*Gestiunea unui*

*lanț cinematografic*

*Project made with Oracle SQL Developer*

Țilică Gabriel-Lucian

Cuprins

1. [Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.](#prima)
2. [Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.](#doi)
3. [Descrierea entităților, incluzând precizarea cheii primare.](#trei)
4. [Descrierea relațiilor, incluzând precizarea cardinalității acestora.](#patru)
5. [Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.](#cinci)
6. [Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.](#sase)
7. [Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.](#sapte)
8. [Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.](#opt)
9. [Realizarea normalizării până la forma normală 3(FN1-FN3).](#noua)
10. [Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).](#zece)
11. [Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.](#unsprezece)
12. [Formularea în limbaj natural și implementarea a 5 cereri SQL complexe ce vor utiliza, în ansamblul lor elemente specifice cerinței.](#doisprezece)
13. [Implementarea a 3 operații de actualizare și de suprimare a datelor, utilizând subcereri.](#treisprezece)

15) [Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minim 4 tabele, o cerere ce utilizează operația divison și o cerere care implementează analiza top-n.](#cincisprezece)

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

În cadrul acestui proiect vă este prezentată baza de date a unui lanț cinematografic, in care se specifica date concrete despre cinematografe, filmele care se difuzează, cei care vizioneză filmele respective, sălile unde se difuzează acestea, magazinul cinematografului, precum si al filmului difuzat la un moment dat care conține articole cu tematica acelui film (accesorii, îmbrăcăminte, jucării etc).

Un cinematograf conține mai multe difuzări de filme la anumite ore din zi. Condiția este ca locul in care se difuzează filmul(adică sala) sa nu fie ocupat in perioada de timp a difuzării filmului, de un altul. Prin urmare o condiție necesară este ca in momentul in care doua difuzări conțin același id de sală, intervalele acestor difuzări nu trebuie să se intersecteze.

Fiecare spectator vizionează un anume film, la care se poate uita daca a cumpărat un bilet pentru acesta. De asemenea cunoaștem atât costul biletului si al tuturor produselor pe care le oferă magazinul, cât și bugetul de care dispune fiecare spectator, astfel putem ști exact ce lucruri poate cumpăra și ce nu.

Se cunoaște și cantitatea fiecărui tip de produs pe care o poate oferi magazinul cinematografului, cât si cantitatea pe care spectatorul o cumpără de la acesta.

Cinematograful a angajat oameni care oferă clienților produsele pe care le are magazinul(orice produs in afara de bilet către film) în stoc.

De asemenea un cinematograf poate avea parteneri care oferă bani în schimbul derulării unei reclame pe ecranul sălii din cinema pe o perioada scurtă de timp.

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

Modelul de date din cadrul acestui proiect oferă constrângeri/ restricții de funcționare, care sunt următoarele:

1. Pot exista mai multe cinematografe la o singură adresă.
2. Un spectator poate cumpăra din cadrul magazinului cinematografului chiar dacă nu are bilet cumpărat.
3. Un spectator nu poate cumpăra un bilet către un film horror, doar dacă are peste 16 ani.
4. Nu există spectator care să nu aibă cel puțin o sumă de bani egală cu cel mai ieftin produs al cinematografului(bilet/produs).
5. Între fiecare film din cadrul aceleiași săli există cel puțin 15 minute în care nu incepe următorul film.
6. Dacă un cinematograf nu are încasări anuale atunci acesta se presupune a fi închis pe o perioadă nedeterminată si prin urmare fiecare tabelă va fi goală.
7. Fiecare partener de reclame poate avea maxim două reclame în pauza dintre filme.
8. Filmele nu se pot difuza duminica și în zilele în care este o sărbătoare(Aceste zile sunt 25 Decembrie, 31 Decembrie, 7 Ianuarie, 30 Noiembrie și 1 Octombrie)
9. Daca un cinematograf nu prezintă nici un comerciant, atunci spectatorii nu pot cumpăra produse de la magazinul cinematografului respective.
10. Un spectator nu poate cumpăra de la magazinul cinematografului mai multe obiecte decât numărul acestora din stocul magazinului).

Exercițiul 3): Descrierea entităților, incluzând precizarea cheii primare.

Pentru modelul de date referitor la informațiile unor cinematografe, structurile ADRESĂ,CINEMA,COMERCIANT,STOC,SALĂ,SPECTATOR,PARTENERI\_RECLAME,DIFUZARE\_FILM,BILET,FILM,PRODUSE\_MAGAZIN,PRODUSE\_FIRMĂ reprezintă entități.

Voi prezenta entitățile modelului de date, oferind o descriere detaliată a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

Toate entitățile sunt independente, cu excepția subentității PRODUSE\_FIRMĂ și a entităților STOC și SALĂ.

ADRESĂ: Este o entitate ce reprezintă locația unde se află un anumit cinematograf, aceasta având cheia primară *#id\_loc.*

CINEMA:Este o entitate ce reprezintă locul în care oamenii vin să vizioneze filme și își pot cumpără pentru un comfort suplimentar produse de la magazinul acestuia.Cheia primară a cinematografului este *#id\_cinema.*

COMERCIANT: Este o entitate ce reprezintă o persoană fizică care se ocupă de servirea oamenilor cu produsele dorite de către aceștia. El este angajat al cinematografului și va servi oamenii cu produse din cadrul stocului cinematografului. Cheia primară a acestei entități este *#id\_comerciant.*

STOC: Este o entitate dependentă de CINEMA și de PRODUSE\_MAGAZIN, care conține informații utile despre numarul de produse de care dispune cinematograful și din care se oferă produse spectatorilor.Cheia primară a acestei entități este compusă din *#id\_stoc, #id\_cinema* și *#id\_produs.*

SALĂ: Entitate dependentă de CINEMA, aceasta reprezentând locul unde se desfășoară filme în cursul zilei pentru cinematograful din care face parte și unde sunt găzduiți spectatorii acestora.Cheia primară a acestei entități este compusă din *#id\_sală* și *#id\_cinema.*

SPECTATOR: Este o entitate ce reprezintă persoana care vizionează filmele și/sau utilizează resursele de la magazinul cinematografului în schimbul a o sumă de bani de care acesta trebuie să dispună.Cheia primară a acestei entități este *#id\_spectator.*

PARTENERI\_RECLAME: Este o entitate ce se ocupă de furnizarea reclamelor în cadrul pauzelor dintre filme, spectatorilor aflați în sălile cinematografului.Parteneri oferă în schimbul difuzării reclamelor înaintea unui film o sumă de bani.Cheia primară a acestei entități este *#id\_partener.*

DIFUZARE\_FILM: Este entitatea ce se ocupă de gestionarea informațiilor desfășurării unui film.Difuzarea are loc în cadrul unei săli din cinematograful corespunzător în care se desfășoară un anumit film la un moment dat.Cheia primară a acestei entități este *#id\_difuzare.*

BILET: Este o entitate ce reprezintă cheia de acces către un anumit film, care poate fi cumpărată de către spectator pentru a-l viziona. Aceștia pentru a folosi un bilet, trebuie să ofere o sumă de bani stabilită de către cinematograf.Cheia primară a acestei entități este *#id\_bilet.*

FILM: Este o entitate ce reprezintă filmul pe care spectatorii îl vizionează, difuzarea acestuia fiind într-o sală a unui cinematograf respective.Filmul oferă spectatorilor, la finalul acestuia, posibilitatea de a cumpăra produse cu marca filmului respectiv.Cheia primară a acestuia este *#id\_film.*

*PRODUSE\_MAGAZIN:* Este o entitate ce oferă informații despre produsele magazinului cinematografului.Aceste produse sunt achiziționate de către spectatori în schimbul unei sume de bani, dacă î-și permit să le cumpere.Produsele se află în stocul magazinului din care sunt vândute.Cheia primară a acestei entități este *#id\_produs.*

PRODUS\_FIRMĂ:Este o subentitate a entității PRODUSE\_MAGAZIN și reprezintă produsele ce conțin marca unui anume film, difuzat la cinematograf.Cheia primară a acestei entități este *#id\_produs.*

Exercițiul 4): Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Vom prezenta relaţiile modelului de date, dând o descriere completă a fiecăreia. De fapt, denumirile acestor legături sunt sugestive, reflectând conţinutul acestora şi entităţile pe care le leagă. Pentru fiecare relaţie se va preciza cardinalitatea minimă şi maximă.

CINEMA\_*are*\_ADRESĂ = relație dintre CINEMA și ADRESĂ(localizarea unui cinematograf).Ținând cont de restricțiile impuse modelului, relația are cardinalitatea minimă 1:1 și cea maximă M:1.

CINEMA\_*are*\_STOC = relație dintre CINEMA și STOC(stocurile cu produse pe care le deține un cinematograf).Relația are cardinalitatea minimă 1:0 și cea maximă 1:M(un cinema poate conține mai multe stocuri sau niciunul, dar un stoc poate aparține unui singur cinema).

COMERCIANT\_*lucrează*\_CINEMA = relație dintre COMERCIANT și CINEMA(angajații pe care îi deține un cinematograf).Relația are cardinalitatea minimă 0:1 și cea maximă M:1(Un cinematograf poate avea mai mulți sau nici un comerciant, dar în schimb, un comerciant poate lucra la un singur cinema).

CINEMA\_*are*\_SALĂ = relație dintre CINEMA și SALĂ(sălile de care dispune un anumit cinematograf).Relația are cardinalitatea minimă 1:1 și cea maximă 1:M(Un cinematograf poate conține o sală sau mai multe, dar în schimb, o sală poate aparține unui singur cinematograf).

PRODUSE\_MAGAZIN\_*sunt\_pe*\_STOC = relație de tip *many-to-many* dintre PRODUSE\_MAGAZIN și STOC(reprezintă locurile în care sunt depozitate produsele magazinului unui cinematograf).Relația are cardinalitatea minimă 0:1 , iar cea maxima este M:N(Un tip de produs poate fi pe un stoc sau pe mai multe, iar un stoc poate conține nici un tip de produs sau mai multe tipuri de produse).

SALĂ*\_se\_difuzează\_*DIFUZARE\_FILM = relație dintre SALĂ și DIFUZARE\_FILM(Difuzările filmelor pe care le poate susține o sală a cinematografului).Relația are cardinalitatea minimă 1:0 și cea maximă 1:M(O sală poate susține mai multe difuzări de filme , dar în schimb un film poate fi difuzat într-o singură sală).

DIFUZARE\_FILM\_*folosește*\_BILET = relație dintre DIFUZARE\_FILM și BILET(Reprezintă biletele care sunt folosite de către spectatori în cadrul difuzării unui film).Relația are cardinalitatea minimă 1:0 și cea maximă 1:M(La o difuzare de film pot fi folosite mai multe bilete sau nici unul, dar în schimb, un bilet poate fi folosit la o singură difuzare de film).

SPECTATOR\_*achiziționează*\_BILET = relație dintre SPECTATOR și BILET(Reprezintă biletele care pot fi cumpărate de spectator dacă acesta dispune de bugetul necesar).Relația are cardinalitatea minima 1:0 și cea maximă 1:M(Un spectator poate cumpăra mai multe bilete la același film sau la filme diferite, dar în schimb un bilet poate fi achiziționat de un singur spectator).

DIFUZARE\_FILM\_*reclame\_difuzare*\_PARTENERI\_RECLAME = relație de tip *many-to-many* dintre DIFUZARE\_FILM și PARTENERI\_RECLAME(Reprezintă reclamele pe care partenerii cinematografului le difuzează înainte de începerea unui film).Relația are cardinalitatea minimă 1:0 și cea maximă M:N(O difuzare a unui film poate avea reclame , înaintea inceperii filmului, de la mai mulți parteneri ai cinematografului sau de la nici unul, iar un partener de reclame al cinematografului poate include reclame înaintea la una sau mai multe difuzări de filme).

DIFUZARE\_FILM*\_se\_difuzează\_*FILM = relație dintre DIFUZARE\_FILM și FILM(Reprezintă gestionarea difuzărilor unui anumit film din cadrul cinematografului).Cardinalitatea minimă este 0:1 și cea maximă M:1(La o difuzare de film se va difuza un singur film, iar un film poate avea mai multe difuzări, sau nici una).

