

Proiect

Modelarea unei baze de date pentru o platformă de filme

1. Descriere

Platforma de filme pentru care se păstrează date se aseamănă site-urilor care se găsesc online și care sunt tot mai des folosite astăzi.

Scopul acesteia este deci de a furniza anumite articole (filme sau seriale) pentru clienți, iar aceștia să le vizioneze după ce își creează un cont. Un client înregistrat pe platformă poate să își deschidă oricâte conturi, dar, de pe fiecare în parte, se va opta pentru un tip de abonament care are asociat un preț lunar. În funcție de abonament, contul va primi anumite reclame pentru oferte și sugestii noi de vizionare, adaptate pe specializarea abonamentului (pentru familie, pentru tineri, standard etc.). Filmele sau serialele sunt accesibile de pe un cont creat numai dacă persoana a plătit pentru un abonament, iar plata nu a expirat încă. De asemenea, contează vârsta clientului care a deschis contul pentru a avea acces la articole, unele filme sau seriale nefiind recomandate pentru toate vîrstele.

Serialele diferă de filme prin faptul că au asociate sezoane împărțite în episoade, astfel că la o accesare a serialului, contează de fapt episodul pe care o persoană dorește să îl vizioneze dintr-un sezon. La o accesare, se va păstra în baza de date procentul parcurs din articolul ales, eventual și procentul parcurs din episod.

Gruparea filmelor și serialelor este una sugestivă: toate articolele pe care site-ul le pune la dispoziție aparțin de o categorie sau, în unele cazuri, de mai multe (thriller, istoric, comedie, romantic etc.).

Este de menționat și că pe site se pot găsi traliere către toate articolele, fiecare film sau serial având un singur trailer. În plus, se pot citi și posta recenzii de pe un cont pentru un anume articol, recenzii care oferă un rating produsului ales.

Informațiile referitoare la articole, clienți, conturi, abonamente, recenzii etc. se vor păstra într-o bază de date destinată platformei și vor fi manipulate cu ajutorul limbajului SQL.

Reguli de funcționare

- Articolul poate fi de tip film sau serial, cele două reprezentând opțiuni distincte.
- Un articol are numele diferit de oricare alt articol de pe platformă.
- Un serial are sezoane care conțin mai multe episoade distincte, numerotate.

- Un client își poate deschide un număr nelimitat de conturi, dar de pe fiecare în parte va plăti separat pentru abonamente.
- Abonamentul se referă la strict la tipul de abonare, nu la un abonament personalizat pentru client.
- Reclamele care sunt incluse într-un abonament sunt personalizate doar pentru acesta și vin la pachet cu abonamentul odată ce se realizează plata.
- Dacă nu s-a optat pentru niciun abonament de pe un cont sau dacă plata a expirat, contul e inactiv.
- Pe platformă există un singur trailer pentru un articol, indiferent de tipul acestuia.
- Categoriile se referă la genul filmului sau serialului.
- Recenziile postează cuprind un rating obligatoriu pentru articolul recenzat și, opțional, comentarii.

2. Constrângeri

Pentru asigurarea corectitudinii informației și pentru a obține un plan realist al bazei de date, trebuie să se țină cont de următoarele restricții:

a) Constrângerea de cheie primară

- Atributul/ grupul de atribute prin care se identifică fiecare entitate în parte trebuie să fie de tip unic și nu poate lua valoarea NULL
 - Spre exemplu, nu pot exista două filme cu același cod sau doi clienți cu același id pentru că altfel nu s-ar mai putea accesa ușor informațiile.
- Unele chei primare sunt compuse
 - Un sezon se identifică prin id-ul serialului și prin numărul de sezon
 - Un episod prin cheia primară a sezonului și un număr de episod.

b) Constrângerea de cheie externă

Pentru a se realiza legături între unele entități, vor exista atribute dependente, care să indice către cheile primare ale unor alte tabele.

- Entitățile introduse pentru a spearge relațiile many to many au chei primare care sunt în același timp chei externe către tabelele între care se află.

- În tabela APARTINE, id_articol arată către atributul cu același nume din tabela de articole, iar id_categorie către atributul specific categoriei.
 - ACCESEAZA_FILM are cheie străină către cheile primare din ARTICOL și CONT, iar ACCESEAZA_SERIAL din EPISOD și CONT.
 - Tabela POSTEaza are cheie externă către trei entități: CONT, ARTICOL și RECENZIE.
 - Pentru entitatea PLATESTE, cheile externe sunt către ABONAMENT și CONT.
 - Trailerul are cheie externă către id-ul unui articol.
 - Pentru un cont conținează persoana care l-a deschis, deci are cheie străină către un client.
 - Reclama are un atribut care indică spre un abonament.
 - Sezonul are cheie externă către un articol, iar episodul cheie externă compusă către un sezon.
- c) Constrângerea de NOT NULL (valori obligatorii pentru unele câmpuri)
- Numele de articol, client și cont nu pot să lipsească.
 - Data nașterii unui client trebuie specificată, pentru a se cunoaște ce filme poate viziona de pe un cont.
 - Tipul unui abonament și prețul său nu pot fi nule.
 - Atributele de tip dată calendaristică trebuie de obicei specificate.
- d) Constrângerea de UNIQUE (unicitatea altor câmpuri decât cheile primare)
- Numele unui articol trebuie să nu se repete (nu pot exista două filme/ seriale denumite la fel).
 - Numele atribuit fiecarui cont este unic, pentru a se diferenția utilizatorii între ei.
 - Emailurile din baza de date nu se repetă.
 - Fiecare categorie are nume unic.
 - Nu pot fi mai multe abonamente de același tip, deci câmpul de tip este unic.
- e) Constrângerea de CHECK (cu o condiție de verificat, creată special pentru baza de date)
- Data de oprire a unei accesări de film/ serial va fi întotdeauna anterioară datei de pornire.
 - Data expirării plășii va fi anterioară celei de realizare.
 - Procentul parcurs dintr-un articol nu poate fi mai mare de 100%.
 - Ratingul atribuit în cadrul unei recenzii are valoarea mai mică decât 5.

3. Entitățile

ARTICOL (cheie primară: id_articol)

Este entitatea principală a bazei de date proiectate, reprezentând produsul pe care il furnizează platforma. Ea pastrează atributele comune pentru cele două subentități ale sale, adică cele două tipuri de articole (film și serial): cheia primară (id_articol), numele articolului, care este unic pentru că nu există două filme/seriale care să se numească la fel, vârsta minimă recomandată pentru publicul căruia i se adresează și producătorul.

FILM (cheie primară specifică unui articol: id_articol)

Filmul este un articol, astfel că în baza de date este o subentitate a celei de mai sus, moștenind atributele sale. Acest lucru este posibil prin transformarea id-ului de articol în cheie primară în , precum și cheie externă către supraentitate. Pentru un film sunt relevante: durata, lista actorilor, directorul, data lansării și bugetul folosit pentru realizare.

SERIAL (cheie primară specifică unui articol: id_articol)

De asemenea un articol, SERIAL devine o subentitate înglobată în tabela de articole, iar cheia sa primară este id-ul din această tabelă. Pentru un serial se cunosc numărul de sezoane apărute până la data curentă (chiar dacă pe platformă nu au fost adăugate încă toate sezoanele și episoadele) și data inițială a lansării sale.

SEZON (cheie primară compusă: nr_sezon și id_articol)

Entitatea sezon depinde de un serial, cheia sa primară fiind compusă din numărul sezonului și id-ul serialului. Cum un film nu poate fi împărțit în sezoane, această tabelă nu se poate lega de cea de articole. În tabela SEZON se specifică actorii și directorul, aceștia putând să difere de la un sezon la altul.

EPISOD (cheie primară compusă: nr_episod și cheie primară sezon)

Un episod are propriul său număr, însă depinde de un sezon dintr-un serial, astfel că are o cheie primară compusă din cele ale entităților de care depinde. Pentru un episod, se poate specifica numele, data lansării (care poate să difere de data lansării serialului), precum și durata.

TRAILER (cheie primară: id trailer)

Fiecare film sau serial are pe platformă un trailer și numai unul (adică cel oficial). Tabela depinde de ARTICOL printr-o cheie externă. Pentru acesta, producătorul poate să difere față de producătorul articolului. Numele trailerului nu este întotdeauna specificat.

CATEGORIE (cheie primară: id categorie)

Fiecare articol este inclus într-o categorie (documentar, acțiune, romantic, thriller, horror, comedie etc.). Categoria are o cheie primară id, are un nume propriu, o descriere etc. și nu este dependentă de nicio altă entitate.

CLIENT (cheie primară: cod client)

Un client este identificat unic printr-un cod, cu ajutorul căruia își va crea unul sau mai multe conturi. Pentru un client se știe numele său, emailul (unic) și data nașterii (necesară pentru a viziona articole dintr-o anumită categorie de vârstă).

