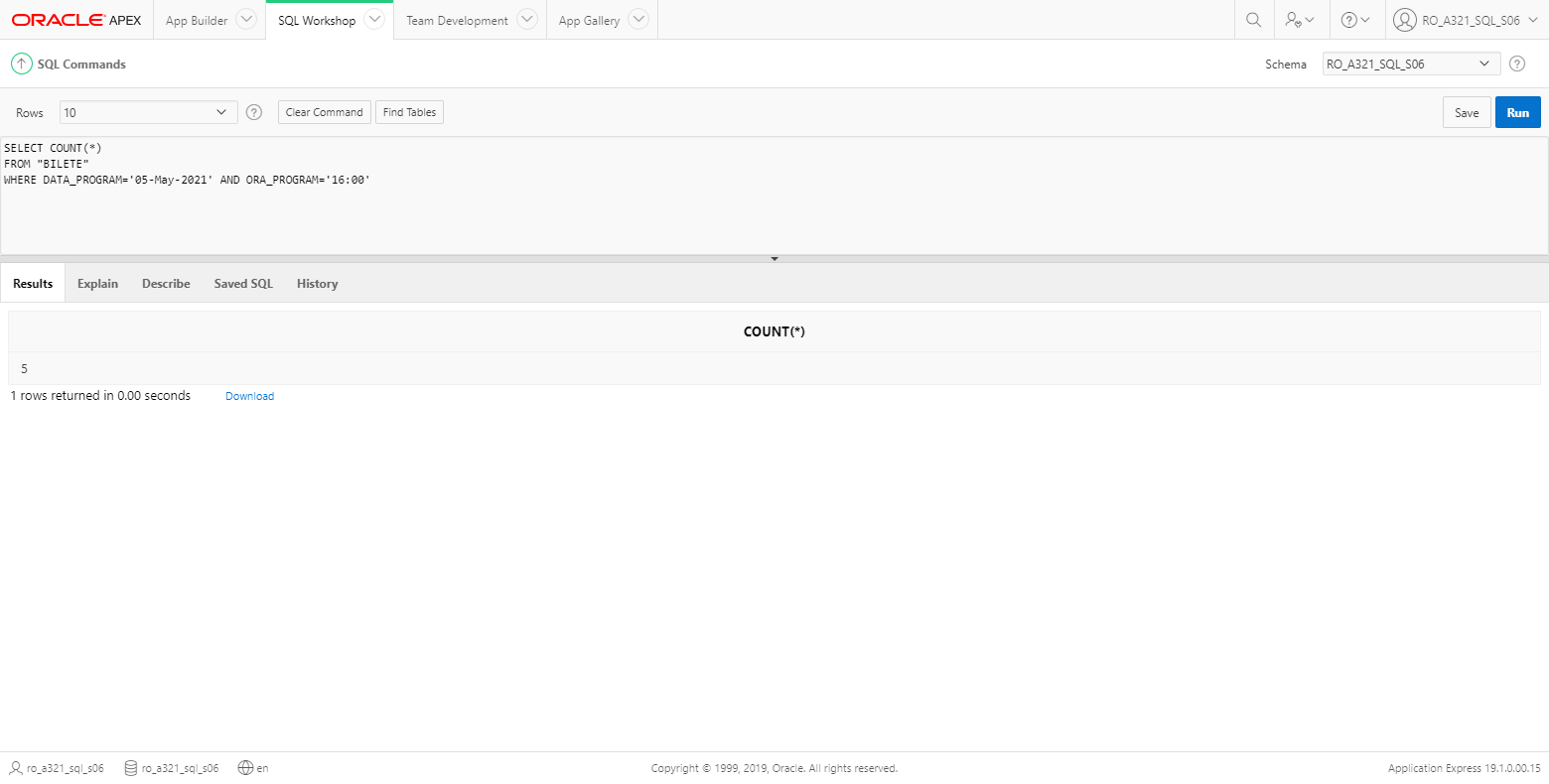
1. Ce filme de gen dramă se proiectează la cinematograf?