SPECTATOR*\_cumpără\_de\_la\_*COMERCIANT*\_produse\_din\_*STOC = relație de tip 3 ce leagă entitățile SPECTATOR, COMERCIANT și STOC reflectând lucrurile pe care le-a cumpărat un spectator din cadrul stocului magazinului, și ce comerciant i le-a vandut.

Exercițiul 5):

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

Entitatea **ADRESĂ** are ca atribute:

Id\_loc = variabilă de tip întreg, de lungime maximă 5, care reprezintă idul adresei unui cinematograf.

Oraș = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele orașului unde este localizat cinematograful.

Stradă = variabilă de tip caracter, de lungime maximă 35, care reprezintă numele străzii unde este localizat cinematograful.

Țară = variabilă de tip caracter, de lungime maximă 20, care reprezintă numele țării unde este localizat cinematograful.

Entitatea **CINEMA** are ca atribute:

Id\_cinema = variabilă de tip întreg, de lungime maximă 5, care reprezintă idul unui cinematograf.

id\_locație = variabilă de tip întreg, de lungime maximă 5, care reprezintă idul adresei cinematografului.

Nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele cinematografului.

Încasări\_an = variabilă de tip float, de lungime maximă 10, care reprezintă încasările anuale în lei ale cinematografului.

Clienți\_an = variabilă de tip întreg, de lungime maximă 7, care reprezintă numărul de clienți de care cinematograful dispune anual.Considerăm faptul că acest număr diferă de la an la an, prin urmare a fost realizată media artimetică a acestor date până în anul curent.

Entitatea **COMERCIANT** are ca atribute:

Id\_comerciant = variabilă de tip întreg, de lungime maximă 5, care reprezintă idul unui comerciant al cinematografului.

Id\_cinema = variabilă de tip întreg, de lungime maximă 5, care reprezintă idul cinematograf la care lucrează comerciantul.

Data\_angajării = variabilă de tip date, cu valori între anii 1950-2023 ce reprezintă data la care a fost angajat comerciantul în cauză.

Salariu = variabilă de tip float, de lungime maximă 6, care reprezintă încasările lunare ale angajatului.

Nume = variabilă de tip caracter, de lungime maximă 25, care reprezintă numele angajatului.

Prenume = variabilă de tip caracter, de lungime maximă 25, care reprezintă prenumele angajatului.

Bonus\_An = variabila de tip întreg, de lungime maximă 4 , ce reprezintă bonusurile pe care le premește angajatul pe parcursul anului , acestea fiind independete de salariul pe care îl primește.

Entitatea **STOC** are ca atribute: id\_stoc, id\_cinema,id\_prod,nr\_prod, nr\_prod (reprezintă numărul produselor de tipul id\_prod pe care le deține stocul respectiv).

Entitatea **SALĂ** are ca atribute: id\_sală,id\_cinema,nr\_locuri(var de tip int de lungime maximă 3),lungime(var de tip int de lungime maximă 3 ce reprezintă lungimea sălii),lățime(var de tip int de lungime maximă 3 ce reprezintă lățimea sălii).

Entitatea **SPECTATOR** are ca attribute:id\_spec, nume,prenume,vârstă,sex(variabilă de tip char ce poate avea valorile de „masculin” sau „feminin”),buget(variabilă de tip int de lungime maximă 4 ce reprezintă suma de bani de care dispune spectatorul ,și care poate fi folosită pentru achiziționarea produselor din magazinul cinematografului).

Entitatea **PARTENERI\_RECLAME** are ca atribute:id\_partener,nume,cost\_film(variabilă de tip int de lungime maximă 5, ce reprezintă un partener al cinematografului care se ocupă cu furnizarea de reclame înainte de începerea unei difuzări a unui film, acesta plătind o sumă de bani pentru difuzarea reclamelor)

Entitatea **DIFUZARE\_FILM** are ca atribute: id\_difuzare, id\_sală,id\_film,ora\_început,ora\_sfârșit(sunt variabile de tip time, a căror interval nu trebuie să se intersecteze cu intervalul difuzării unui alt film în aceași sală),data\_film(variabilă de tip date ce reprezintă data la care se difuzează filmul acesta având restricții la 25 Decembrie, 31 Decembrie, 7 Ianuarie, 30 Noiembrie și 1 Octombrie).

Entitatea **BILET** are ca atribute: id\_bilet, id\_spec,id\_difuzare,preț(variabilă de tip int, de lungime maximă 3 ce reprezintă costul biletului).

Entitatea **FILM** are ca attribute : id\_film,categorie(variabilă de tip char de lungime maximă 10 ce poate lua valorile “horror”, ”comedie”, ”acțiune”, “dragoste” ,”SF” ,”dramă”, “anime”),tip\_vizualizare(variabilă de tip char de lungime maximă 10 ce poate lua valorile “3D”, ”normal”, ”VR”), nume\_film(variabilă de tip char de lungime maximă 20 ce reprezintă numele filmului vizionat).

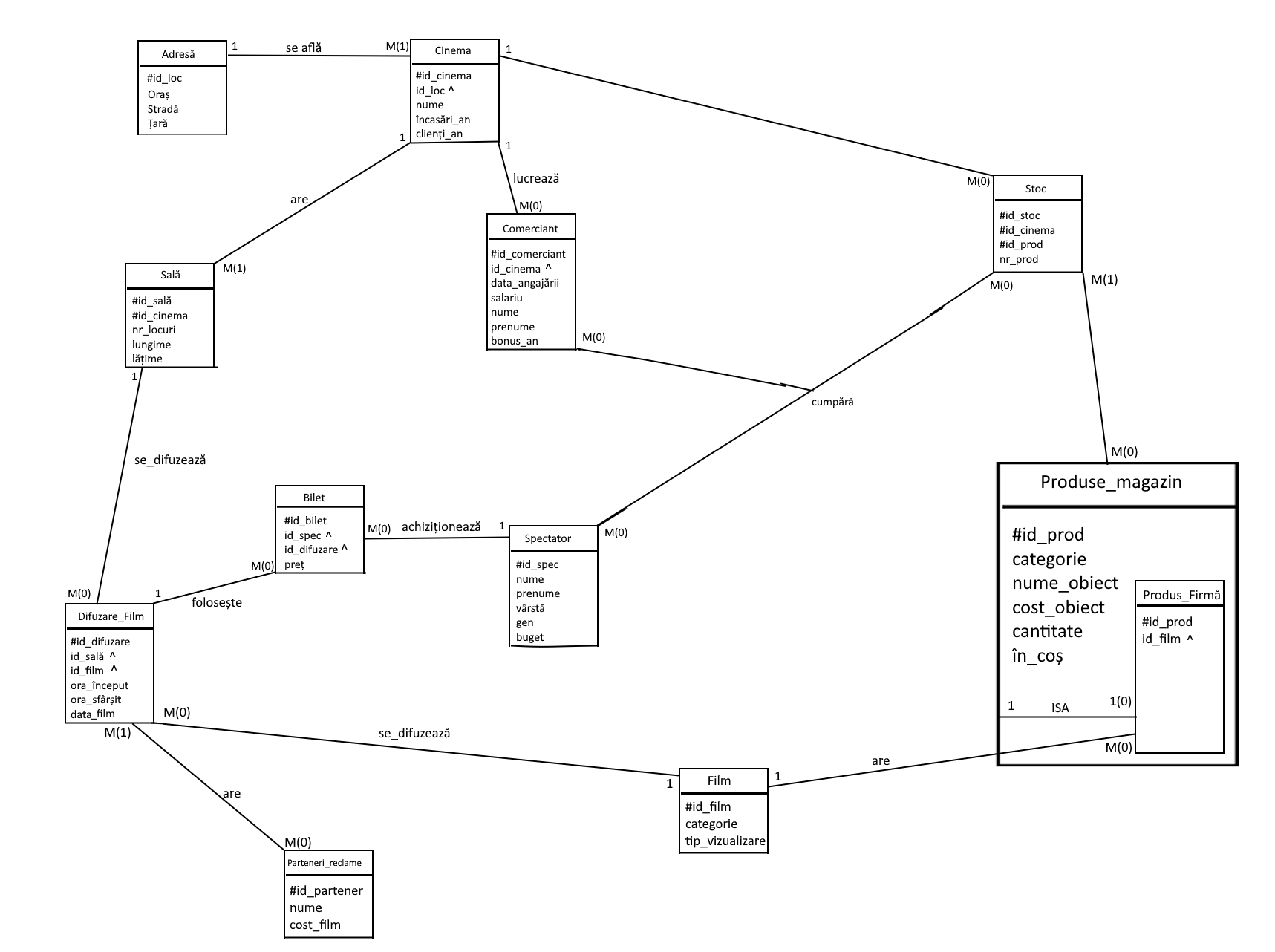
Entitatea **PRODUSE\_MAGAZIN** are ca attribute: id\_prod, categorie(variabilă de tip char de lungime maximă 10 ce poate lua valorile „băuturi”, “dulciuri”, “snacks” sau “firmă”), nume\_obiect(variabilă de tip char cu lungime maximă de 30 ce poate lua valorile “apă”, „coca-cola”, „fanta”, „sprite”, „pepsi”, „jeleuri”, „corn-ciocolata”, „corn-vanilie”, „snickers”, „bounty”, „nachos”, „floricele”, „chipsuri” ,”churos” sau nume specifice unui produs de firmă), cost\_obiect(variabilă de tip intreg de lungime maximă 2 ce reprezintă costul obiectului), cantitate(variabilă de tip int de lungime maximă 2 ce reprezintă numărul de obiecte de tipul respectiv de care dispune magazinul cinematografului), în\_coș(variabilă de tip intreg de lungime maximă 2 ce reprezintă numărul total de obiecte de acel tip care au fost cumpărate de către specatori).

Subentitatea **PRODUSE\_FIRMA** are ca atribute: id\_produs și id\_film.Un produs de firma va avea automat categoria ca fiind “firmă”, iar produsele vor avea nume\_obiect sub forma “tricou\_*numefilm*”, “șapcă\_*numefilm”, “cană\_numefilm”, “jucărie\_numefilm”, “sticker\_numefilm”*.

Relația DIFUZARE\_FILM\_**reclame\_difuzare**\_PARTENERI\_RECLAME are ca atribute: id\_partener,id\_difuzare,perioada\_difuzare(variabilă de tip time ce reprezintă perioada in care a fost difuzată o anumită reclama înainte de începerea filmului), tip\_reclamă(variabilă de tip char de lungime maximă 10 ce poate să ia valorile “Publicitară” sau “Informativă”).