CONT (cheie primară: id cont)

Contul îi aparține unui client, dar nu este unic pentru el. De pe cont se vizionează articole, se postează recenzii și se fac plăti, astfel că se specifică și detaliile de card în tabelă. Un cont este inactiv dacă nu s-a cumpărat pentru el niciun abonament sau dacă plata pentru abonament a expirat și redevine activ la o nouă plată. Tabela depinde de CLIENT.

RECENZIE (cheie primară: nr recenzie)

O recenzie este făcută de pe un cont pentru un articol. Tabelele menționate se află în relație multiplă pentru unicitatea tripletelor. Recenzia are un comentariu optional, însă ratingul pe care îl transmite pentru un articol este obligatoriu.

ABONAMENT (cheie primară: nr abonament)

De pe un cont se poate face un abonament, identificat printr-un număr de abonament. Un abonament este de un anume tip: standard, premium, pentru copii etc., are atribuit un preț care este pentru plată lunară/ o data la două luni/ anuală. Tipul abonamentului determină reclamele care vor apărea: acestea reprezintă de fapt oferte și recomandări de filme și seriale noi.

RECLAMA (cheie primară: nr reclamă)

Reclama este inclusă într-un abonament, ea fiind specifică tipului de abonare ales. De exemplu, dacă s-a optat pentru un abonament standard, atunci vor apărea reclame referitoare la filme incluse în pachetul standard sau reclame cu oferte de îmbunătățire a abonamentului cumpărat: noi facilități contra cost.

Entități noi prezente în diagrama conceptuală:**ACCESEAZA_FILM** (cheie primară compusă: id cont, id articol și nr accesare)

Pentru a se putea ulterior crea rapoarte referitoare la ce s-a vizionat de pe fiecare cont, dar și pentru a sparge relația dintre cont și film, e nevoie de o tabelă de accesări pentru filme. Deoarece un cont poate accesa același film de mai multe ori, se introduce și atributul #nr_accesare_film, care diferențiază toate vizionările între ele.

ACCESEAZA_SERIAL (cheie primară compusă: id cont, cheie primară episod și nr accesare)

Această tabelă este similară celei de mai sus, dar se referă la vizionarea unui serial, deci conținează episodul ales și sezonul din care face parte acesta. Se vor reține atât procentul parcurs din episodul curent, cât și procentul din întreg serialul.

APARTINE(cheie primară compusă: id_articol și id_categorie)

Entitatea este introdusă pentru a elimina relația de mai multe articole la mai multe categorii și nu are attribute proprii adăugate.

POSTEAZA(cheie primară compusă: id cont, id articol și id postare)

Această entitate apare pentru a sparge relația dintre trei entități, de care depinde, legând un cont de un articol căruia îi relizează recenzia. Fiecare postare corespunde unei recenzi și reține data la care a fost urcată aceasta pe platformă.

PLATESTE(cheie primară compusă din cheie primară cont, nr abonament și id plată)

Această tabelă este dependentă de cont și de abonament: pentru o plată contează contul de pe care se realizează, dar și de abonamentul căruia îi e destinată. Înținându-se cont de suma de bani dată, plata are o durată de valabilitate atribuită, reprezentată de diferența dintre data realizării ei și data de expirare. Pentru o sumă egală cu prețul lunar al abonamentului, se consideră data expirării la exact o lună de la data plășii.

3. Relațiile dintre entități

a) Relații de tipul unu la unu

- Un articol are un singur trailer (1 : 1)

b) Relații de tipul unu la mai multe/ mai multe la unul

- Un serial are cel puțin un sezon (1 : M(1))
- Un sezon are unul sau mai multe episoade (1 : M(1))
- Un client trebuie să aibă un cont pentru a viziona filme, dar poate avea și mai multe conturi (1 : M(1))
- Un abonament are atașate una sau mai multe reclame (1 : M(1))

c) Relații de tipul mai multe la mai multe

! Aceste relații nu vor fi permise în modelul conceptual al proiectului, ele trebuind să fie separate în două relații de tipuri diferite.

- Una sau multe articole pot să aparțină de una sau mai multe categorii (M(0) : M(1))
- De pe mai multe conturi se pot viziona mai multe filme sau episoade din seriale (M(0) : M(0))
- Mai multe conturi optează pentru mai multe abonamente; un cont trebuie să plătească pentru minimum un abonament ca să fie activ, poate achiziționa oricâte (M(0) : M(1))

d) Relație între trei entități

- CONT-ARTICOL-RECENZIE
- Tripletul trebuie să fie unic
- Mai multe recenzii diferite pot fi posteate de pe un cont pentru un articol (M(0):1:1)

5. Atributele

Atributele pot reține informații de tipul unor siruri de caractere-VARCHAR2 (de exemplu, numele unui articol sau al unui client), numere-NUMBER (de exemplu, ratingul pentru o recenzie sau suma de plată pentru un abonament) sau date calendaristice-DATE (de exemplu, data nașterii a unui client sau data de lansare a unui film). În particular, pentru date calendaristice se va folosi și TIMESTAMP, care reține și ora după un format ales (pentru accesări de filme sau seriale trebuie să se cunoască atât data, cât și ora de pornire și de oprire a accesării).

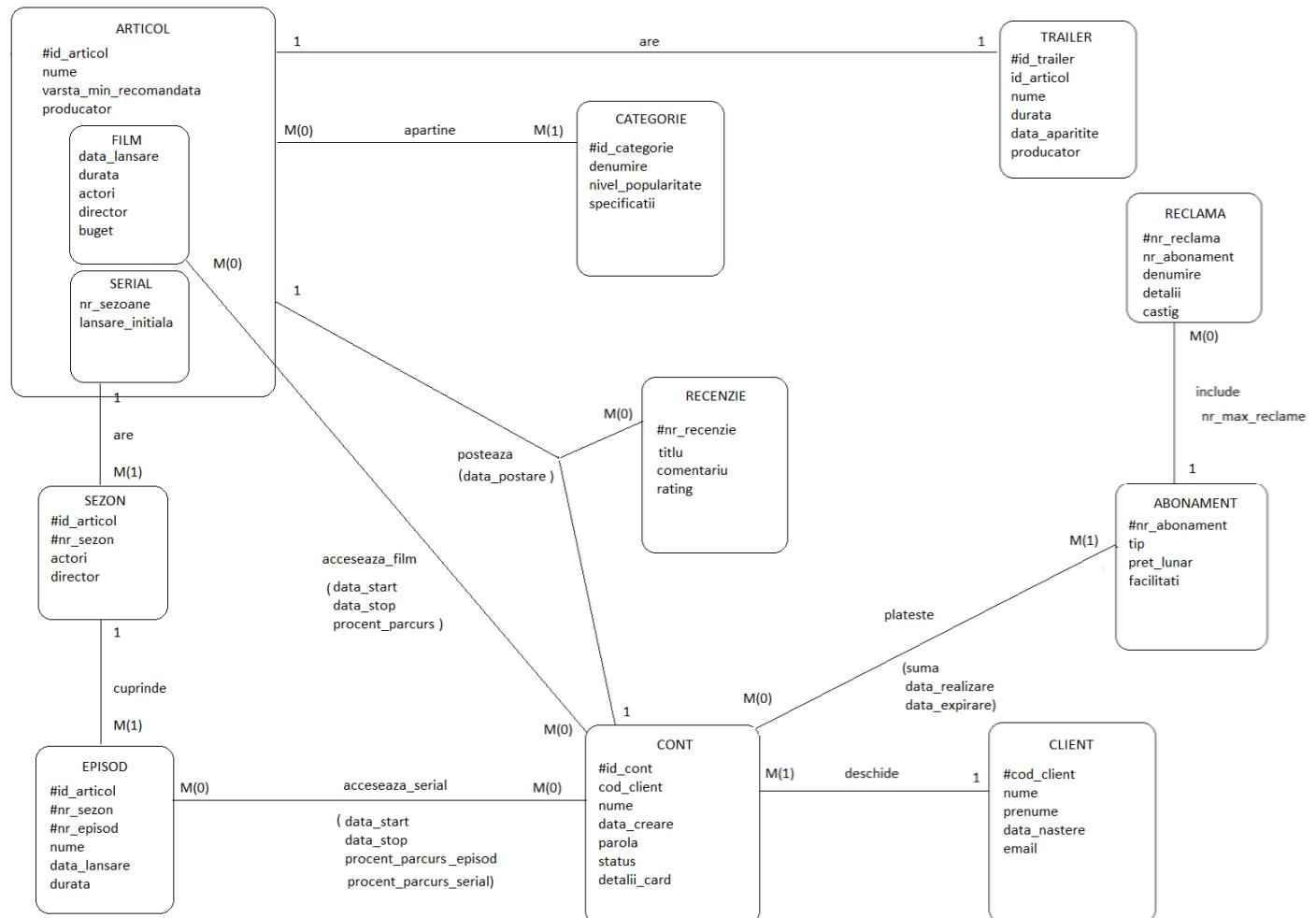
Atributele care au valoare obligatorie și unică sunt cheile primare ale entităților. Pentru celelalte se pot adăuga constrângeri pentru a le asigura unicitatea, existența unei valori sau îndeplinirea unei condiții specifice.