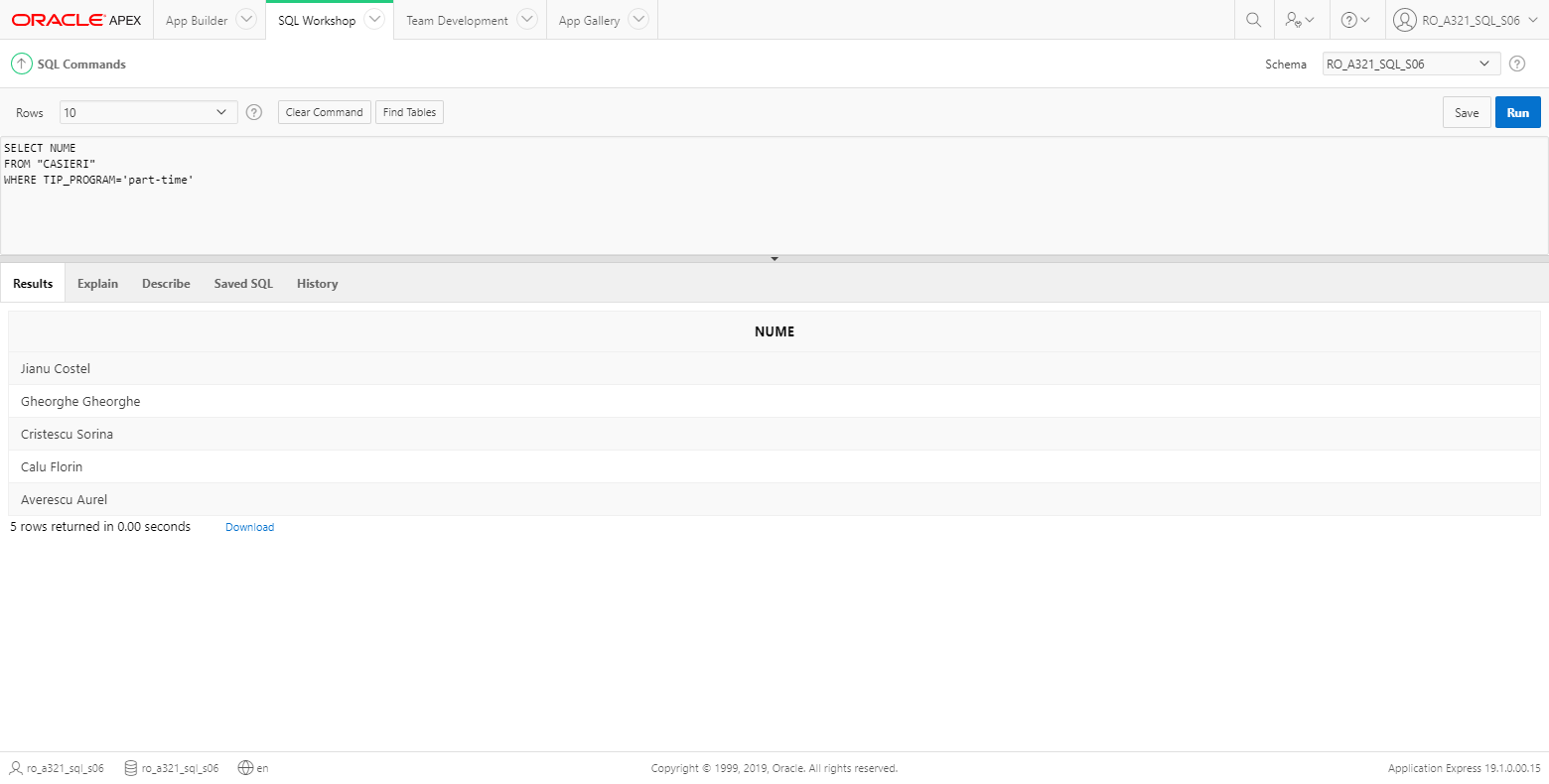
1. Când se proiectează filmul “Nașul”?