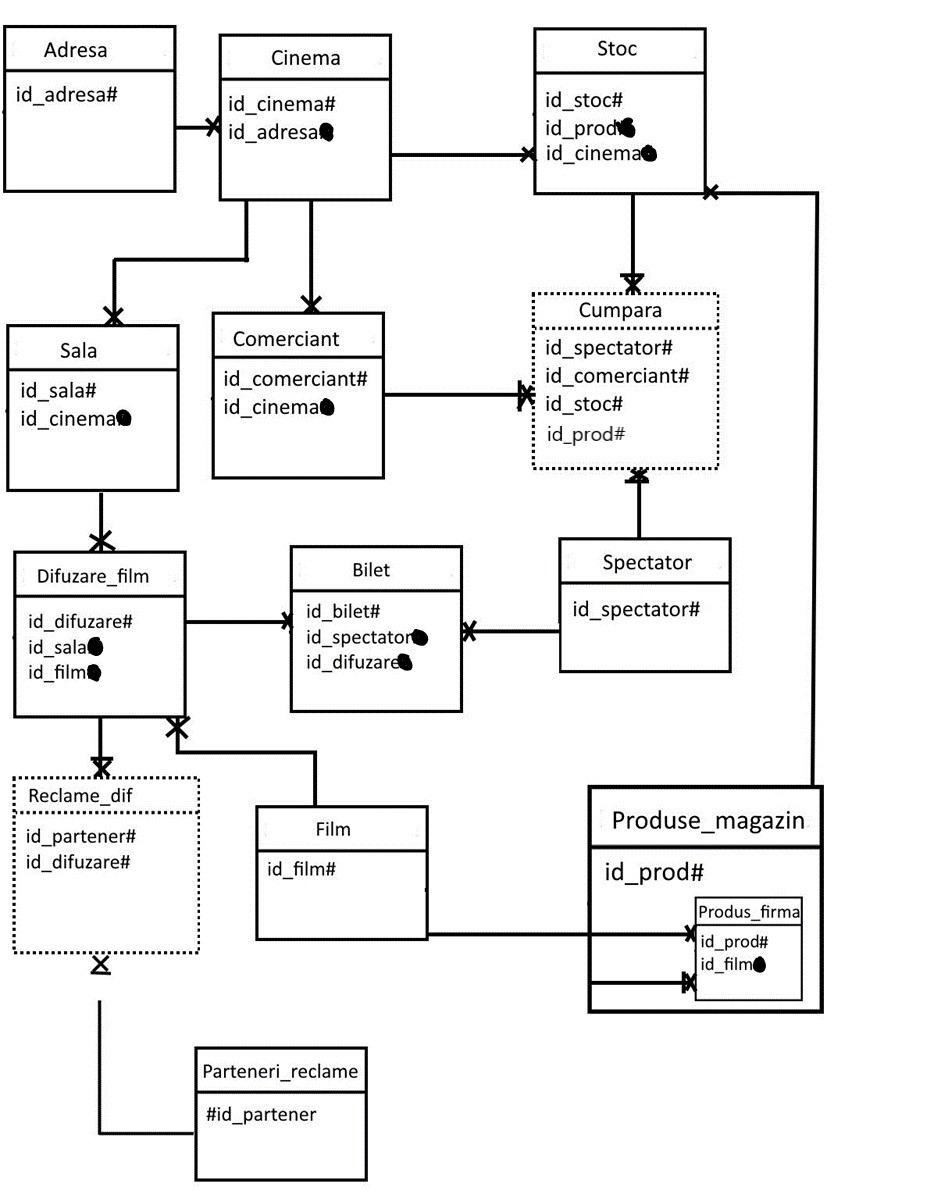
Relația SPECTATOR***\_cumpără\_de\_la****\_*COMERCIANT*\_****produse\_din****\_*STOC are ca atribute:id\_spec,id\_comerciant,id\_stoc,id\_produs,nr\_produse,metodă\_plată(variabilă de tip char cu lungimea maximă de 10 , ce poate lua valorile “cash”, “card” sau “cash\_card”).

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 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.



Reprezintă în cadrul desenului de mai sus cheia externă.

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

Schemele relaţionale corespunzătoare diagramei conceptuale sunt următoarele:

ADRESĂ(#id\_loc,Oraș,Stradă,Țară)

CINEMA(#id\_cinema,id\_loc,nume,încasări\_an,clienți\_an)

COMERCIANT(#id\_comerciant,id\_cinema,data\_angajării,salariu,nume,prenume,bonus\_an)

STOC(#id\_cinema,#cod\_cinema ,nr\_prod)

SALĂ(#id\_sală,#id\_cinema,nr\_locuri,lungime,lățime)

SPECTATOR(#id\_spec,nume,prenume,vârsta,gen,buget)

CUMPĂRĂ (#id\_spec,#id\_comerciant,#id\_stoc,#id\_prod,nr\_produse metodă\_plată)

PARTENERI\_RECLAME(#id\_partener,nume,cost\_film)

DIFUZARE\_FILM(#id\_difuzare,id\_cinema,id\_sală,id\_film,ora\_început,ora\_sfârșit,data\_film,zi\_săpt)

BILET(#id\_bilet,id\_spec,id\_difuzare,preț)

RECLAME\_DIFUZARE(#id\_partener,#id\_difuzare,perioada\_difuzare,tip\_reclamă)

FILM(#id\_film, categorie,tip\_vizualizare,nume)

PRODUSE\_MAGAZIN(#id\_prod,categorie,nume\_obiect,cost\_obiect,cantitate,în\_coș)

PRODUSE\_FIRMĂ(#id\_prod,id\_film)

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

FN1

Dacă relația M:M DIFUZARE\_FILM\_are\_PARTENERI\_RECLAME ar fi fost rezolvată prin introducerea unui atribut în DIFUZARE\_FILM care conține enumerarea partenerilor, atunci modelul nu s-ar mai fi aflat în FN1. O rezolvare a acestei relații care se află în FN1 este crearea unui tabel asociativ între DIFUZARE\_FILM și PARTENERI\_RECLAME.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_difuzare | Id\_sala | Id\_cinema | Id\_film | Ora\_inceput | Ora\_sfarsit | Data\_film | Id\_partener |
| 1 | 1 | 1 | 2 | 18:00:00 | 20:00:00 | 25-06-2019 | 1,2,3 |
| 2 | 2 | 1 | 4 | 20:15:00 | 21:30:00 | 25-06-2019 | 1,4,3 |

Exemplu NON-FN1

Exemplu FN1

Tabela Difuzare\_Film

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id\_difuzare | Id\_sala | Id\_cinema | Id\_film | Ora\_incepere | Ora\_sf | Data\_film |
| 1 | 1 | 1 | 2 | 18:00:00 | 20:00:00 | 25-06-2019 |
| 2 | 2 | 1 | 4 | 20:15:00 | 21:30:00 | 25-06-2019 |

Tabela Reclame\_dif

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_partener | Id\_difuzare | Perioada\_dif | Tip\_reclama |
| 1 | 1 | 17:50:00 | Informativa |
| 2 | 1 | 17:55:00 | Informativa |
| 3 | 1 | 18:00:00 | Publicitara |
| 1 | 2 | 20:05:00 | Publicitara |
| 4 | 2 | 20:10:00 | Informativa |
| 3 | 2 | 20:15:00 | Publicitara |

Tabela Parteneri Reclame

|  |  |  |
| --- | --- | --- |
| Id\_partener | nume | Cost\_film |
| 1 | Samsung | 50 |
| 2 | Pro-TV | 100 |
| 3 | Kaufland | 80 |
| 4 | Decathlon | 90 |

FN2

Dacă în entitatea SALĂ ar fi avut un atribut „adresă” care indică adresa cinemaului, atunci acest atribut ar fi depins doar de o parte din cheia primară, fapt ce nu ar fi fost în FN2. Pentru a intra în FN2 ar trebui ca atributul „adresă” să fie într-o entitate în care id\_cinema este cheie primară, cum ar fi entitatea CINEMA.

Non FN2

Tabela Cinema

|  |  |  |  |
| --- | --- | --- | --- |
| id\_cinema | id\_loc | incasari\_an | clienti\_an |

Tabela Comerciant

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id\_comerciant | id\_cinema | nume\_cinema | data\_angajarii | salariu | nume | prenume | bonus\_an |

Tabela comerciant nu se afla in FN2 deoarece nume\_cinema nu depinde in totaliate de cheia (id\_comerciant,id\_cinema) ci doar de id\_cinema.  
  
Pentru ca tabela sa fie in FN2, vom muta nume\_cinema in cinema

Tabela Cinema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_cinema | id\_loc | incasari\_an | clienti\_an | nume\_cinema |

Tabela Comerciant

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id\_comerciant | id\_cinema | data\_angajarii | salariu | nume | prenume | bonus\_an |

FN3

Avem tabela Spectator

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id\_spec | nume | prenume | varsta | Categorie\_varsta | gen | buget |

Categorie\_varsta retine daca spectatorul are acces sau nu la vreo categorie restrictionata de filme.

Tabela incalca FN3 deoarece exista o dependenta tranzitiva intre ID\_spec -> varsta -> categorie\_varsta.Pentru ca tabela sa fie in FN3 vom renunta la coloana categorie\_varsta.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id\_spec | nume | prenume | varsta | gen | buget |

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

CREATE SEQUENCE seq\_adresa

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_cinema

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_comerciant

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_stoc

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_sala

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_spectator

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_parteneri\_reclame

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_difuzare\_film

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_bilet

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_ film

start with 1

increment by 1

maxvalue 1000

nocycle;

CREATE SEQUENCE seq\_produse\_magazin

start with 1

increment by 1

maxvalue 1000

nocycle;

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).

create table adresa(

id\_loc number(5) primary key not null,

oras varchar(25) not null,

strada varchar(35) not null,

tara varchar(20) not null);

create table cinema(

id\_cinema number(5) primary key not null,

id\_loc number(5) not null,

constraint pk\_loc foreign key (id\_loc) references adresa(id\_loc) on delete cascade,

nume varchar(25) not null,

incasari\_an float(10),

clienti\_an number(7)

);

create table comerciant(

id\_comerciant number(5) primary key not null,

id\_cinema number(5) not null,

constraint pk\_cinema foreign key (id\_cinema) references cinema(id\_cinema) on delete cascade,

data\_angajarii date not null,

salariu float(6) not null,

nume varchar(25) not null,

prenume varchar(25) not null,

bonusuri\_an number(4)

--constraint c\_data\_angajarii check (data\_angajarii<

--1 1 1950 sysdate

);

create table sala(

id\_sala number(5) ,

id\_cinema number(5) ,

constraint pk3\_cinema foreign key (id\_cinema) references cinema(id\_cinema) on delete cascade,

constraint pk\_sala primary key (id\_sala, id\_cinema),

nr\_locuri number(3) not null,

lungime number(3) not null,

latime number(3) not null);

create table spectator(

id\_spectator number(5) primary key not null,

nume varchar(25) not null,

prenume varchar(25) not null,

varsta number(3) not null,

gen varchar(8) not null,

buget number(4) not null,

constraint c\_gen check (gen in('Masculin','Feminin'))

);

create table parteneri\_reclame(

id\_partener number(5) primary key not null,

nume varchar(25) not null,

cost\_film number(5) not null);

create table reclame\_dif(

id\_partener number(5) not null,

id\_difuzare number(5) not null,

perioada\_dif timestamp not null,

tip\_reclama varchar(11) not null,

constraint c\_tip check (tip\_reclama in('Publicitara','Informativa')),

primary key(id\_partener,id\_difuzare)

);

create table film(

id\_film number(5) primary key not null,

categorie varchar(10) not null,

constraint c\_categorie check (categorie in('Horror','Comedie','Actiune','Dragoste','SF','Drama','Anime')),

tip\_vizualizare varchar(10) not null,

constraint c\_vizualizare check (tip\_vizualizare in('3D','Normal','VR')),

nume\_film varchar(20) not null);

create table difuzare\_film(

id\_difuzare number(5) primary key not null,

id\_sala number(5) not null,

id\_cinema number(5) not null,

id\_film number(5) not null,

ora\_incepere timestamp not null,

ora\_sf timestamp not null,

data\_film date not null,

constraint fk\_sala foreign key (id\_sala,id\_cinema) references sala on delete cascade,

constraint fk\_film foreign key (id\_film) references film on delete cascade

);

create table produse\_magazin(

id\_prod number(5) primary key not null,

categorie varchar(10) not null,

constraint c2\_categorie check (categorie in('Bauturi','Dulciuri','Snacks','Firma')),

nume\_obiect varchar(30) not null,

--constraint c\_obiect check (nume\_obiect in('Apa','Coca-Cola','Fanta','Sprite','Pepsi','Jeleuri','Corn-ciocolata','Corn-vanilie','Snickers','Bounty','Nachos','Floricele')),

cost\_obiect number(2) not null,

cantitate number(2) not null,

in\_cos number(2));

create table produse\_firma(

id\_prod number(5) primary key not null,

constraint fk2\_prod foreign key (id\_prod) references produse\_magazin(id\_prod),

id\_film number(5) not null,

constraint fk2\_film foreign key (id\_film) references film(id\_film)

);

create table stoc(

id\_stoc number(5) primary key not null,

id\_cinema number(5) not null,

constraint pk2\_cinema foreign key (id\_cinema) references cinema(id\_cinema) on delete cascade,

id\_prod number(5) not null,

constraint pk\_prod foreign key (id\_prod) references produse\_magazin(id\_prod) on delete cascade,

nr\_prod number(5) not null

);

create table cumpara(

id\_spectator number(5) not null,

id\_comerciant number(5) not null,

id\_stoc number(5) not null,

id\_prod number(5) not null,

nr\_produse number(3) not null,

metoda\_plata varchar(10) not null,

constraint fk\_spectator foreign key (id\_spectator) references spectator(id\_spectator) on delete cascade,

constraint id\_comerciant foreign key (id\_comerciant) references comerciant(id\_comerciant) on delete cascade,

constraint id\_stoc foreign key (id\_stoc) references stoc(id\_stoc) on delete cascade,

constraint id\_prod foreign key (id\_prod) references produse\_magazin(id\_prod) on delete cascade,

constraint c\_metoda\_plata check(metoda\_plata in('Cash','Card','Cash\_Card')),

primary key(id\_spectator,id\_comerciant,id\_stoc,id\_prod)