Atributele specificate pentru legătura dintre tabela de CONT și ABONAMENT, precum și cele pentru relația dintre CONT și FILM/EPISOD sunt cele care definesc entitățile PLATESTE, ACCESEAZA_FILM și ACCESEAZA_SERIAL, create la spargerea relațiilor de mai multe la mai multe. Totodată, atributul data_postare specificat pentru relația dintre trei entități va fi înglobat în tabela POSTEaza, apărută pentru a sparge această relație în modelul conceptual.

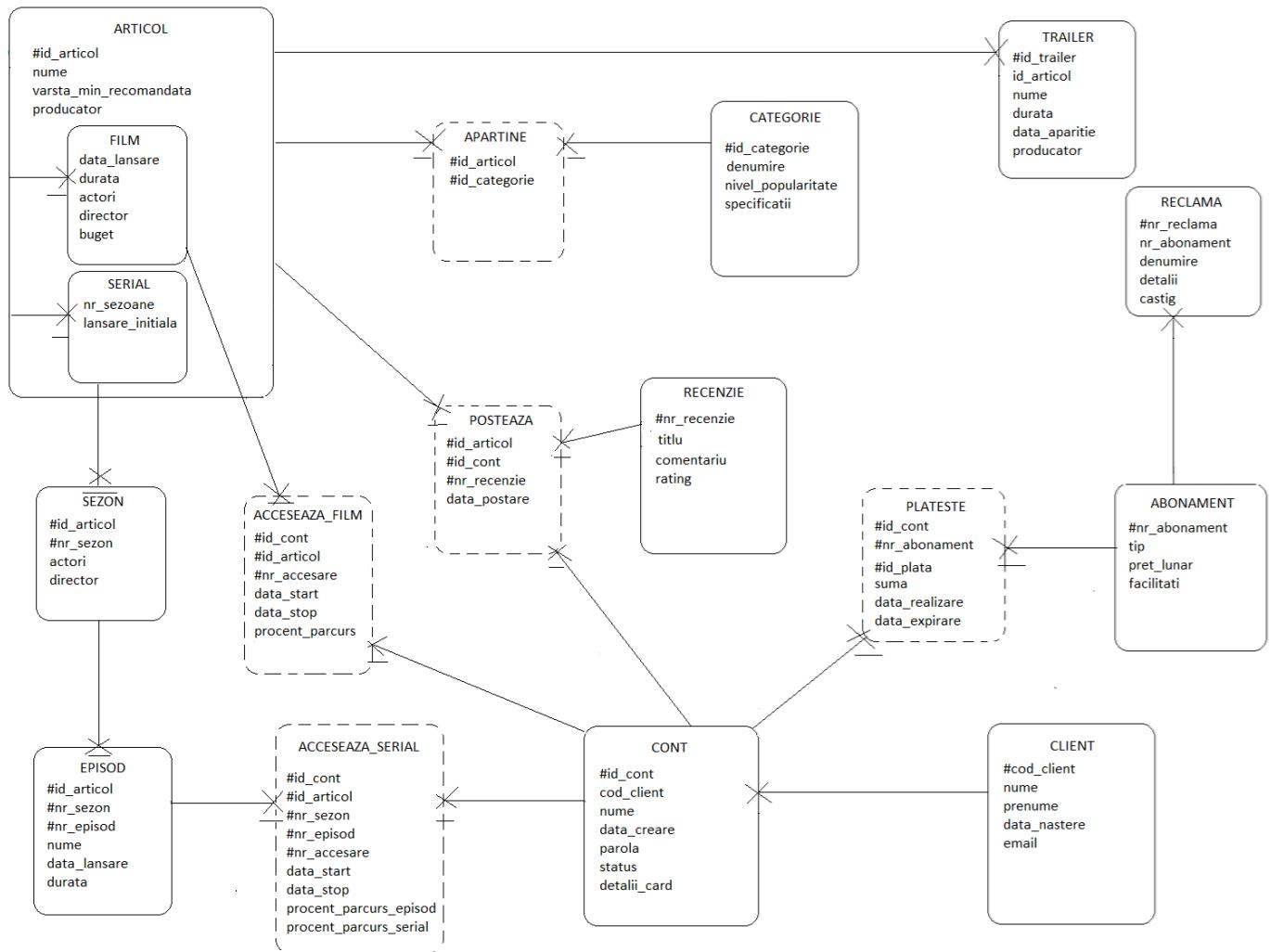
Unele atribute sunt optionale, adică în baza de date ele pot avea valoarea NULL, nefiind specificate pentru toate înregistrările. Printre acestea se află câmpul *specificatii* din tabela CATEGORIE sau *comentariu* din RECENZIE. Alte cazuri sunt: denumirea unei reclame, procentul parcurs dintr-un articol la o accesare (deoarece se poate accesa fără o vizionare efectivă a filmului sau serialului) etc.

Există atribute care primesc valori default, cum ar fi data_start și data_stop dintr-o accesare de film/ serial, care au automat data și ora curentă de la introducerea datelor în bază. De asemenea, pentru o plată, suma trimisă are by default valoarea 0, sau, pentru cont, statusul este inițial *inactiv*.

6. Diagrama e/r



7. Diagrama conceptuală



8. Schemele relaționale

ARTICOL(#id_articol, nume, varsta_min_recomandata, producator)

FILM(#id_articol, nume, varsta_min_recomandata, producator, data_lansare, durata, actori, director, buget)

SERIAL(#id_articol, nume, varsta_min_recomandata, producator, nr_sezoane, lansare_initiala)

SEZON(#id_articol, #nr_sezon, actori, director)

EPISOD(#id_articol, #nr_sezon, #nr_episod, nume, data_lansare, durata)

TRAILER(#id_trailer, id_articol, nume, durata, data_aparitie, producator)

CATEGORIE(#id_categorie, denumire, nivel_popularitate, specificatii)

APARTINE(#id_articol, #id_categorie)

CLIENT(#cod_client, nume, prenume, data_nastere, email)

CONT(#id_cont, cod_client, nume, data_creare, parola, status, detalii_card)

RECENZIE(#nr_recenzie, titlu, comentariu, rating)

POSTEaza(#id_articol, #id_cont, #nr_recenzie, data_postare)

ABONAMENT(#nr_abonament, tip, pret_lunar, facilitati)

RECLAMA(#nr_reclama, nr_abonament, denumire, detalii, castig)

PLATESTE(#id_cont, #nr_abonament, #id_plata, suma, data_realizare, data_expirare)

ACCESEAZA_FILM(#id_cont, #id_articol, #nr_accesare, data_start, data_stop, procent_parcurs)

ACCESEAZA_SERIAL(#id_cont, #id_articol, #nr_sezon, #nr_episod, #nr_accesare, data_start, data_stop, procent_parcurs_episodes, procent_parcurs_serial)

9. Normalizarea

FN1

Pentru a se respecta forma normală 1, trebuie ca niciun tabel să nu conțină grupuri repetitive.

Fie entitatea CLIENT care are un cod de client, nume, prenume, data_nastere, email și cont.

Deoarece un client își poate crea mai multe conturi, ar însemna că grupul care se repetă sunt informațiile despre cont pentru fiecare client.

Pentru normalizare, se elimină din tabelă datele despre cont și tabela inițială devine:

CLIENT(#cod_client, nume, prenume, data_nastere, email)

Datele conturilor sunt păstrate într-o nouă entitate, legată de CLIENT prin cheie străină (cod_client). Relația este de mai multe conturi la un client.

CONT(#id_cont, cod_client, nume, data_creare, parola, status, detalii_card)

FN2

Pentru forma normală 2, trebuie îndeplinite condițiile pentru prima formă normală și fiecare atribut care nu este cheie primară să fie total dependent funcțional de cheia primară.

Fie tabela următoare care **nu** îndeplinește FN2, dar este în FN1:

ARTICOL(#id_articol, nume, varsta_min_recomandata, producator, denumire_categorie, nivel_popularitate_categorie, specificatii_categorie)

Atributele denumire_categorie, nivel_popularitate_categorie și specificatii_categorie nu depind de id-ul articolului. Se produce normalizarea prin crearea unei tabele CATEGORIE, care înglobează datele respective. Toate acestea vor depinde total de #id_categorie.

Tabelele devin:

ARTICOL(#id_articol, nume, varsta_min_recomandata, producator)

CATEGORIE(#id_categorie, denumire, nivel_popularitate, specificatii)

FN3

Trebuie respectate condițiile pentru a doua formă normală și, în plus, niciun atribut care nu e cheie primară nu poate fi dependent tranzitiv de cheia primară.