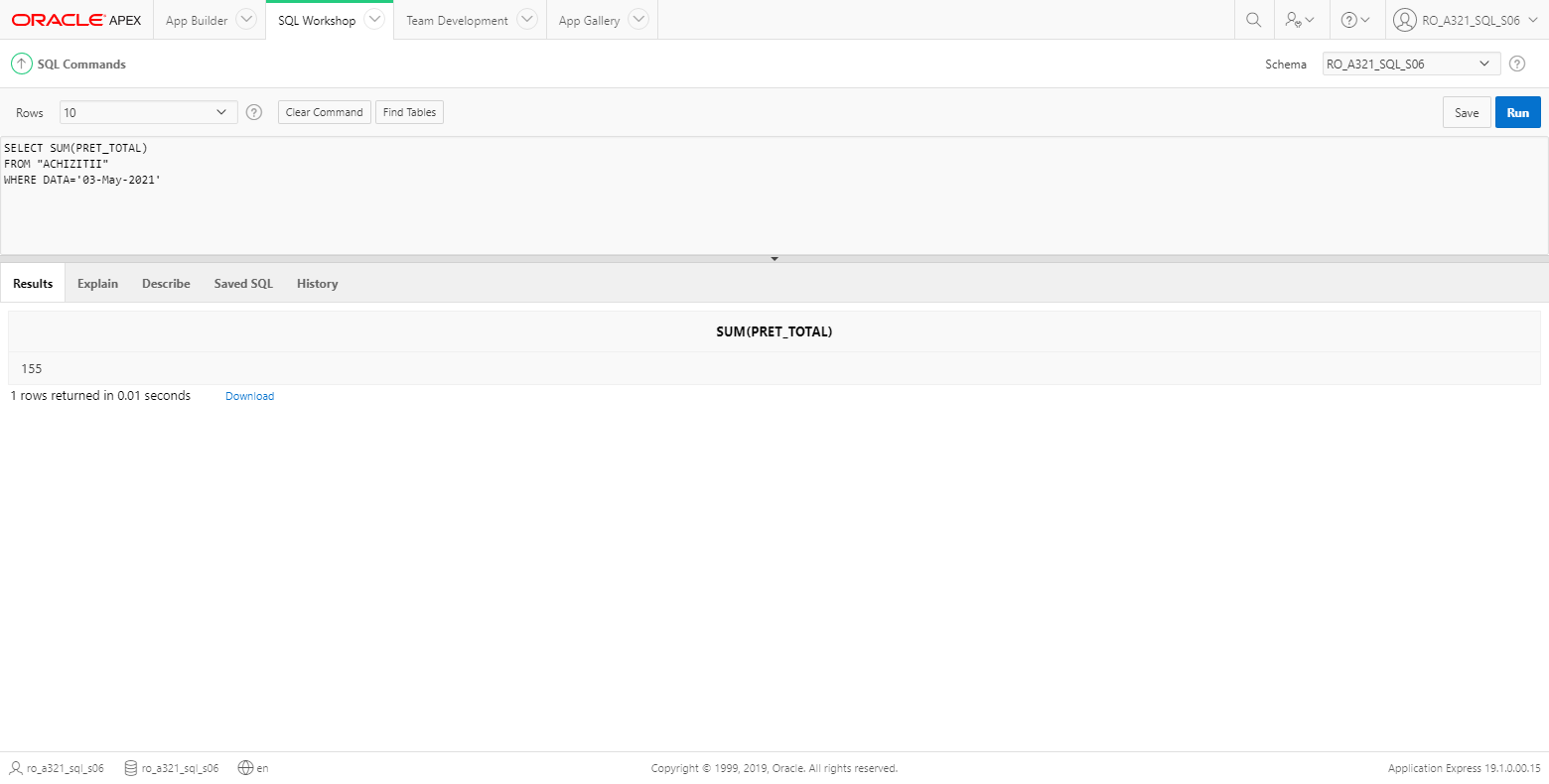
1. Câte bilete sunt disponibile pentru proiecții la ora 16:00 în data de 5 mai 2021?