);

create table bilet(

id\_bilet number(5) primary key not null,

id\_spectator number(5) not null,

id\_difuzare number(5) not null,

constraint fk2\_spec foreign key (id\_spectator) references spectator(id\_spectator),

constraint fk2\_difuzare foreign key(id\_difuzare) references difuzare\_film(id\_difuzare),

pret number(3) not null);

describe adresa;

insert into adresa values(seq\_adresa.nextval,'București','Mihai Bravu','România');

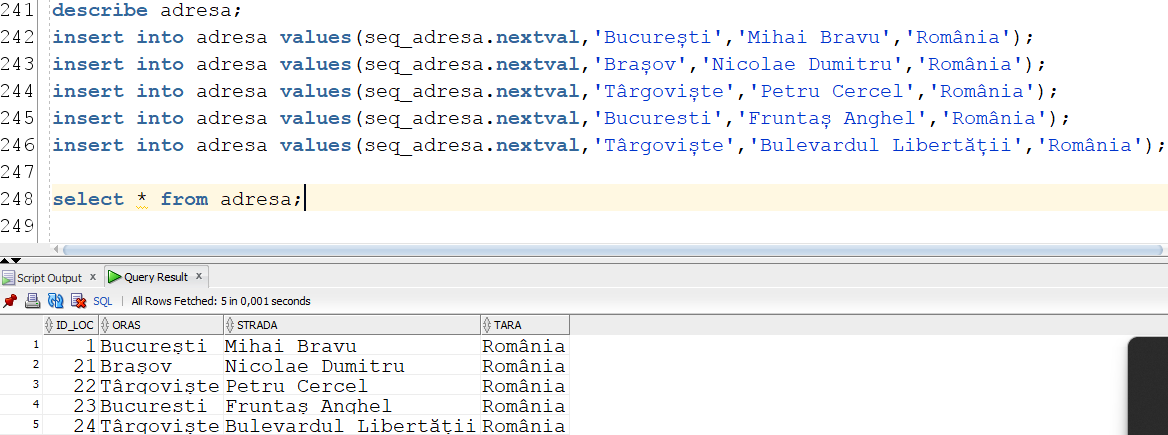
insert into adresa values(seq\_adresa.nextval,'Brașov','Nicolae Dumitru','România');

insert into adresa values(seq\_adresa.nextval,'Târgoviște','Petru Cercel','România');

insert into adresa values(seq\_adresa.nextval,'Bucuresti','Fruntaș Anghel','România');

insert into adresa values(seq\_adresa.nextval,'Târgoviște','Bulevardul Libertății','România');

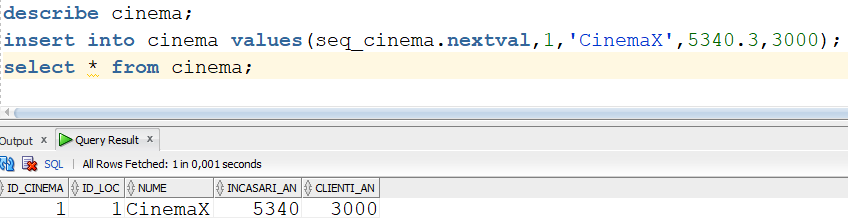
select \* from adresa;



describe cinema;

insert into cinema values(seq\_cinema.nextval,1,'CinemaX',5340.3,3000);

select \* from cinema;



describe comerciant;

insert into comerciant values(seq\_comerciant.nextval,1,'20-04-2018',3000,'Lemnaru','Mihai',500);

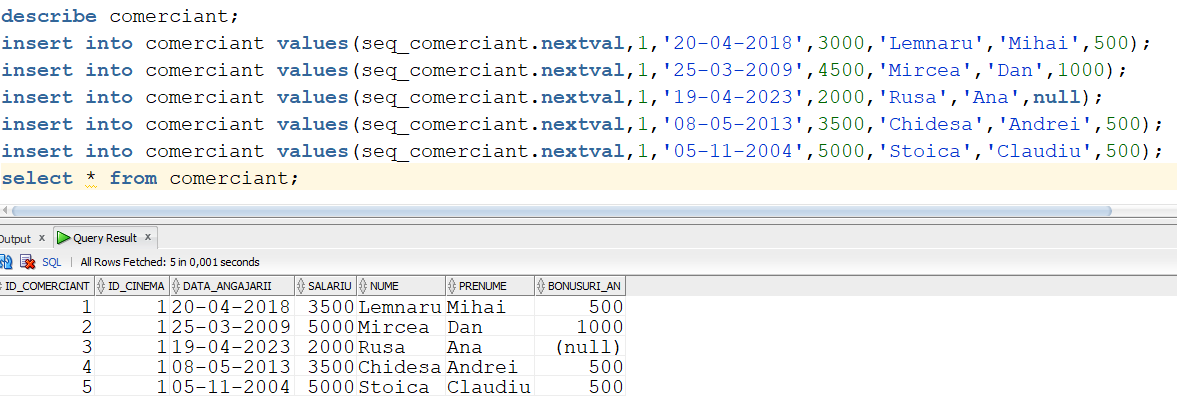
insert into comerciant values(seq\_comerciant.nextval,1,'25-03-2009',4500,'Mircea','Dan',1000);

insert into comerciant values(seq\_comerciant.nextval,1,'19-04-2023',2000,'Rusa','Ana',null);

insert into comerciant values(seq\_comerciant.nextval,1,'08-05-2013',3500,'Chidesa','Andrei',500);

insert into comerciant values(seq\_comerciant.nextval,1,'05-11-2004',5000,'Stoica','Claudiu',500);

select \* from comerciant;



describe film;

insert into film values(seq\_film.nextval,'Horror','3D','IT');

insert into film values(seq\_film.nextval,'Comedie','3D','Marea Mahmureală');

insert into film values(seq\_film.nextval,'SF','3D','Avatar');

insert into film values(seq\_film.nextval,'Actiune','3D','Vacanță Criminală');

insert into film values(seq\_film.nextval,'Horror','3D','Conjuring');

insert into film values(seq\_film.nextval,'Horror','Normal','IT');

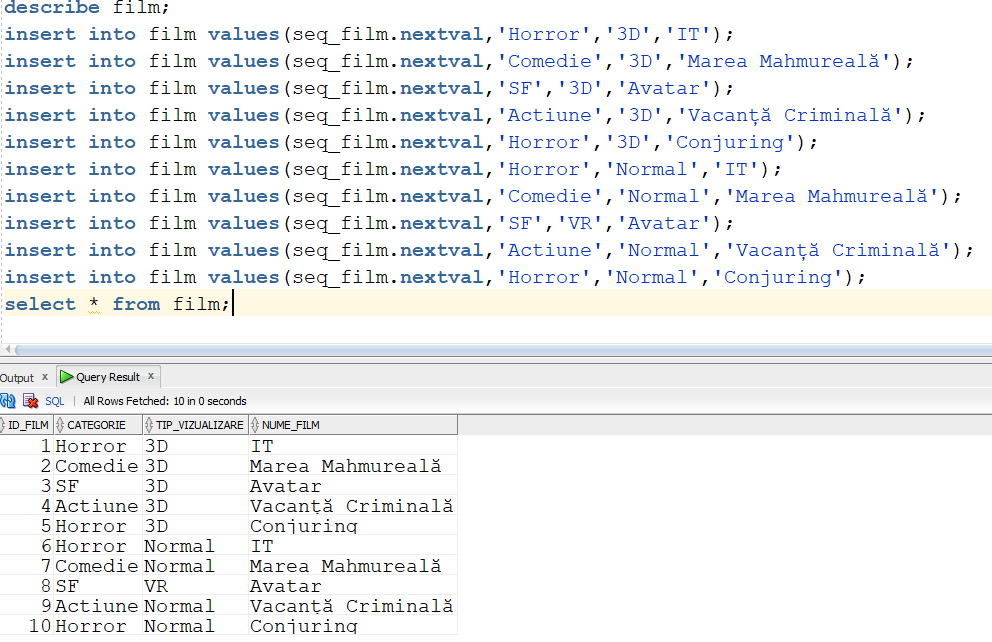
insert into film values(seq\_film.nextval,'Comedie','Normal','Marea Mahmureală');

insert into film values(seq\_film.nextval,'SF','VR','Avatar');

insert into film values(seq\_film.nextval,'Actiune','Normal','Vacanță Criminală');

insert into film values(seq\_film.nextval,'Horror','Normal','Conjuring');

select \* from film;



describe produse\_magazin;

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Bauturi','Apa',10,99,15);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Bauturi','Fanta',7,99,10);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Dulciuri','Jeleuri',15,99,5);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Snacks','Floricele',20,99,7);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Firma','Tricou\_IT',10,80,8);

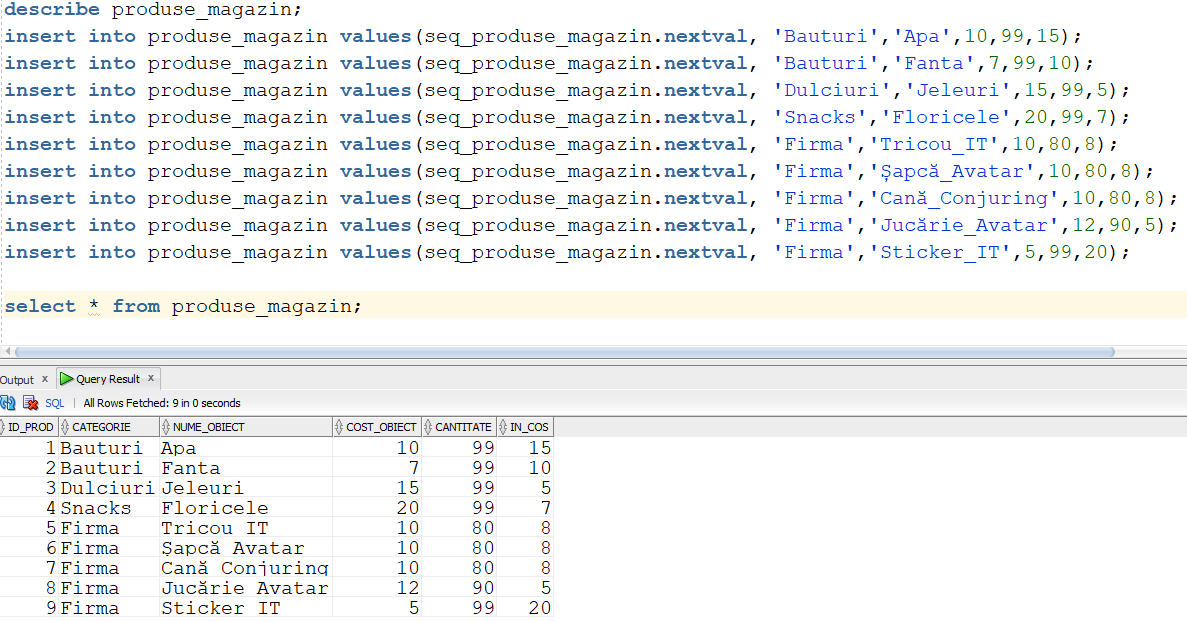
insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Firma','Șapcă\_Avatar',10,80,8);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Firma','Cană\_Conjuring',10,80,8);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Firma','Jucărie\_Avatar',12,90,5);

insert into produse\_magazin values(seq\_produse\_magazin.nextval, 'Firma','Sticker\_IT',5,99,20);

select \* from produse\_magazin;



describe produse\_firma;

insert into produse\_firma values(5,1);