Se consideră entitatea care nu respectă forma normală 3, dar le respectă pe celelalte două:

ARTICOL(#id_articol, nume_articol, varsta_min_recomandata, producator_articol, id_trailer, nume_trailer, durata_trailer, data_aparitie_tariler, producator_tariler)

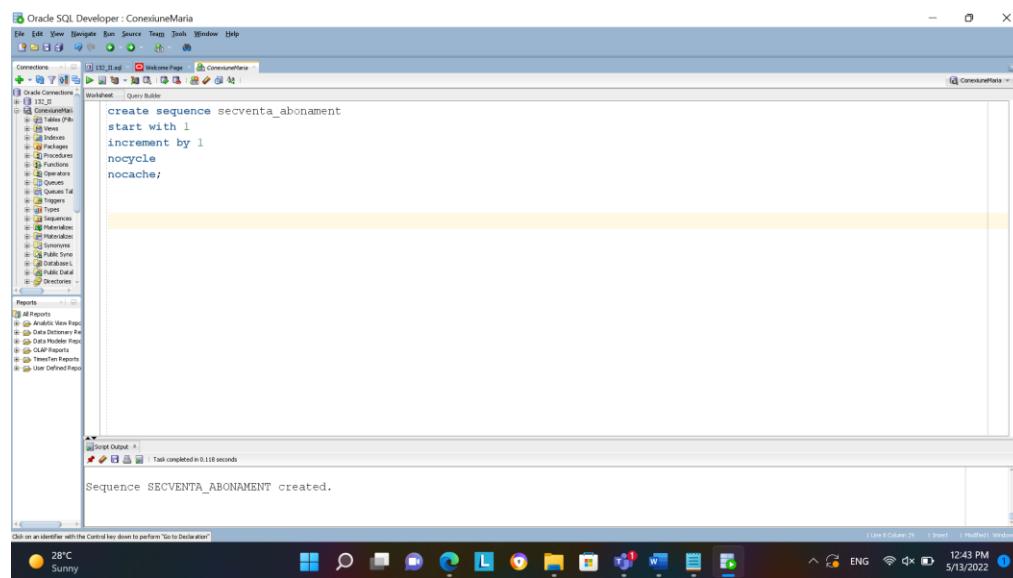
Audem dependențe tranzitive: nume_trailer/ durata_trailer/ data_aparitie_tariler/ producator_trariler depind de id_trailer, iar acesta este dependent de id_articol. Se vor separa cele două entități astfel:

ARTICOL(#id_articol, nume, varsta_min_recomandata, producator)
TRAILER(#id_trailer, id_articol, nume, durata, data_aparitie, producator)

În urma exemplificării normalizării, se observă că diagrama conceptuală a proiectului respectă toate formele normale ale unei baze de date.

10. Secvențe

```
create sequence secventa_abonament
start with 1
increment by 1
nocycle
nocache;
```



```

create sequence secenta_articol
start with 1
increment by 1
nocycle
nocache;

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a script is being run:

```

create sequence secenta_articol
start with 1
increment by 1
nocycle
nocache;

```

Below the workspace, the output window displays the confirmation message:

Sequence SECVENTA_ARTICOL created.

The status bar at the bottom indicates the task completed in 0.111 seconds.

```

create sequence secenta_acceseaza
start with 1
increment by 1
nocycle
nocache;

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a script is being run:

```

create sequence secenta_acceseaza
start with 1
increment by 1
nocycle
nocache;

```

Below the workspace, the output window displays the confirmation message:

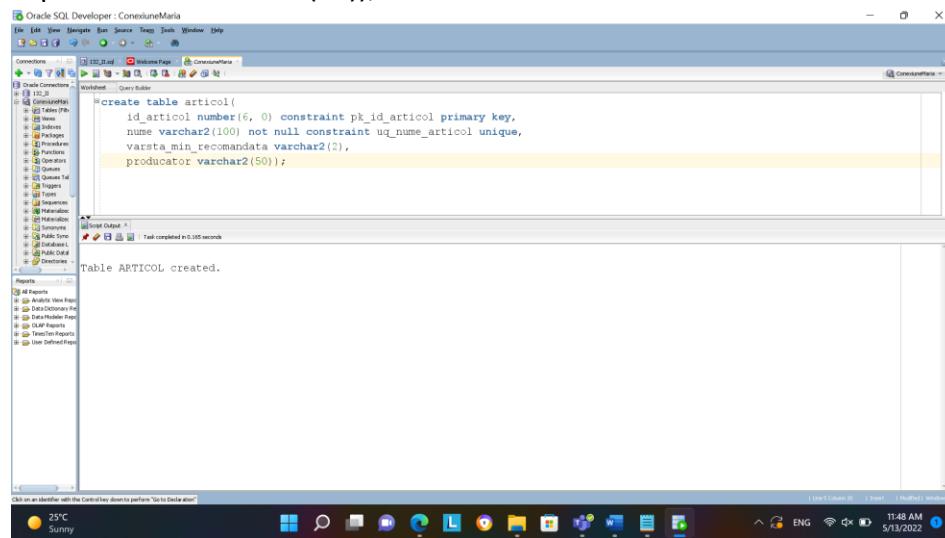
Sequence SECVENTA_ACCESEAZA created.

The status bar at the bottom indicates the task completed in 0.047 seconds.

11. Creare tabele

ARTICOL

```
create table articol(
    id_articol number(6, 0) constraint pk_id_articol primary key,
    nume varchar2(100) not null constraint uq_nume_articol unique,
    varsta_min_recomanda varchar2(2),
    producator varchar2(50));
```



FILM

```
create table film(
    id_articol number(6, 0) constraint pk_id_film primary key,
    data_lansare date default sysdate not null,
    durata number(3, 2) not null,
    actori varchar2(3000),
    director varchar2(50),
    buget number(7, 0),
    constraint fk_id_film_articol foreign key(id_articol) references articol(id_articol));
```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query window displays the SQL code for creating the FILM table:

```
create table film(
    id_articol number(6, 0) constraint pk_id_film primary key,
    data_lansare date default sysdate not null,
    durata number(3, 2) not null,
    actori varchar2(3000),
    director varchar2(50),
    buget number(7, 0),
    constraint fk_id_film_articol foreign key(id_articol) references articol(id_articol));
```

Below the code, the message "Table FILM created." is displayed. The bottom status bar shows the system time as 3:54 PM on 5/14/2022.

SERIAL

```
create table serial(
    id_articol number(6, 0) constraint pk_id_serial primary key,
    nr_sezoane number(2),
    lansare_initiala date not null,
    constraint fk_id_serial_articol foreign key(id_articol) references articol(id_articol));
```

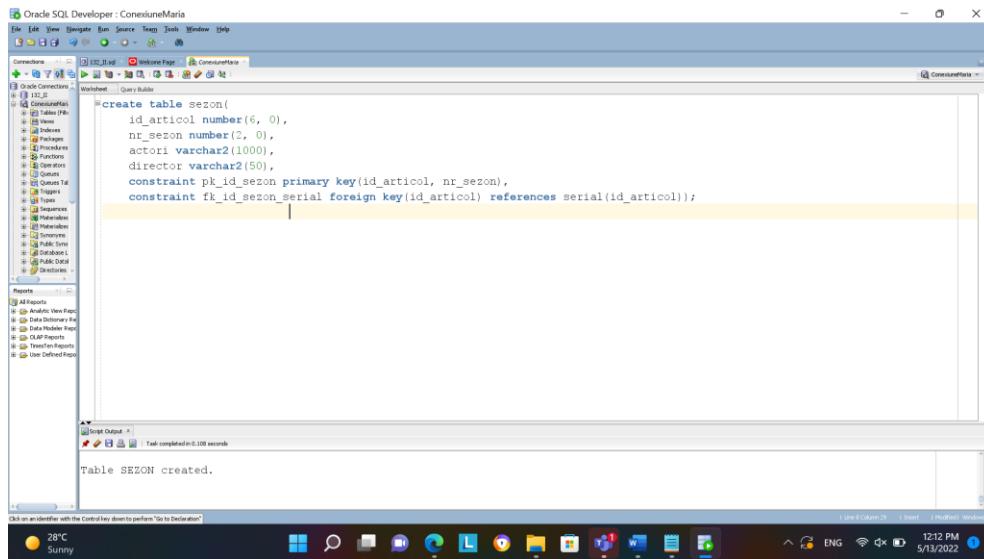
The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query window displays the SQL code for creating the SERIAL table:

```
create table serial(
    id_articol number(6, 0) constraint pk_id_serial primary key,
    nr_sezoane number(2),
    lansare_initiala date not null,
    constraint fk_id_serial_articol foreign key(id_articol) references articol(id_articol));
```

Below the code, the message "Table SERIAL created." is displayed. The bottom status bar shows the system time as 12:04 PM on 5/13/2022.