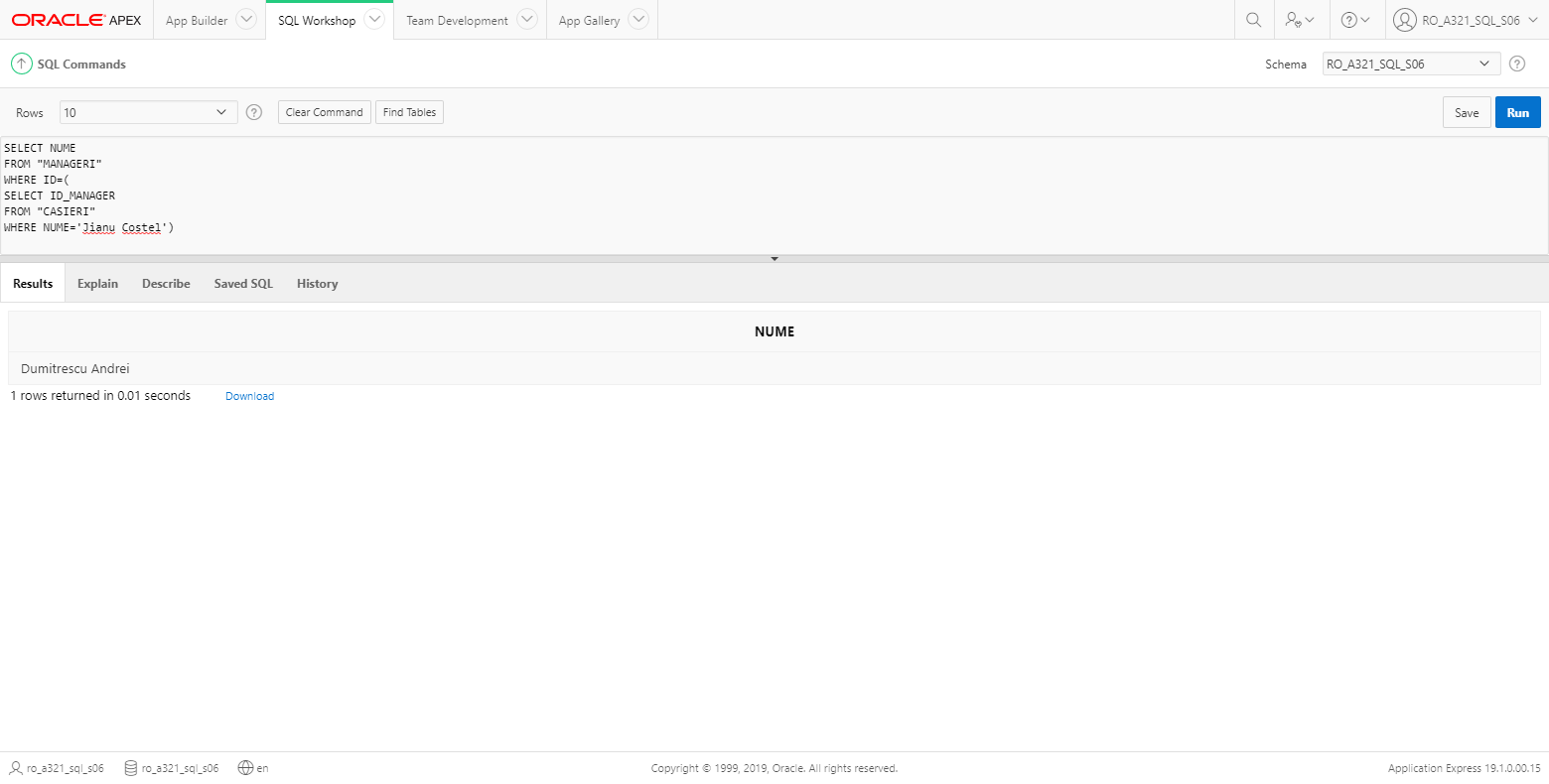
1. Care sunt angajații care lucrează part-time?