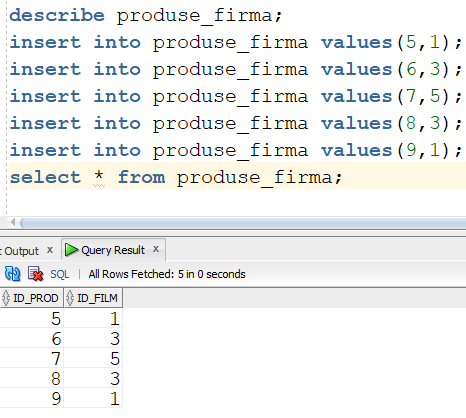
insert into produse\_firma values(6,3);

insert into produse\_firma values(7,5);

insert into produse\_firma values(8,3);

insert into produse\_firma values(9,1);

select \* from produse\_firma;



describe stoc;

insert into stoc values(seq\_stoc.nextval,1,1,49);

insert into stoc values(seq\_stoc.nextval,1,1,50);

insert into stoc values(seq\_stoc.nextval,1,2,49);

insert into stoc values(seq\_stoc.nextval,1,2,50);

insert into stoc values(seq\_stoc.nextval,1,3,49);

insert into stoc values(seq\_stoc.nextval,1,3,50);

insert into stoc values(seq\_stoc.nextval,1,4,49);

insert into stoc values(seq\_stoc.nextval,1,4,50);

insert into stoc values(seq\_stoc.nextval,1,5,40);

insert into stoc values(seq\_stoc.nextval,1,5,40);

insert into stoc values(seq\_stoc.nextval,1,6,40);

insert into stoc values(seq\_stoc.nextval,1,6,40);

insert into stoc values(seq\_stoc.nextval,1,7,40);

insert into stoc values(seq\_stoc.nextval,1,7,40);

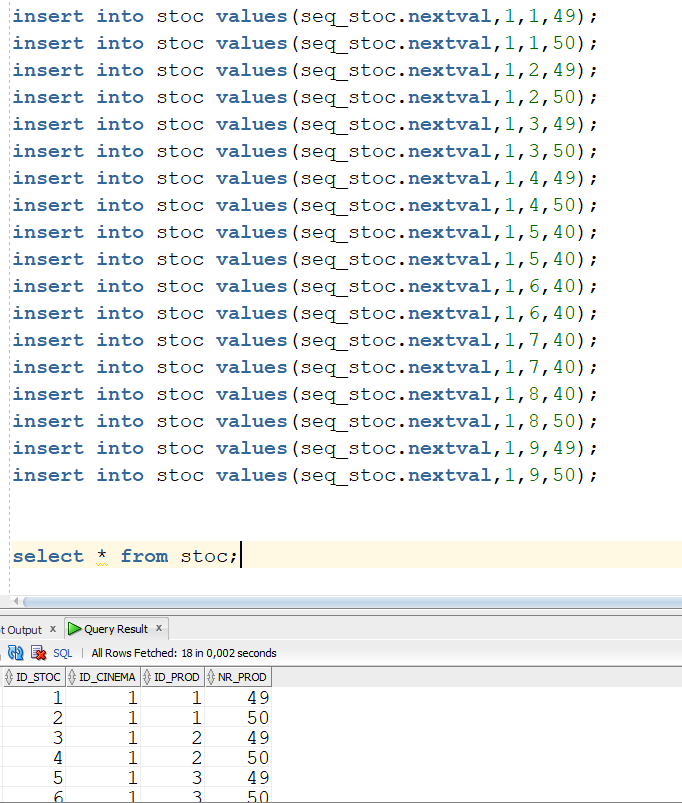
insert into stoc values(seq\_stoc.nextval,1,8,40);

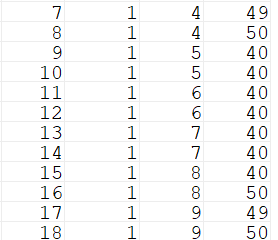
insert into stoc values(seq\_stoc.nextval,1,8,50);

insert into stoc values(seq\_stoc.nextval,1,9,49);

insert into stoc values(seq\_stoc.nextval,1,9,50);

select \* from stoc;





describe spectator;

insert into spectator values(seq\_spectator.nextval,'Ionescu','Alexandru',18,'Masculin',1000);

insert into spectator values(seq\_spectator.nextval,'Ionescu','Cristina',17,'Feminin',800);

insert into spectator values(seq\_spectator.nextval,'Miclea','Dan',23,'Masculin',1100);

insert into spectator values(seq\_spectator.nextval,'Chirvasa','ion',20,'Masculin',500);

insert into spectator values(seq\_spectator.nextval,'Dan','Octavian',32,'Masculin',350);

insert into spectator values(seq\_spectator.nextval,'Brasoveanu','Stefania',19,'Feminin',2400);

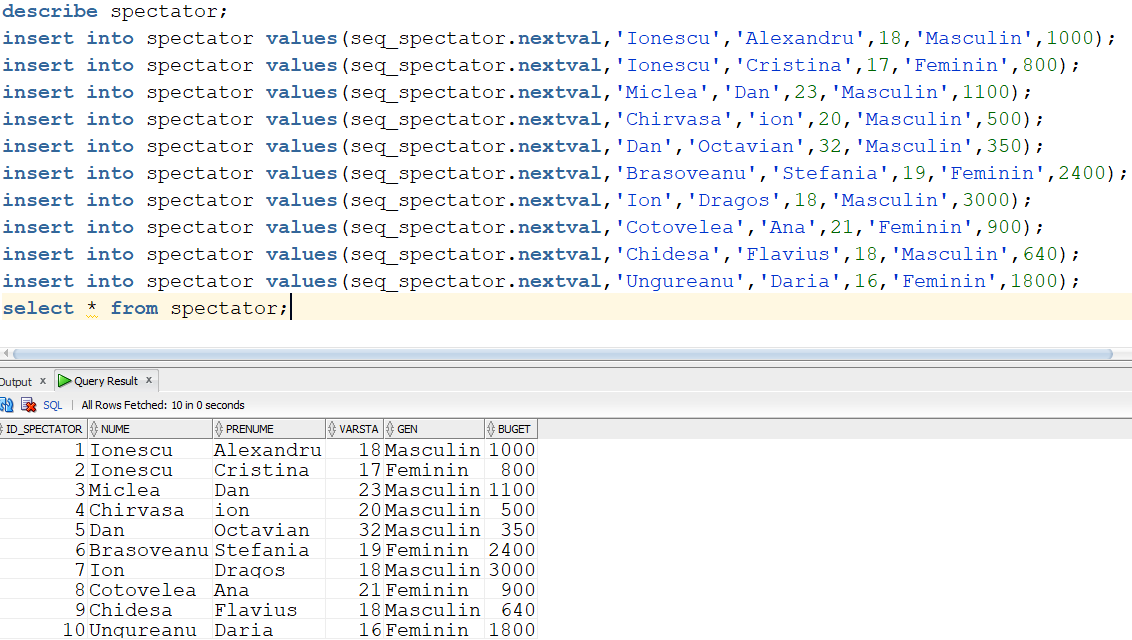
insert into spectator values(seq\_spectator.nextval,'Ion','Dragos',18,'Masculin',3000);

insert into spectator values(seq\_spectator.nextval,'Cotovelea','Ana',21,'Feminin',900);

insert into spectator values(seq\_spectator.nextval,'Chidesa','Flavius',18,'Masculin',640);

insert into spectator values(seq\_spectator.nextval,'Ungureanu','Daria',16,'Feminin',1800);

select \* from spectator;



describe cumpara;

INSERT INTO cumpara VALUES (1, 1, 1, 1, 3,'Cash');

INSERT INTO cumpara VALUES (3, 2, 2, 1, 7,'Cash');

INSERT INTO cumpara VALUES (6, 4, 1, 1, 5,'Card');

INSERT INTO cumpara VALUES (2, 3, 3, 2, 3, 'Cash\_Card');

INSERT INTO cumpara VALUES (1, 2, 4, 2, 2, 'Card');

INSERT INTO cumpara VALUES (8, 5, 3, 2, 3, 'Cash');

INSERT INTO cumpara VALUES (4, 1, 4, 2, 2, 'Cash');

INSERT INTO cumpara VALUES (3, 5, 5, 3, 3, 'Cash');

INSERT INTO cumpara VALUES (2, 2, 6, 3, 2, 'Cash');

INSERT INTO cumpara VALUES (1, 1, 7, 4, 3, 'Cash');

INSERT INTO cumpara VALUES (2, 2, 7, 4, 3, 'Cash');

INSERT INTO cumpara VALUES (8, 3, 7, 4, 1, 'Card');

INSERT INTO cumpara VALUES (5, 2, 9, 5, 1, 'Cash');

INSERT INTO cumpara VALUES (2, 2, 10, 5, 1, 'Cash');

INSERT INTO cumpara VALUES (8, 3, 10, 5, 2, 'Card');

INSERT INTO cumpara VALUES (1, 1, 9, 5, 1, 'Cash');

INSERT INTO cumpara VALUES (3, 4, 10, 5, 2, 'Cash');

INSERT INTO cumpara VALUES (6, 1, 9, 5, 1, 'Cash\_Card');

INSERT INTO cumpara VALUES (1, 4, 11, 6, 2, 'Cash');

INSERT INTO cumpara VALUES (9, 5, 11, 6, 2, 'Card');

INSERT INTO cumpara VALUES (10, 2, 11, 6, 2, 'Card');

INSERT INTO cumpara VALUES (5, 3, 12, 6, 2, 'Cash');

INSERT INTO cumpara VALUES (6, 3, 13, 7, 2, 'Cash');

INSERT INTO cumpara VALUES (4, 5, 14, 7, 4, 'Cash');

INSERT INTO cumpara VALUES (9, 1, 14, 7, 2, 'Cash');

INSERT INTO cumpara VALUES (2, 3, 15, 8, 2, 'Cash');

INSERT INTO cumpara VALUES (9, 1, 15, 8, 3, 'Card');

INSERT INTO cumpara VALUES (2, 3, 17, 9, 2, 'Cash');

INSERT INTO cumpara VALUES (10, 1, 18, 9, 3, 'Card');

INSERT INTO cumpara VALUES (3, 4, 17, 9, 2, 'Cash');

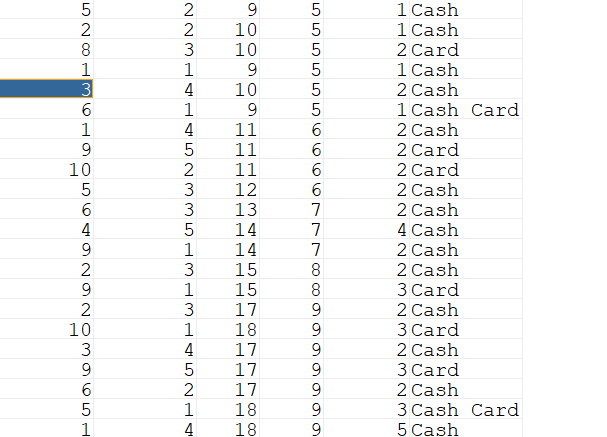
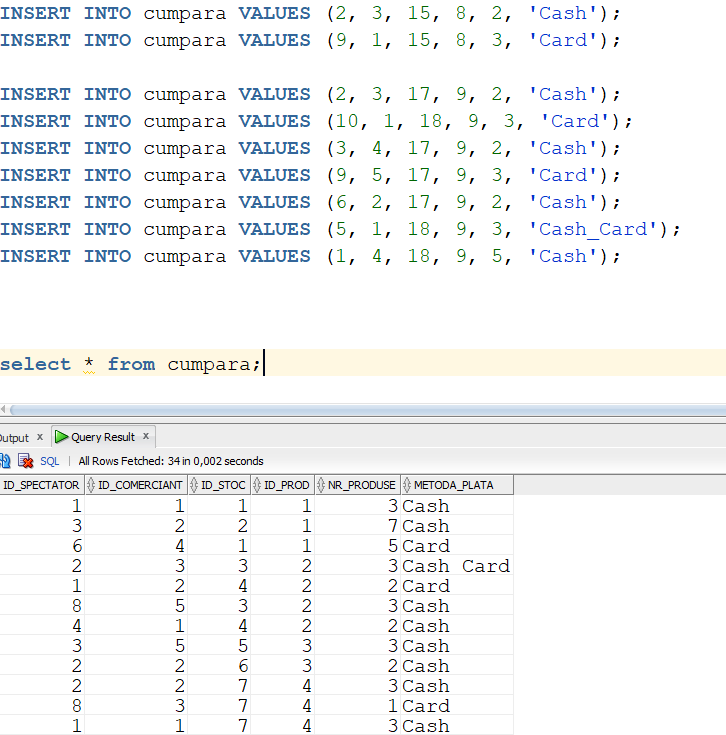
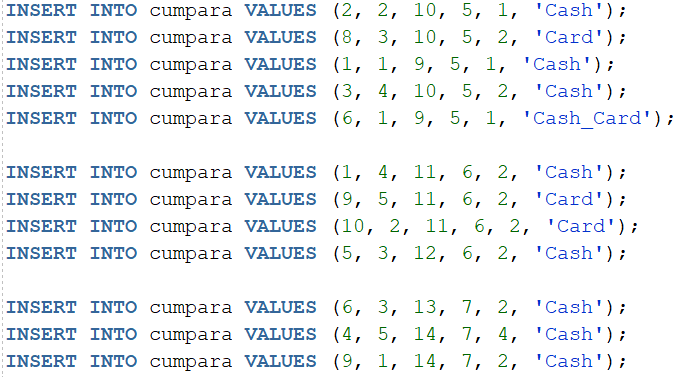
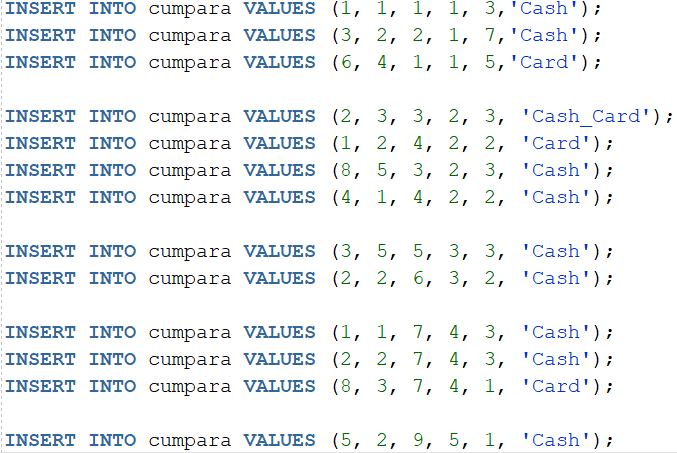
INSERT INTO cumpara VALUES (9, 5, 17, 9, 3, 'Card');