SEZON

```
create table sezon(
    id_articol number(6, 0),
    nr_sezon number(2, 0),
    actori varchar2(1000),
    director varchar2(50),
    constraint pk_id_sezon primary key(id_articol, nr_sezon),
    constraint fk_id_sezon_serial foreign key(id_articol) references serial(id_articol));
```



EPISOD

```
create table episod(
    id_articol number(6, 0),
    nr_sezon number(2, 0),
    nr_episod number(2, 0),
    nume varchar2(100),
    data_lansare date default sysdate not null,
    durata number(3, 2) not null,
    constraint pk_id_episod primary key(id_articol, nr_sezon, nr_episod),
    constraint fk_id_episod_sezon foreign key(id_articol, nr_sezon) references sezon(id_articol,
    nr_sezon));
```

Oracle SQL Developer : ConnexineMaria

```
create table episod(
    id_articol number(6, 0),
    nr_sezon number(2, 0),
    nr_episod number(2, 0),
    nume varchar2(100),
    data_lansare date default sysdate not null,
    durata number(3, 2) not null,
    constraint pk_id_episod primary key(id_articol, nr_sezon, nr_episod),
    constraint fk_id_episod_sezon foreign key(id_articol, nr_sezon) references sezon(id_articol, nr_sezon));
```

Table EPISOD created.

9°C
Cloudy

TRAILER

```
create table trailer(
    id_trailer number(6, 0) constraint pk_id_trailer primary key,
    id_articol number(6, 0),
    nume varchar2(100),
    durata number(3, 2) not null,
    data_aparitie date not null,
    producator varchar2(50),
    constraint fk_trailer_articol foreign key(id_articol) references articol(id_articol));
```

Oracle SQL Developer : ConnexineMaria

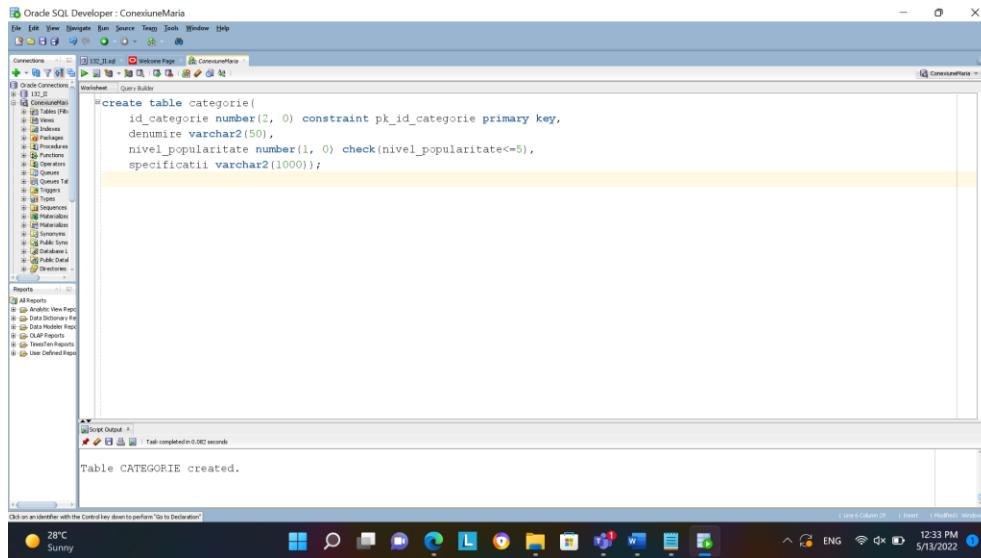
```
create table trailer(
    id_trailer number(6, 0) constraint pk_id_trailer primary key,
    id_articol number(6, 0),
    nume varchar2(100),
    durata number(3, 2) not null,
    data_aparitie date not null,
    producator varchar2(50),
    constraint fk_trailer_articol foreign key(id_articol) references articol(id_articol));
```

Table TRAILER created.

9°C
Cloudy

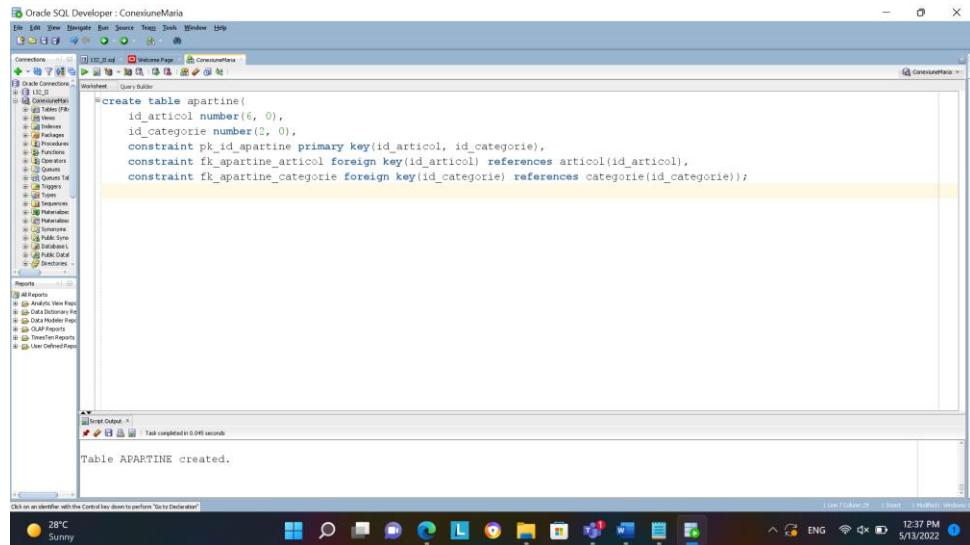
CATEGORIE

```
create table categorie(
    id_categorie number(2, 0) constraint pk_id_categorie primary key,
    denumire varchar2(50),
    nivel_popularitate number(1, 0) check(nivel_popularitate<=5),
    specificatii varchar2(1000));
```



APARTINE

```
create table apartine(
    id_articol number(6, 0),
    id_categorie number(2, 0),
    constraint pk_id_apartine primary key(id_articol, id_categorie),
    constraint fk_apartine_articol foreign key(id_articol) references articol(id_articol),
    constraint fk_apartine_categorie foreign key(id_categorie) references
    categorie(id_categorie));
```



Oracle SQL Developer : ConecioneMaria

File Edit View Navigate Run Source Test Tools Window Help

Connections ConecioneMaria

Worksheet Query Builder

```
create table apartine(
    id_articol number(6, 0),
    id_categorie number(2, 0),
    constraint pk_id_apartine primary key(id_articol, id_categorie),
    constraint fk_apartine_articol foreign key(id_articol) references articol(id_articol),
    constraint fk_apartine_categorie foreign key(id_categorie) references categorie(id_categorie));
```

Script Output X Task completed in 0.000 seconds

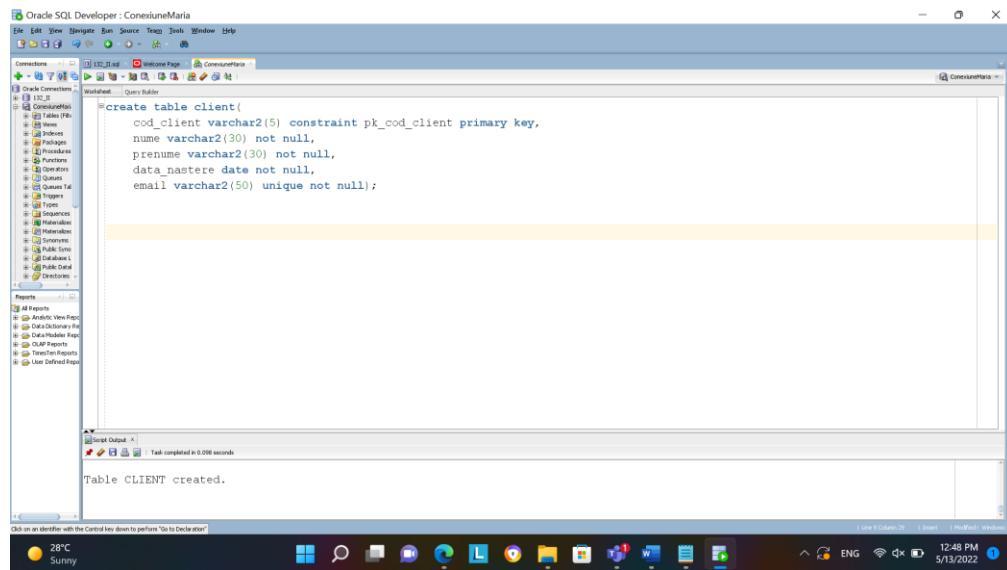
Table APARTINE created.

Click on an identifier with the Control key down to perform "Go to Declaration".