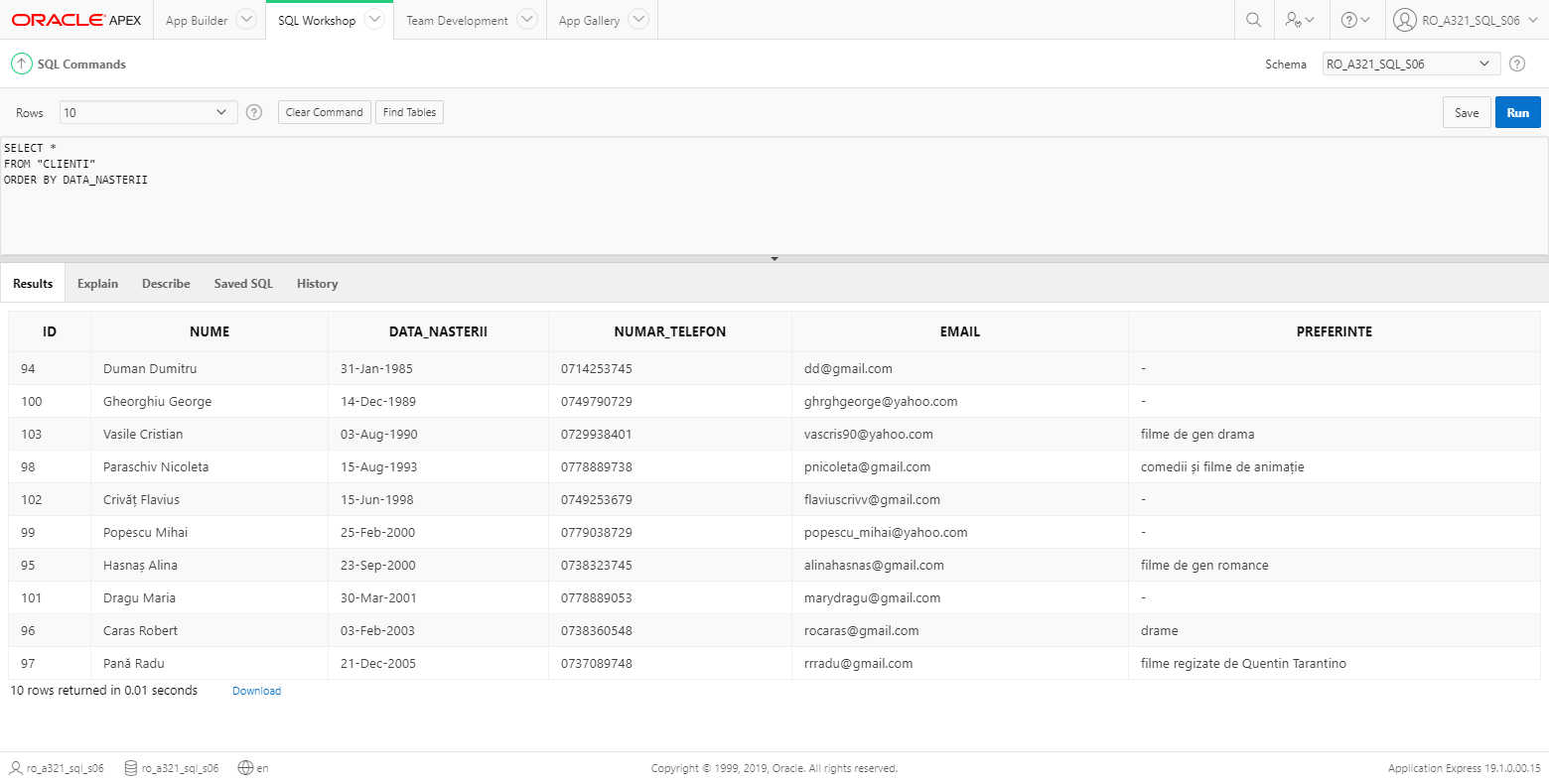
1. Care e suma totală încasată în data de 3 mai 2021?