INSERT INTO cumpara VALUES (6, 2, 17, 9, 2, 'Cash');

INSERT INTO cumpara VALUES (5, 1, 18, 9, 3, 'Cash\_Card');

INSERT INTO cumpara VALUES (1, 4, 18, 9, 5, 'Cash');

select \* from cumpara;



describe sala;

insert into sala values(seq\_sala.nextval,1,30,20,25);

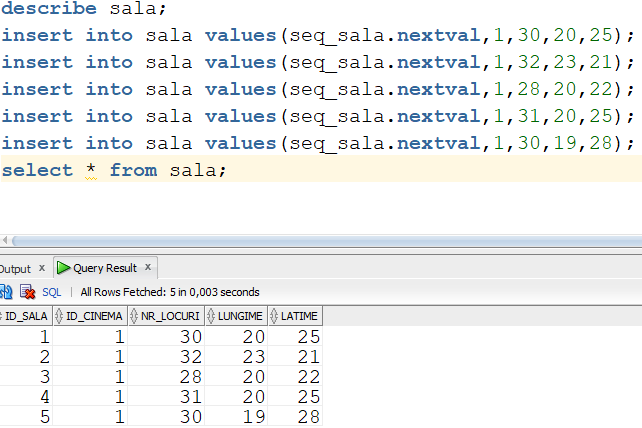
insert into sala values(seq\_sala.nextval,1,32,23,21);

insert into sala values(seq\_sala.nextval,1,28,20,22);

insert into sala values(seq\_sala.nextval,1,31,20,25);

insert into sala values(seq\_sala.nextval,1,30,19,28);

select \* from sala;



describe difuzare\_film;

insert into difuzare\_film values(seq\_difuzare\_film.nextval,1,1,2,TO\_TIMESTAMP('18:00:00', 'HH24:MI:SS'),TO\_TIMESTAMP('20:00:00', 'HH24:MI:SS'),'25-06-2019');

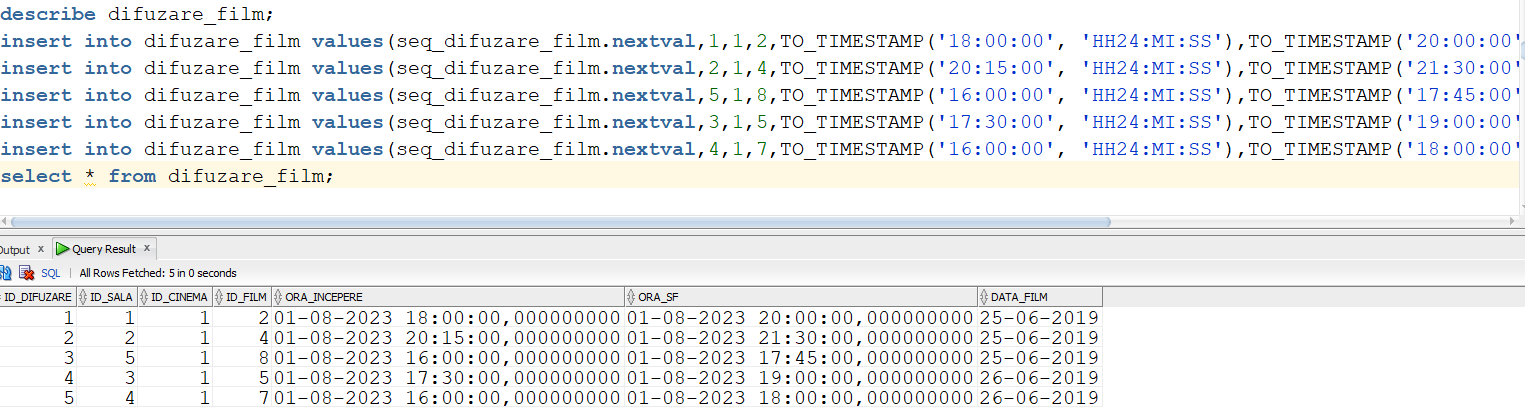
insert into difuzare\_film values(seq\_difuzare\_film.nextval,2,1,4,TO\_TIMESTAMP('20:15:00', 'HH24:MI:SS'),TO\_TIMESTAMP('21:30:00', 'HH24:MI:SS'),'25-06-2019');

insert into difuzare\_film values(seq\_difuzare\_film.nextval,5,1,8,TO\_TIMESTAMP('16:00:00', 'HH24:MI:SS'),TO\_TIMESTAMP('17:45:00', 'HH24:MI:SS'),'25-06-2019');

insert into difuzare\_film values(seq\_difuzare\_film.nextval,3,1,5,TO\_TIMESTAMP('17:30:00', 'HH24:MI:SS'),TO\_TIMESTAMP('19:00:00', 'HH24:MI:SS'),'26-06-2019');

insert into difuzare\_film values(seq\_difuzare\_film.nextval,4,1,7,TO\_TIMESTAMP('16:00:00', 'HH24:MI:SS'),TO\_TIMESTAMP('18:00:00', 'HH24:MI:SS'),'26-06-2019');

select \* from difuzare\_film;



describe bilet;

insert into bilet values(seq\_bilet.nextval,1,1,80);

insert into bilet values(seq\_bilet.nextval,2,2,80);

insert into bilet values(seq\_bilet.nextval,1,3,80);

insert into bilet values(seq\_bilet.nextval,4,1,80);

insert into bilet values(seq\_bilet.nextval,5,2,80);

insert into bilet values(seq\_bilet.nextval,6,3,80);

insert into bilet values(seq\_bilet.nextval,7,1,80);

insert into bilet values(seq\_bilet.nextval,8,2,80);

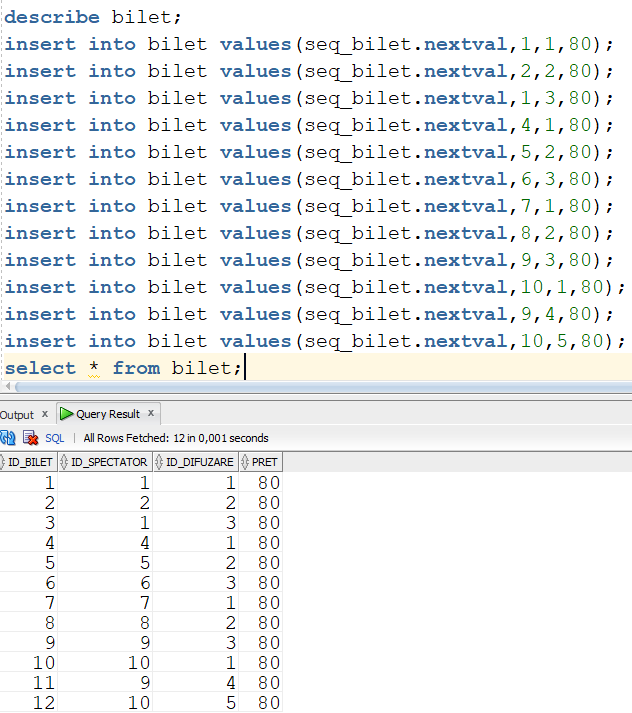
insert into bilet values(seq\_bilet.nextval,9,3,80);

insert into bilet values(seq\_bilet.nextval,10,1,80);

insert into bilet values(seq\_bilet.nextval,9,4,80);

insert into bilet values(seq\_bilet.nextval,10,5,80);

select \* from bilet;



describe parteneri\_reclame;

insert into parteneri\_reclame values(seq\_parteneri\_reclame.nextval,'Samsung',50);

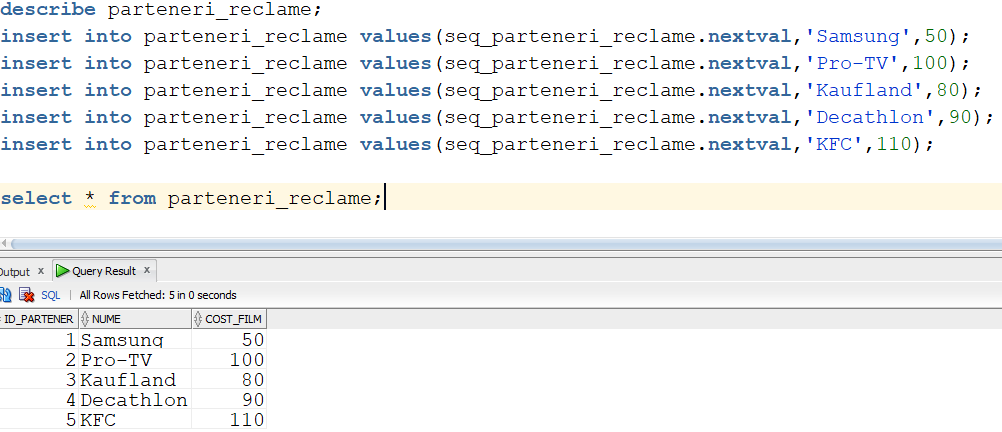
insert into parteneri\_reclame values(seq\_parteneri\_reclame.nextval,'Pro-TV',100);

insert into parteneri\_reclame values(seq\_parteneri\_reclame.nextval,'Kaufland',80);

insert into parteneri\_reclame values(seq\_parteneri\_reclame.nextval,'Decathlon',90);

insert into parteneri\_reclame values(seq\_parteneri\_reclame.nextval,'KFC',110);

select \* from parteneri\_reclame;



describe reclame\_dif;

insert into reclame\_dif values(1,2,TO\_TIMESTAMP('20:05:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(4,2,TO\_TIMESTAMP('20:10:00', 'HH24:MI:SS'),'Informativa');

insert into reclame\_dif values(3,2,TO\_TIMESTAMP('20:15:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(1,1,TO\_TIMESTAMP('17:50:00', 'HH24:MI:SS'),'Informativa');

insert into reclame\_dif values(2,1,TO\_TIMESTAMP('17:55:00', 'HH24:MI:SS'),'Informativa');

insert into reclame\_dif values(3,1,TO\_TIMESTAMP('18:00:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(1,3,TO\_TIMESTAMP('15:45:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(5,3,TO\_TIMESTAMP('15:50:00', 'HH24:MI:SS'),'Informativa');

insert into reclame\_dif values(3,3,TO\_TIMESTAMP('15:55:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(1,4,TO\_TIMESTAMP('17:15:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(2,4,TO\_TIMESTAMP('17:20:00', 'HH24:MI:SS'),'Informativa');

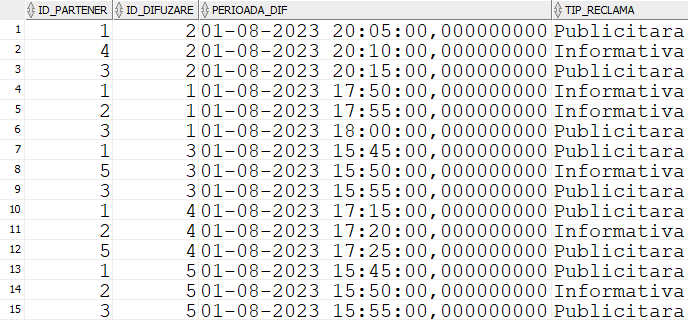
insert into reclame\_dif values(5,4,TO\_TIMESTAMP('17:25:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(1,5,TO\_TIMESTAMP('15:45:00', 'HH24:MI:SS'),'Publicitara');

insert into reclame\_dif values(2,5,TO\_TIMESTAMP('15:50:00', 'HH24:MI:SS'),'Informativa');

insert into reclame\_dif values(3,5,TO\_TIMESTAMP('15:55:00', 'HH24:MI:SS'),'Publicitara');

select \* from reclame\_dif;



Exercitiul 12)Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor următoarele elemente:1)subcereri sincronizate în care intervin cel puțin 3 tabele;2)subcereri nesincronizate în clauza FROM;3)grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup,filtrare la nivel de grupuri(în cadrul aceleiași cereri);4)ordonări și utilizarea funcțiilor NVL și DECODE(în cadrul aceleiași cereri);5)utilizarea a cel puțin 2 funcții pe șiruri de caractere,2 funcții pe date calendaristice, a cel puțin unei expresii CASE;6)utilizarea a cel puțin un bloc de cerere(clauza WITH).