28°C Sunny 12:37 PM 5/13/2022 ENG

CLIENT

```
create table client(
    cod_client varchar2(5) constraint pk_cod_client primary key,
    nume varchar2(30) not null,
    prenume varchar2(30) not null,
    data_nastere date not null,
    email varchar2(50) unique not null);
```



Oracle SQL Developer : ConecioneMaria

File Edit View Navigate Run Source Test Tools Window Help

Connections ConecioneMaria

Worksheet Query Builder

```
create table client(
    cod_client varchar2(5) constraint pk_cod_client primary key,
    nume varchar2(30) not null,
    prenume varchar2(30) not null,
    data_nastere date not null,
    email varchar2(50) unique not null);
```

Script Output X Task completed in 0.000 seconds

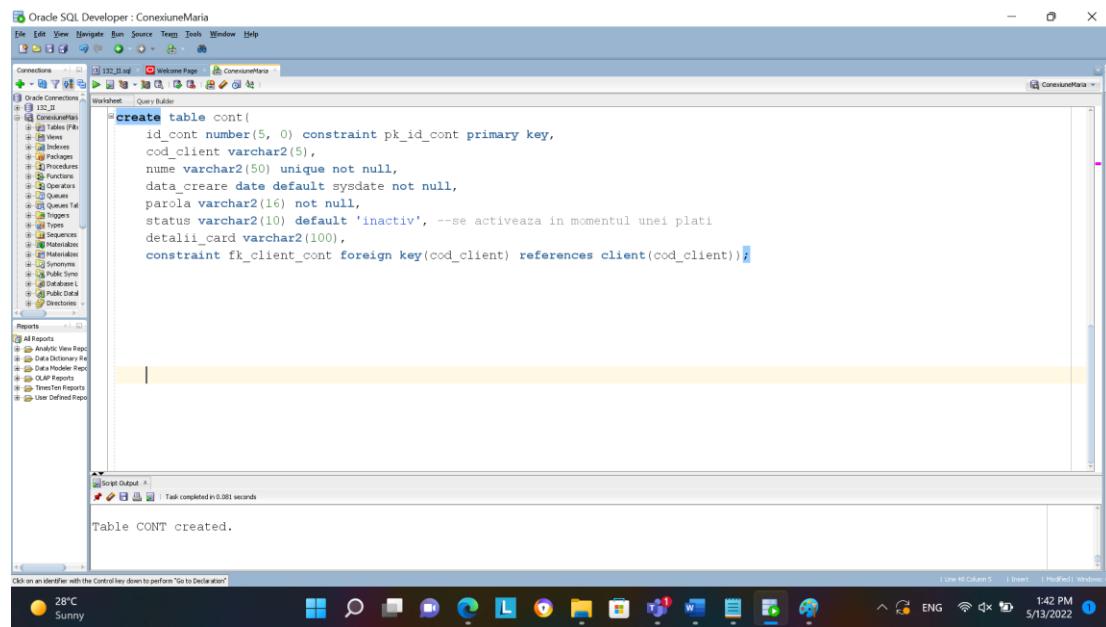
Table CLIENT created.

Click on an identifier with the Control key down to perform "Go to Declaration".

28°C Sunny 12:48 PM 5/13/2022 ENG

CONT

```
create table cont(
    id_cont number(5, 0) constraint pk_id_cont primary key,
    cod_client varchar2(5),
    nume varchar2(50) unique not null,
    data_creeare date default sysdate not null,
    parola varchar2(16) not null,
    status varchar2(10) default 'inactiv', --se activeaza in momentul unei plati
    detalii_card varchar2(100)
constraint fk_client_cont foreign key(cod_client) references client(cod_client));
```



ACCESEAZA_FILM

```
create table acceseaza_film(
    id_cont number(5, 0),
    id_articol number(6, 0),
    nr_accesare number(8, 0),
    data_start timestamp default sysdate not null,
    data_stop timestamp default sysdate not null,
    procent_parcurs number(5, 2) check(procent_parcurs<=100),
    constraint ck_start_stop check(data_stop>=data_start),
    constraint pk_accesaza_film primary key(id_cont, id_articol, nr_accesare),
```

```

constraint fk_cont_acceseaza_film foreign key(id_cont) references cont(id_cont),
constraint fk_film_acceseaza_film foreign key(id_articol) references film(id_articol));

```

The screenshot shows the Oracle SQL Developer interface. A query window is open with the following SQL code:

```

create table acceseaza_film(
    id_cont number(5, 0),
    id_articol number(6, 0),
    nr_accesare number(8, 0),
    data_start timestamp default sysdate not null,
    data_stop timestamp default sysdate not null,
    procent_parcurs number(5, 2) check(procent_parcurs<=100),
    constraint ck_start_stop check(data_stop>=data_start),
    constraint pk_acceseaza_film primary key(id_cont, id_articol, nr_accesare),
    constraint fk_cont_acceseaza_film foreign key(id_cont) references cont(id_cont),
    constraint fk_film_acceseaza_film foreign key(id_articol) references film(id_articol));

```

The code is highlighted in yellow. Below the code, the message "Table ACCESEAZA_FILM created." is displayed. The status bar at the bottom right shows "5/16/2022 5:18 PM".

ACCESEAZA_SERIAL

```

create table acceseaza_serial(
    id_cont number(5, 0),
    id_articol number(6, 0),
    nr_sezon number(2, 0),
    nr_episod number(2, 0),
    nr_accesare number(8, 0),
    data_start timestamp default sysdate not null,
    data_stop timestamp default sysdate not null,
    procent_parcurs_episodes number(5, 2) check(procent_parcurs_episodes<=100),
    procent_parcurs_serial number(5, 2) check(procent_parcurs_serial<=100),
    constraint ck_start_stop1 check(data_stop>=data_start),
    constraint pk_acceseaza_serial primary key(id_cont, id_articol, nr_sezon, nr_episodes,
    nr_accesare),
    constraint fk_cont_acceseaza_serial foreign key(id_cont) references cont(id_cont),
    constraint fk_episodes_acceseaza_serial foreign key(id_articol, nr_sezon, nr_episodes) references
    episodes(id_articol, nr_sezon, nr_episodes));

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a query window containing the following SQL code:

```
create table accezeaza_serial(
    id_cont number(5, 0),
    id_articol number(6, 0),
    nr_sezon number(2, 0),
    nr_episod number(2, 0),
    nr_accesare number(8, 0),
    data_start timestamp default sysdate not null,
    data_stop timestamp default sysdate not null,
    procent_parcurs_episod number(5, 2) check(procent_parcurs_episod<=100),
    procent_parcurs_serial number(5, 2) check(procent_parcurs_serial<=100),
    constraint pk_accezeaza_serial primary key(id_cont, id_articol, nr_sezon, nr_episod, nr_accesare),
    constraint fk_cont_accezeaza_serial foreign key(id_cont) references cont(id_cont),
    constraint fk_episod_accezeaza_serial foreign key(id_articol, nr_sezon, nr_episod) references episod(id_articol, nr_sezon, nr_episod));

```

Below the code, the script output shows the message: "Table ACCESEAZA_SERIAL created." The status bar at the bottom indicates the task completed in 0.075 seconds.

ABONAMENT

create table abonament(

```
    nr_abonament number(6, 0) constraint pk_nr_abonament primary key,
    tip varchar2(20) unique not null,
    pret_lunar number(6, 2) not null,
    facilitati varchar2(1000));
```

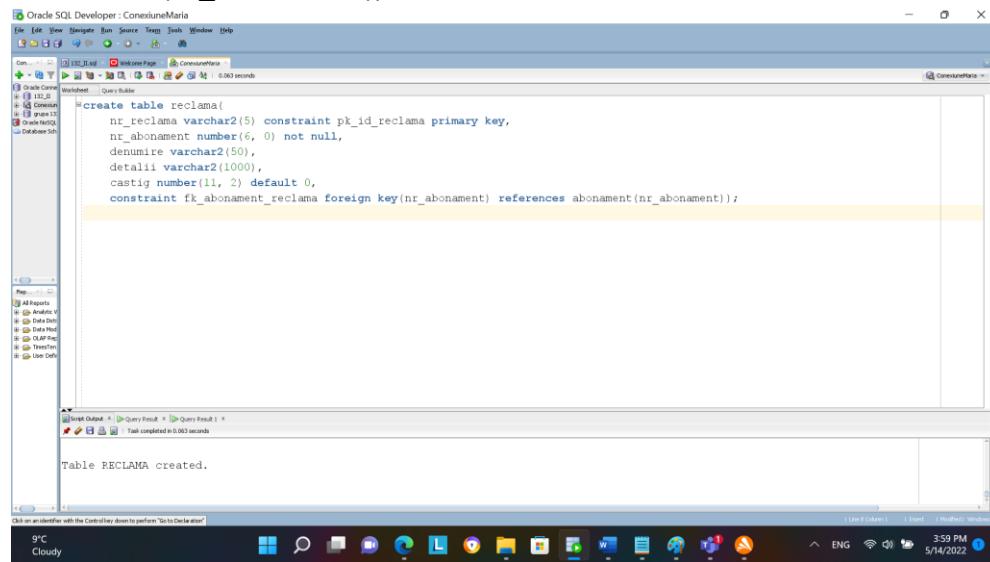
The screenshot shows the Oracle SQL Developer interface. In the central workspace, there is a query window containing the following SQL code:

```
create table abonament(
    nr_abonament number(6, 0) constraint pk_nr_abonament primary key,
    tip varchar2(20) unique not null,
    pret_lunar number(6, 2) not null,
    facilitati varchar2(1000));
```

Below the code, the script output shows the message: "Table ABONAMENT created." The status bar at the bottom indicates the task completed in 0.025 seconds.

RECLAMA

```
create table reclama(
    nr_reclama varchar2(5) constraint pk_id_reclama primary key,
    nr_abonament number(6, 0) not null,
    denumire varchar2(50),
    detalii varchar2(1000),
    castig number(11, 2) default 0,
    constraint fk_abonament_reclama foreign key(nr_abonament) references
abonament(nr_abonament));
```



PLATESTE

```
create table plateste(
    id_cont number(5, 0),
    nr_abonament number(6, 0),
    id_plata varchar2(6),
    suma number(6, 2) default 0,
    data_realizare date not null,
    data_expirare date not null,
    constraint ck_realizare_expirare check(data_expirare>data_realizare),
    constraint pk_plateste primary key(id_cont, nr_abonament, id_plata),
    constraint fk_cont_plateste foreign key(id_cont) references cont(id_cont),
    constraint fk_abonament_plateste foreign key(nr_abonament) references
abonament(nr_abonament));
```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query window displays the SQL code for creating the PLATESTE table:

```

create table plateste(
    id_cont number(5, 0),
    nr_abonament number(6, 0),
    id_plata varchar2(6),
    suma number(6, 2) default 0,
    data_realizare date not null,
    data_expirare date not null,
    constraint ck_realizare_expirare check(data_expirare>data_realizare),
    constraint pk_plateste primary key(id_cont, nr_abonament, id_plata),
    constraint fk_cont_plateste foreign key(id_cont) references cont(id_cont),
    constraint fk_abonament_plateste foreign key(nr_abonament) references abonament(nr_abonament));

```

Below the code, the message "Table PLATESTE created." is displayed. The status bar at the bottom right shows the date and time: 5/14/2022 4:00 PM.

RECENZIE

```

create table recenzie(
    nr_recenzie number(6, 0) constraint pk_nr_recenzie primary key,
    titlu varchar2(30),
    comentariu varchar2(500),
    rating number(3, 2) not null check(rating<=5));

```

The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query window displays the SQL code for creating the RECENZIE table:

```

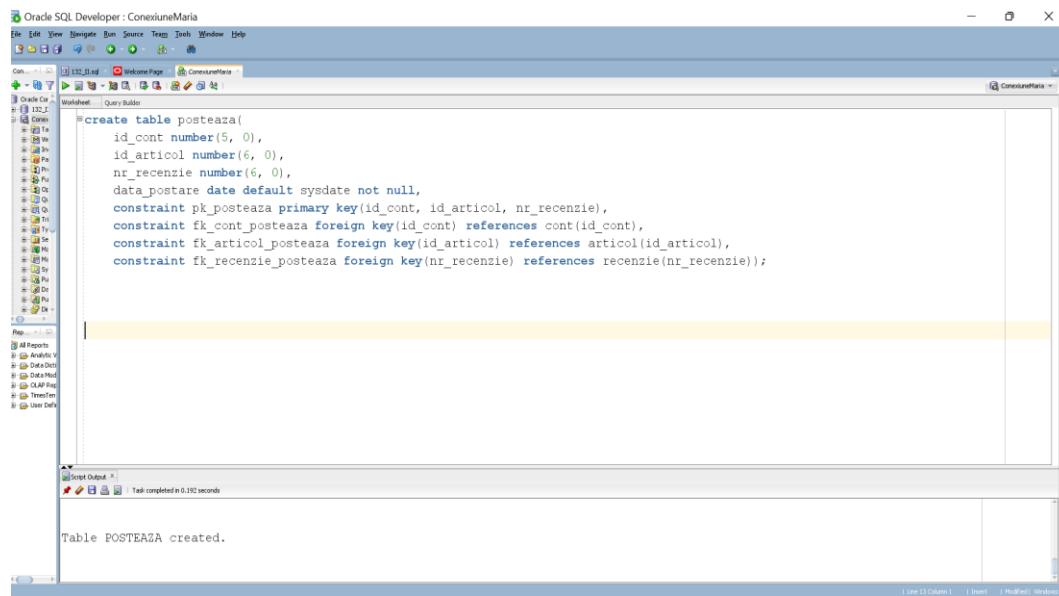
create table recenzie(
    nr_recenzie number(6, 0) constraint pk_nr_recenzie primary key,
    titlu varchar2(30),
    comentariu varchar2(500),
    rating number(3, 2) not null check(rating<=5));

```

Below the code, the message "Table RECENZIE created." is displayed. The status bar at the bottom right shows the date and time: 5/14/2022 4:00 PM.

POSTEAZA

```
create table posteaza(
    id_cont number(5, 0),
    id_articol number(6, 0),
    nr_recenzie number(6, 0),
    data_postare date default sysdate not null,
    constraint pk_posteaza primary key(id_cont, id_articol, nr_recenzie),
    constraint fk_cont_posteaza foreign key(id_cont) references cont(id_cont),
    constraint fk_articol_posteaza foreign key(id_articol) references articol(id_articol),
    constraint fk_recenzie_posteaza foreign key(nr_recenzie) references recenzie(nr_recenzie));
```



Inserări

```
insert into articol
values(secventa_articol.nextval, 'Pe aripile vantului', NULL, 'Selznick International Pictures');
insert into articol
values(secventa_articol.nextval, 'Inception', NULL, 'Warner Bros. Pictures');
insert into articol
values(secventa_articol.nextval, 'Vikingii', 16, 'Shaw Media');
insert into articol
values(secventa_articol.nextval, 'Urzeala tronurilor', 12, 'HBO Entertainment');
insert into articol
```

```

values(secventa_articol.nextval, 'Diamantul sangeriu', 12, 'Bedford Falls Productions');
insert into articol
values(secventa_articol.nextval, 'Mandrie si prejudecata', NULL, 'Studio Canal');
insert into articol
values(secventa_articol.nextval, 'King Richard', NULL, 'Westbrook Studios');
insert into articol
values(secventa_articol.nextval, 'Dosarele X', 16, 'FOX Films');
insert into articol
values(secventa_articol.nextval, 'Friends', 12, 'Warner Bros. Studios');
insert into articol
values(secventa_articol.nextval, 'Oglinda neagra', 18, 'Netflix Production');
insert into articol
values(secventa_articol.nextval, 'Madagascar', NULL, 'DreamWorks Animation');

```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a tree view of database objects. The main workspace contains a query editor window with the following SQL statement:

```
select * from articol; -- rez
```

Below the query editor is a results grid titled "Query Result". The grid displays 11 rows of data, each corresponding to a movie entry in the ARTICOL table. The columns shown are ID_ARTICOL, NUMAR, VARSTA_PRODUCERANATA, and PUBLICATOR. The data is as follows:

ID_ARTICOL	NUMAR	VARSTA_PRODUCERANATA	PUBLICATOR
1	Te artipile vantului	(null)	Selznick International Pictures
2	21 sau 20	(null)	Warner Bros. Pictures
3	3Vikingii	16	Shaw Media
4	4Urzeala tronurilor	12	HBO Entertainment
5	5Diamantul sangeriu	12	Bedford Falls Productions
6	6Mandrie si prejudecata	(null)	Studio Canal
7	7King Richard	(null)	Westbrook Studios
8	8Dosarele X	16	FOX Films
9	9Friends	12	Warner Bros. Studios
10	10Oglinda neagra	18	Netflix Production
11	11Madagascar	(null)	DreamWorks Animation

```

insert into film
values(1, to_date('December 15, 1939', 'Month dd, yyyy'), 3.7,
      'Clark Gable, Vivien Leigh, Leslie Howard, Olivia de Havilland', 'Victor Fleming',
      3850000);
insert into film
values(2, to_date('July 8, 1939', 'Month dd, yyyy'), 2.35,
      'Leonardo DiCaprio, Ken Watanabe, Joseph Gordon-Levitt, Marion Cotillard',
      'Christopher Nolan', 9870000);
insert into film

```