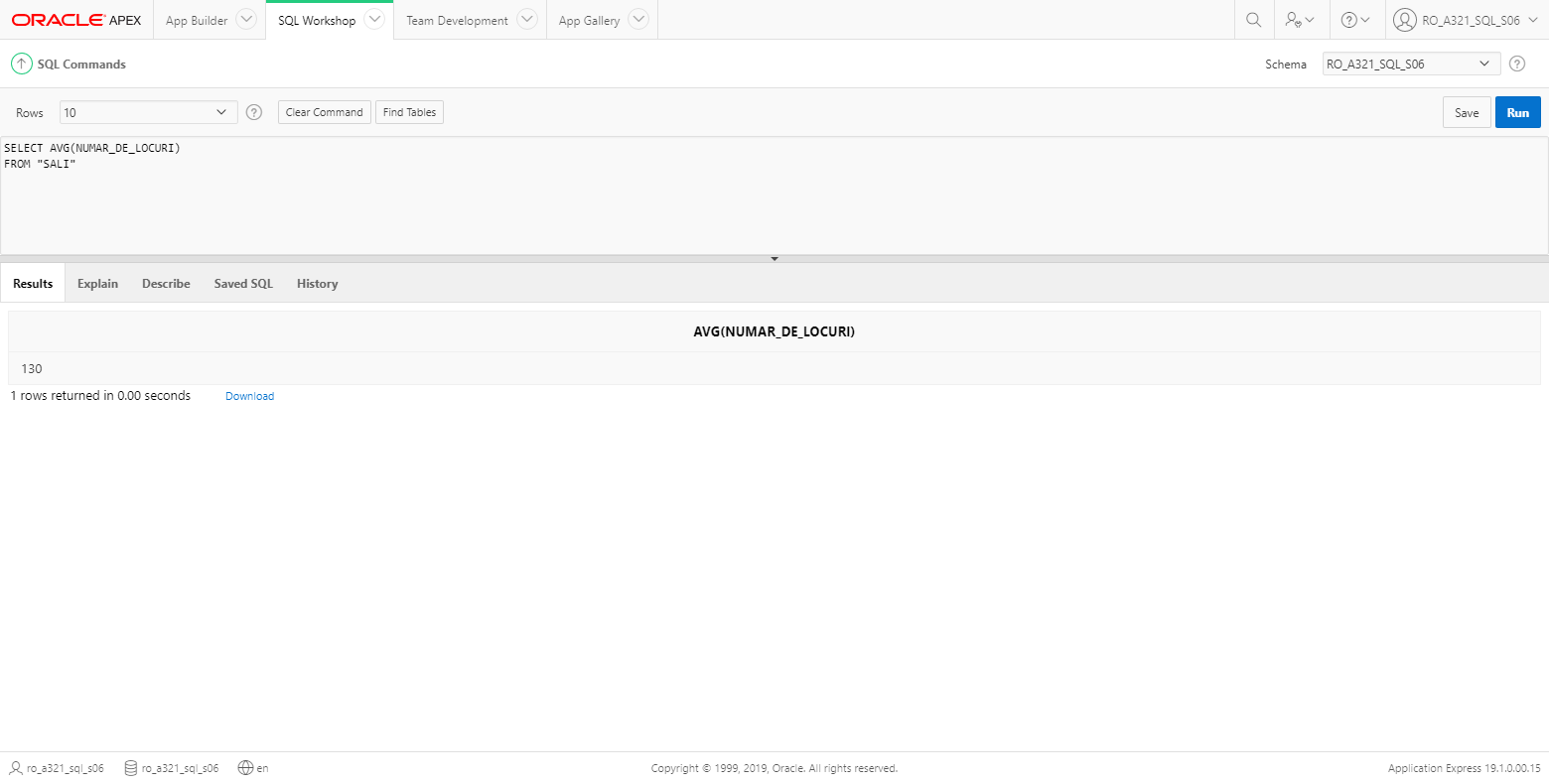
1. Ce manager îl coordonează pe Jianu Costel?