Această interogare selectează id-urile, numele și prenumele spectactorilor masculini,cu vârsta de 18 ani sau mai mult, și care au bugetul mai mare sau egal cu media costurilor produselor pe care le-au cumpărat si lungimea numelui mai mare sau egala cu lungimea numelui zilei curente din saptamana.

In aceasta cerere am folosit subcerere nesincronizata in clauza from, o functie pe data calendaristica(sysdate)si am utilizat expresia CASE

select s.id\_spectator, s.nume, s.prenume

from (select id\_spectator,nume, prenume,varsta,gen,buget

from spectator

where varsta >= 18 and ascii(gen) = 77) s

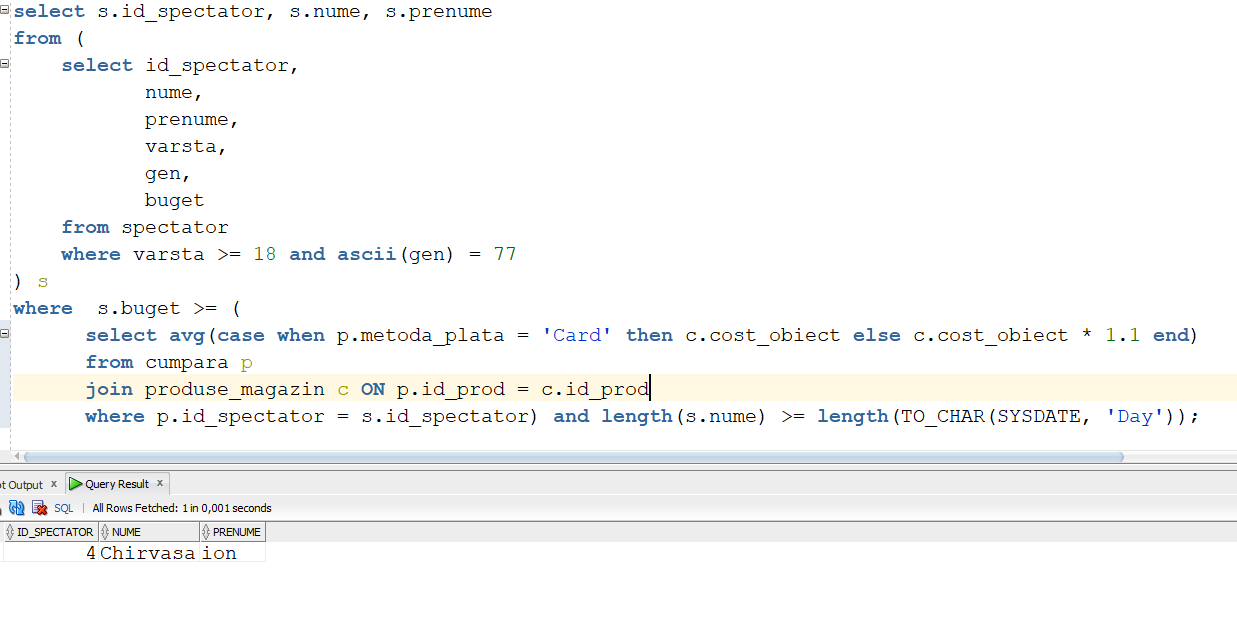
where s.buget >= (

select avg(case when p.metoda\_plata = 'Card' then c.cost\_obiect else c.cost\_obiect \* 1.1 end)

from cumpara p

join produse\_magazin c ON p.id\_prod = c.id\_prod

where p.id\_spectator = s.id\_spectator) and length(s.nume) >= length(TO\_CHAR(SYSDATE, 'Day'));



In urmatoarea cerere am afisat numele, prenumele, salariul si bonusurile anuale pe care le au comeciantii al caror nume se termina in 'a', lungimea numelui lor este mai mare decat cea mai mare lungime a numelor angajatilor din formula 'length(nume)/2 + 1', a caror bonus anual este egal cu 1500 si care sunt angajati de cel putin 11 ani.

Pentru aceasta cerere am folosit functiile NVL si DECODE, doua functii pe siruri de caractere si o functie pe date calendaristice.

select \* from comerciant;

select nume, prenume, salariu, nvl(bonusuri\_an,0)

from comerciant

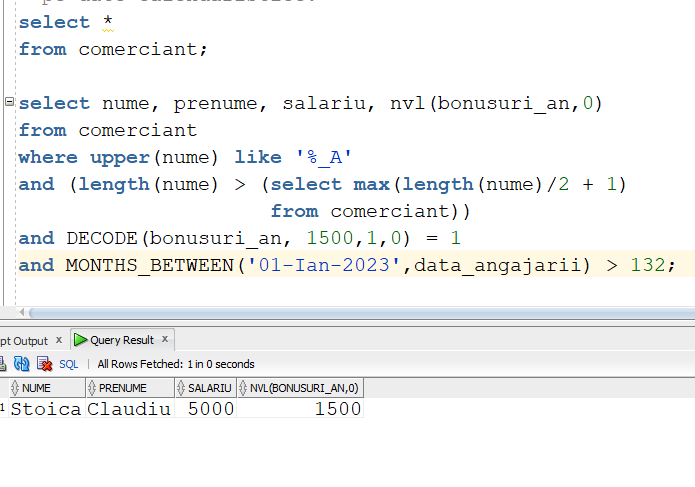
where upper(nume) like '%\_A'

and (length(nume) > (select max(length(nume)/2 + 1)

from comerciant))

and DECODE(bonusuri\_an, 1500,1,0) = 1

and MONTHS\_BETWEEN('01-Ian-2023',data\_angajarii) > 132;



Urmatoarea este o subcerere sincronizată în SQL în care sunt implicate cel puțin 3 tabele: spectator, cumpara și produse\_magazin.

Subcererea selectează toți spectatorii care au cumpărat produse din magazin cu metoda de plată "Card".

SELECT s.id\_spectator, s.nume, s.prenume

FROM spectator s

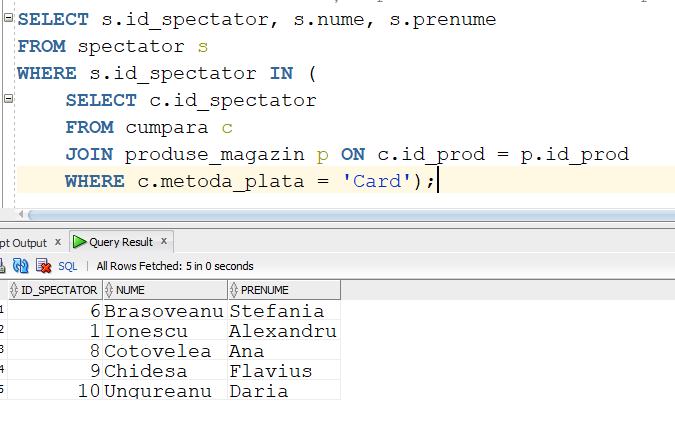
WHERE s.id\_spectator IN (

SELECT c.id\_spectator

FROM cumpara c

JOIN produse\_magazin p ON c.id\_prod = p.id\_prod

WHERE c.metoda\_plata = 'Card');



Această cerere va calcula media vârstelor cu o singură zecimală după virgulă pentru fiecare categorie de film și va afișa doar categoriile cu o medie de vârstă mai mare de 18 ani, sortate descrescător după media vârstelor.

De asemenea aceasta utilizeză grupari de date cu subcereri nesincronizate, in care intervin cel putin 3 tabele, functii grup si filtrare la nivel de grupuri.

select f.categorie, ROUND(avg(s.varsta), 1) as media\_varstei

from film f

join difuzare\_film df on f.id\_film = df.id\_film

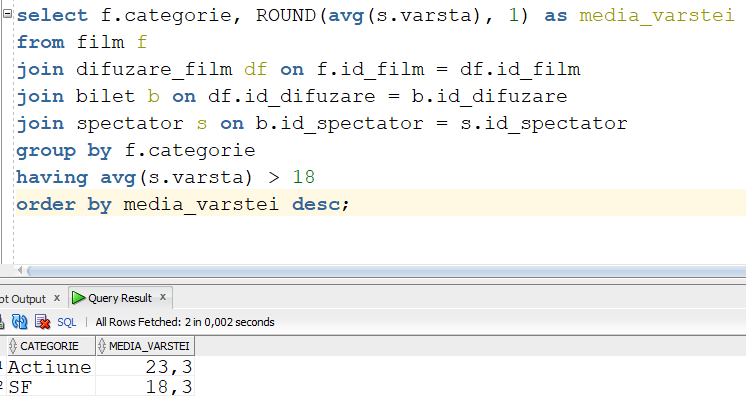
join bilet b on df.id\_difuzare = b.id\_difuzare

join spectator s on b.id\_spectator = s.id\_spectator

group by f.categorie

having avg(s.varsta) > 18

order by media\_varstei desc;



În acest exemplu, am uitlizat un bloc de cerere cu clauza WITH numit VanzariProduse pentru a calcula totalul vânzărilor pentru fiecare produs din tabela cumpara.

Apoi, cererea principală afișează numele produselor, totalul vânzărilor,precum și numele și prenumele spectatorilor care au efectuat achizițiile pentru produsele respective,numai pentru produsele cu totalul vânzărilor mai mare sau egal cu 10.

with VanzariProduse as (

select id\_prod, sum(nr\_produse) as total\_vanzari

from cumpara

group by id\_prod

)

select p.nume\_obiect, vp.total\_vanzari, s.nume, s.prenume

from VanzariProduse vp

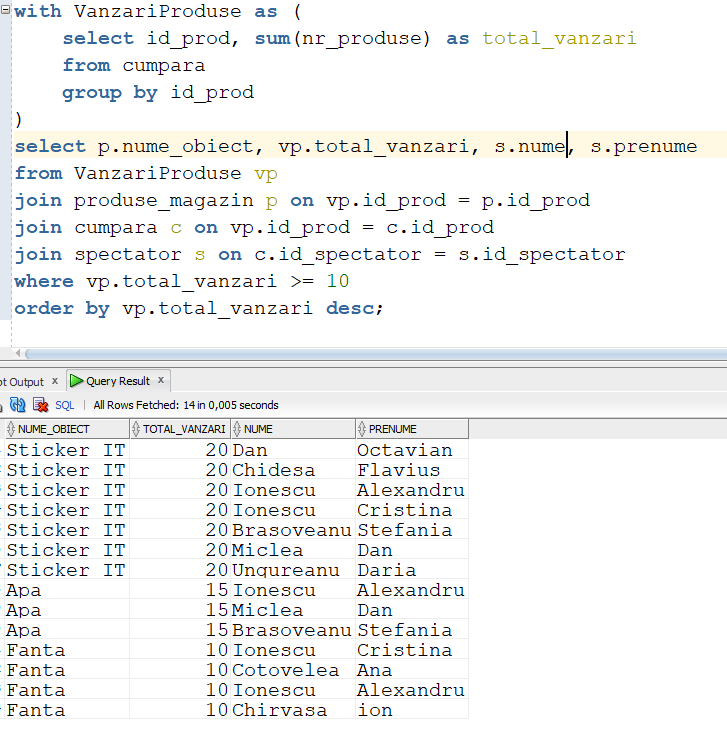
join produse\_magazin p on vp.id\_prod = p.id\_prod

join cumpara c on vp.id\_prod = c.id\_prod

join spectator s on c.id\_spectator = s.id\_spectator

where vp.total\_vanzari >= 10

order by vp.total\_vanzari desc;



Exercițiul 13)

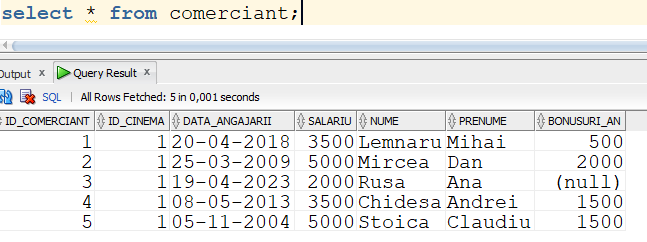
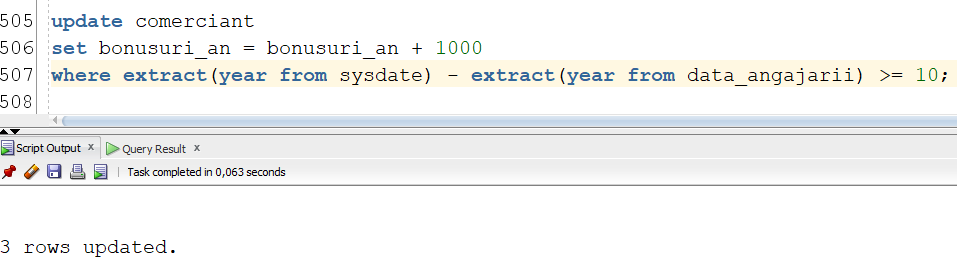
Implementarea a 3 operații de actualizare și de suprimare a datelor, utilizând subcereri.

Pentru angajatii care lucrează de cel puțin 10 ani(de dinainte sau din 2013) se vor adăuga cate 1000 la bonusurile anuale//UPDATE

update comerciant

set bonusuri\_an = bonusuri\_an + 1000

where extract(year from sysdate) - extract(year from data\_angajarii) >= 10;

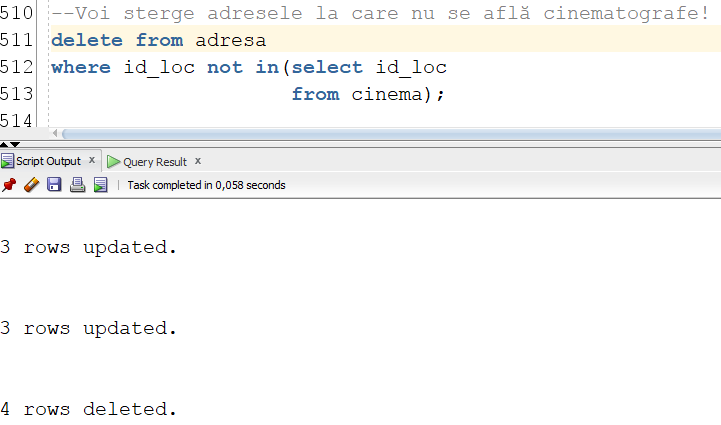


--Voi sterge adresele la care nu se află cinematografe!

delete from adresa

where id\_loc not in(select id\_loc

from cinema);



--Pentru angajatii care au servit cel putin de 8 ori clienți le voi mării salariul cu 500

describe comerciant;

update comerciant

set salariu = salariu + 500

where id\_comerciant in (

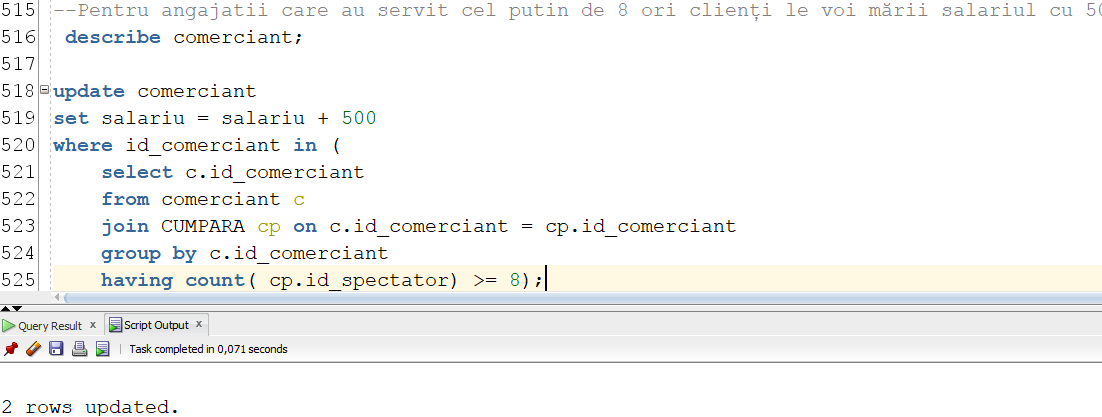
select c.id\_comerciant

from comerciant c

join CUMPARA cp on c.id\_comerciant = cp.id\_comerciant

group by c.id\_comerciant

having count( cp.id\_spectator) >= 8);



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

O cerere care utilizează operația division

--Această cerere va returna toți spectatorii care au cumpărat toate produsele din tabela "produse\_magazin".

--Subinterogarea din clauza NOT EXISTS verifică dacă un anumit spectator nu are nicio achiziție pentru un anumit produs.

SELECT DISTINCT C.id\_spectator

FROM cumpara C

WHERE NOT EXISTS (

SELECT P.id\_prod

FROM produse\_magazin P

WHERE NOT EXISTS (

SELECT NULL

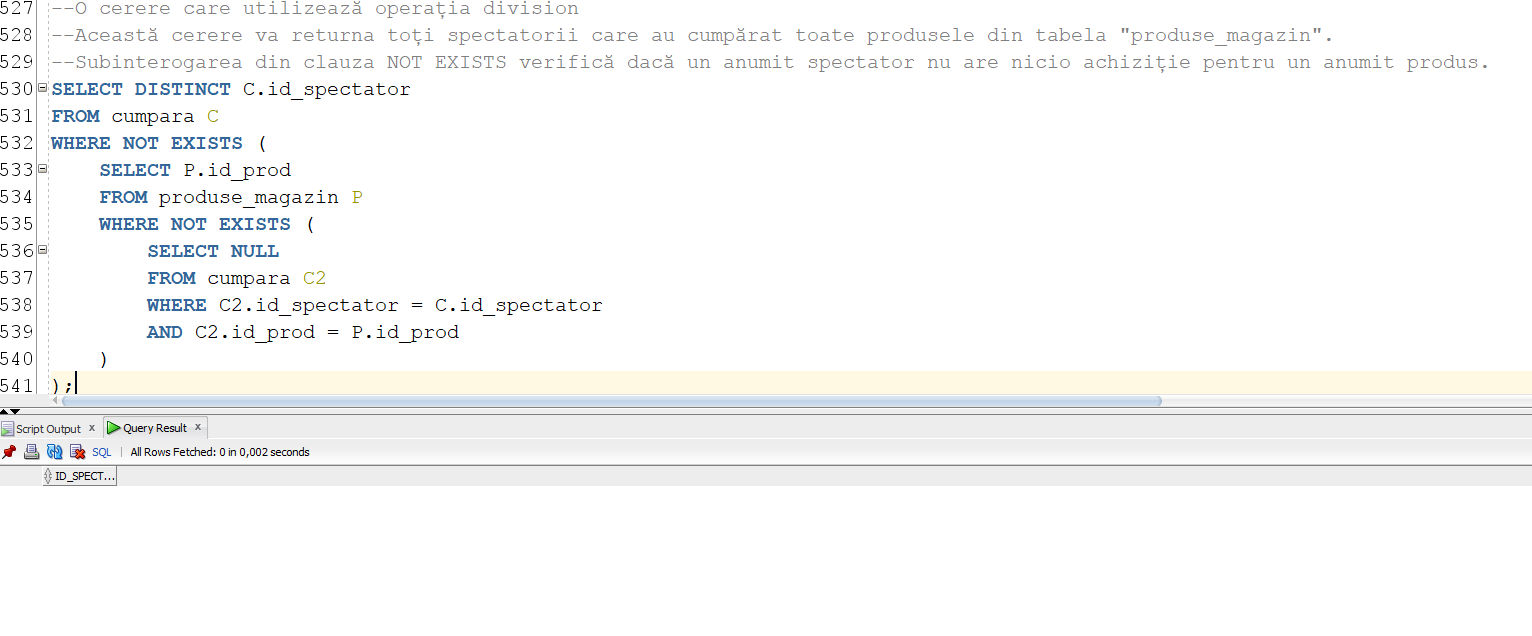
FROM cumpara C2

WHERE C2.id\_spectator = C.id\_spectator

AND C2.id\_prod = P.id\_prod

)

);



O cerere care implementează analiza top-n

--Utilizând rownum am afișat primele 5 cele mai ieftine produse de care dispune magazinul cinematografului

select id\_prod, nume\_obiect, cost\_obiect

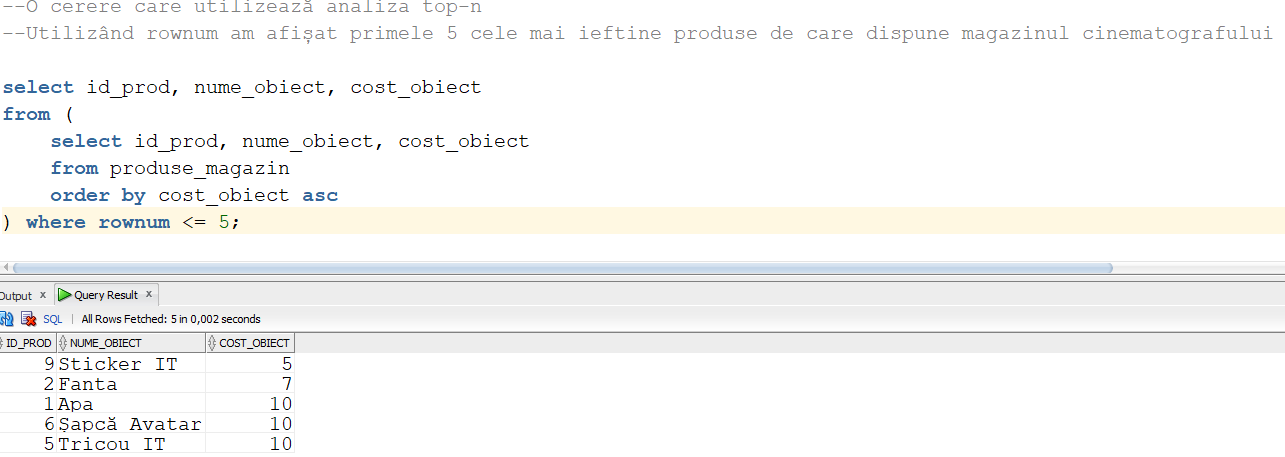
from (

select id\_prod, nume\_obiect, cost\_obiect

from produse\_magazin

order by cost\_obiect asc

) where rownum <= 5;



O cerere ce utilizează operația outer-join pe minimum 4 tabele.

Următoarea cerere ne oferă informații despre spectatorii, achizițiile acestora, numărul lor,metoda de plată pe care au folosit-o și comercianții care iau servit.

select s.nume, s.prenume, c.metoda\_plata, cm.nume||' '||cm.prenume as nume\_comerciant, c.nr\_produse,p.nume\_obiect

from spectator s

left join cumpara c on s.id\_spectator = c.id\_spectator

left join comerciant cm on c.id\_comerciant = cm.id\_comerciant

left join stoc st on c.id\_stoc = st.id\_stoc

left join produse\_magazin p on c.id\_prod = p.id\_prod;