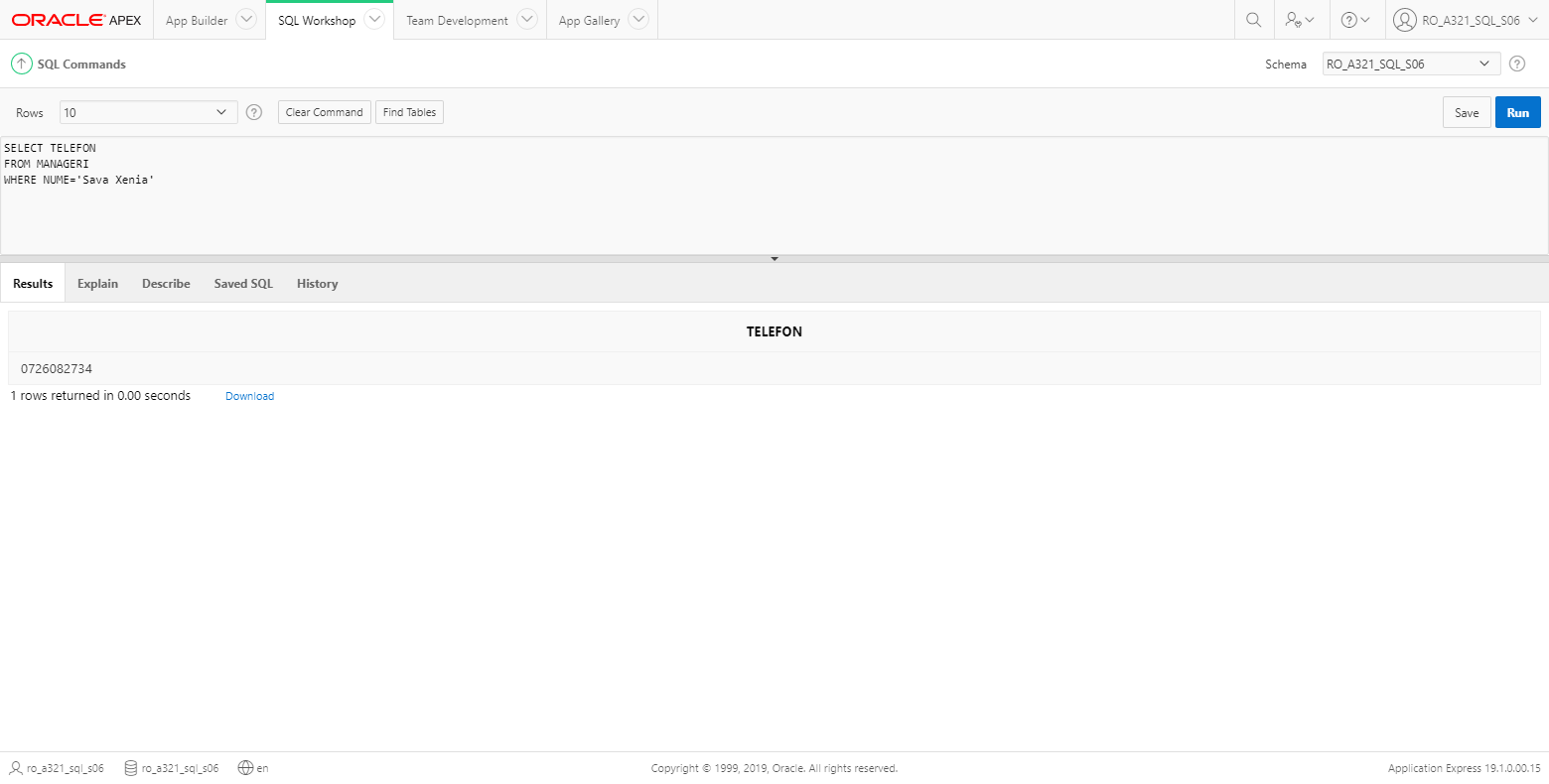
1. Cum arată lista clienților ordonați în funcție de vârstă?



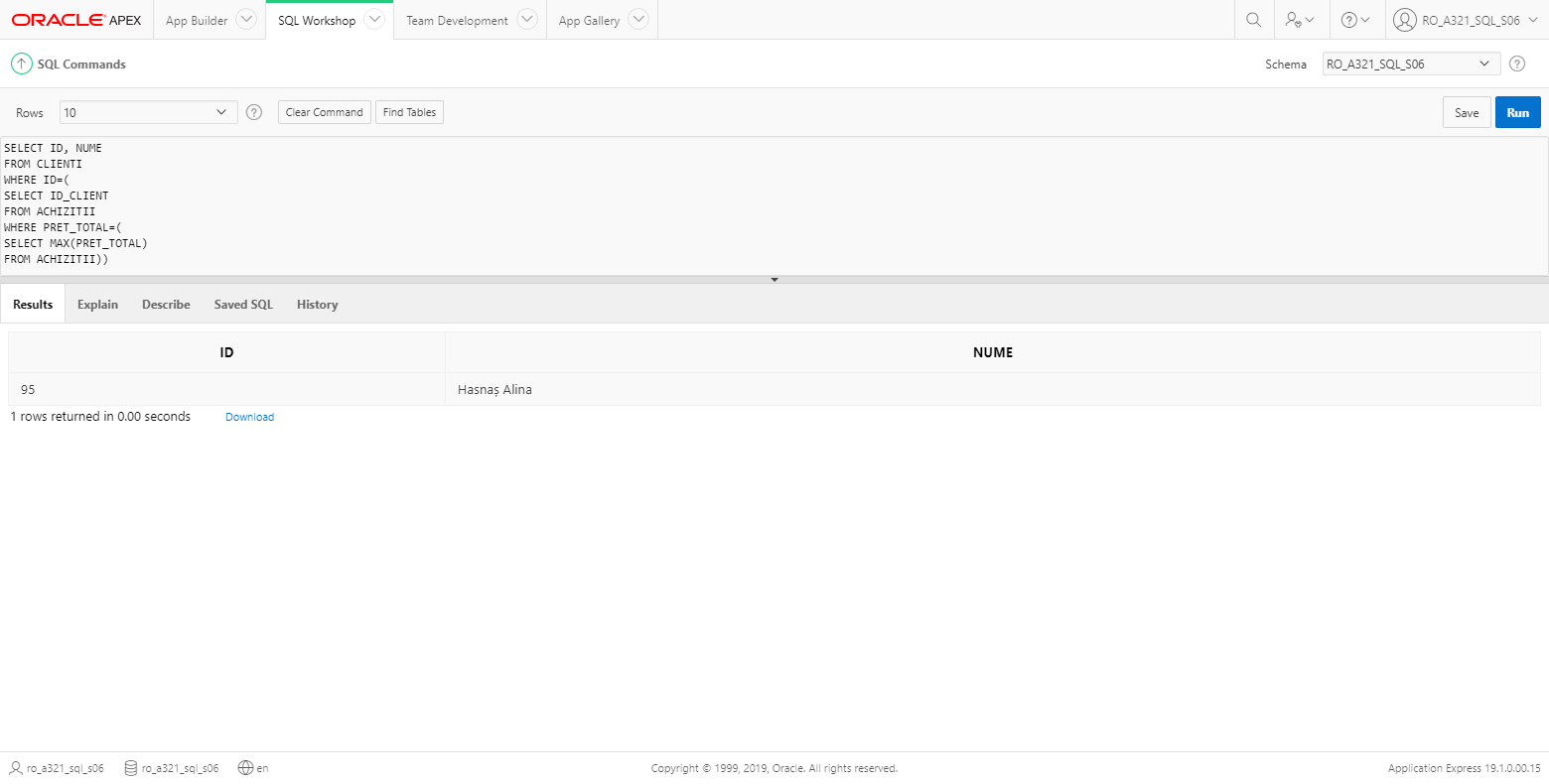
1. Care este, în medie, numărul de locuri dintr-o sală a cinematografului?



1. Care este numărul de telefon al managerului Sava Xenia?



1. Care este clientul care a făcut achiziția cu cel mai mare preț?



# Bibliografie

1. Mioara Gheorghe, Monica Tătărâm, Corina Achinica, Ioana Pestrițu: Informatică, Manual pentru clasa a XII-lea, editura Corint, anul 2007
2. Carmen Popescu: Manual de informatică pentru clasa a XII-a, editura L&S Informat, anul 2007
3. Carmen Popescu, Vlad Tudor (Hutanu): Manual de informatică intensiv pentru clasa a XII-a, editura L&S Infomat, anul 2007
4. <https://ro.wikipedia.org/wiki/Baz%C4%83_de_date>
5. <https://en.wikipedia.org/wiki/Database>
6. <https://en.wikipedia.org/wiki/Oracle_Database>
7. <https://www.smartdraw.com/entity-relationship-diagram/>