```

values(5, to_date('December 8, 2006', 'Month dd, yyyy'), 2.5,
      'Leonardo DiCaprio, Jennifer Connelly, Djimon Hounsou', 'Eduardo Serra', 9900000);
insert into film
values(6, to_date('September 16, 2005', 'Month dd, yyyy'), 2.35,
      'Keira Knightley, Matthew Macfadyen, Brenda Blethyn, Donald Sutherland, Tom
Hollander', 'Joe Wright',
      7230000);
insert into film
values(7, to_date('September 2, 2021', 'Month dd, yyyy'), 2.7,
      'Will Smith, Aunjanue Ellis, Saniyya Sidney', 'Reinaldo Marcus Green', 7070000);
insert into film
values(11, to_date('May 27, 2005', 'Month dd, yyyy'), 1.4, NULL, 'Eric Darnell', 7500000);

```

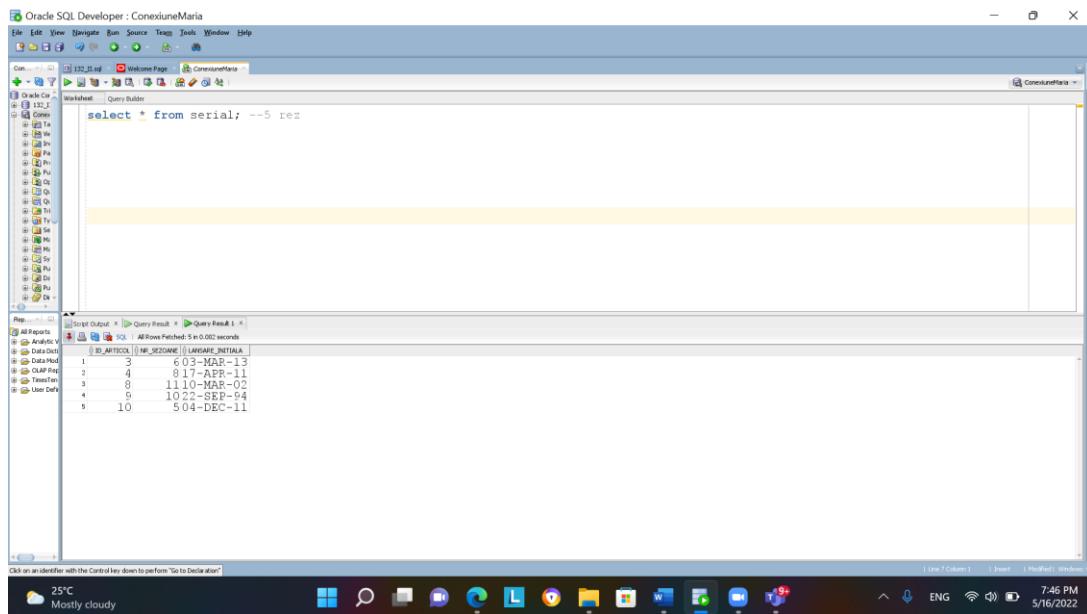
The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of the schema structure under '132.film'. The main workspace contains a query editor with the command: 'select * from film; --6 rez'. Below it is a 'Script Output' window showing the results of the query. The results are presented in a table with columns: ID, ARTISTOL, DATA_LANPAGE, DURATA, ACTOR1, DIRECTOR, and BUDGET. The data rows are:

ID	ARTISTOL	DATA_LANPAGE	DURATA	ACTOR1	DIRECTOR	BUDGET
1	115-DEC-39	3.7	Clark Gable, Vivien Leigh, Leslie Howard, Olivia de Havilland	Victor Fleming	3850000	
2	208-JUL-39	2.35	Leonardo DiCaprio, Ken Watanabe, Joseph Gordon-Levitt, Marion Cotillard	Christopher Nolan	9870000	
3	508-DEC-06	2.5	Leonardo DiCaprio, Jennifer Connelly, Djimon Hounsou	Eduardo Serra	923000	
4	616-SEP-05	2.35	Keira Knightley, Matthew Macfadyen, Brenda Blethyn, Donald Sutherland, Tom Hollander	Joe Wright	7230000	
5	702-SEP-21	2.7	Will Smith, Aunjanue Ellis, Saniyya Sidney	Reinaldo Marcus Green	7070000	
6	1127-MAY-05	1.4 (null)		Eric Darnell	7500000	

```

insert into serial
values(3, 6, to_date('March 3, 2013', 'Month dd, yyyy'));
insert into serial
values(4, 8, to_date('April 17, 2011', 'Month dd, yyyy'));
insert into serial
values(8, 11, to_date('March 10, 2002', 'Month dd, yyyy'));
insert into serial
values(9, 10, to_date('September 22, 1994', 'Month dd, yyyy'));
insert into serial
values(10, 5, to_date('December 4, 2011', 'Month dd, yyyy'));

```



```

insert into client
values('C001', 'Popescu', 'Daria', to_date('March 22, 2002', 'Month dd, yyyy'),
      'dariapopescu@gmail.ro');
insert into client
values('C002', 'Ionescu', 'Marin', to_date('January 03, 1967', 'Month dd, yyyy'),
      'marini67@yahoo.com');
insert into client
values('C003', 'Popovici', 'Andrei', to_date('September 10, 1990', 'Month dd, yyyy'),
      'andrei_popovici@gmail.com');
insert into client
values('C004', 'Anton', 'Bianca', to_date('April 13, 2001', 'Month dd, yyyy'),
      'antonbianca13@hotmail.com');
insert into client
values('C005', 'Dulgheru', 'Delia', to_date('August 07, 1994', 'Month dd, yyyy'),
      'deliadd@gmail.com');
insert into client
values('C006', 'Militaru', 'Ionel', to_date('October 22, 1983', 'Month dd, yyyy'),
      'Mmilitaru@yahoo.com');
insert into client
values('C007', 'Neacsu', 'Alina', to_date('October 22, 2003', 'Month dd, yyyy'),
      'alinaN@yahoo.com');
insert into client
values('C008', 'Barbulescu', 'Stelian', to_date('September 26, 2000', 'Month dd, yyyy'),
      )

```

```

'stelianB@hotmail.com');

insert into client
values('C009', 'Dragomir', 'Mihaela', to_date('January 6, 1975', 'Month dd, yyyy'),
      'dragomir1@gmail.com');

```

The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, it says "Oracle SQL Developer : ConexiuneMaria". The main area has a "Worksheet" tab selected, displaying the query:

```
select * from client; -- rez
```

Below the query, the results are shown in a table titled "Query Result 1". The table has columns: COID_CLIENT, NAME, PREGNUME, DATA_NASTERE, and EMAIL. The data is as follows:

COID_CLIENT	NAME	PREGNUME	DATA_NASTERE	EMAIL
C001	Popescu	Daria	22-MAR-02	dariapopescu@gmail.com
C002	Ionescu	Marin	03-JAN-67	marini67@yahoo.com
C003	Popovici	Andrei	10-SEP-90	andrei.popovici@gmail.com
C004	Anton	Bianca	13-APR-01	antonbianca13@hotmail.com
C005	Dumitru	Delia	07-AUG-84	delia.dumitru@yahoo.com
C006	Mihail	Paul	22-NOV-01	mihailmihail@yahoo.com
C007	Necasu	Alina	22-OCT-03	alinah@yahoo.com
C008	Barbulescu	Stelian	26-SEP-00	stelianb@hotmail.com
C009	Dragomir	Mihaela	06-JAN-75	dragomir1@gmail.com

```

insert into cont(id_cont, cod_client, nume, parola)
values(secventa_cont.nextval,'C001', 'dariap', 'daria');
insert into cont
values(secventa_cont.nextval,'C002', 'marin1',to_date('September 9, 2017', 'Month dd, yyyy'),
      'parolamarin', 'activ', 'ING 002');
insert into cont
values(secventa_cont.nextval,'C002', 'marin2',to_date('February 18, 2020', 'Month dd, yyyy'),
      'parolamarin', 'activ', 'ING 002');
insert into cont
values(secventa_cont.nextval,'C003', 'andreipop',to_date('September 15, 2016', 'Month dd,
      yyyy'), 'adp', 'activ', 'BRD 016');
insert into cont(id_cont, cod_client, nume, parola, detalii_card)
values(secventa_cont.nextval,'C004', 'bibi3', 'anton', 'BCR 098');
insert into cont(id_cont, cod_client, nume, parola, detalii_card)
values(secventa_cont.nextval,'C004', 'bibi5', 'anton', 'ING 270');
insert into cont
values(secventa_cont.nextval,'C005', 'delia1',to_date('May 06, 2020', 'Month dd, yyyy'), 'del',
      'activ', 'BT 221');

```

```

insert into cont
values(secventa_cont.nextval,'C005', 'delia2,to_date('April 27, 2021', 'Month dd, yyyy'), 'del',
'inactiv', 'BT 221');
insert into cont
values(secventa_cont.nextval,'C005', 'delia3,to_date('May 12, 2021', 'Month dd, yyyy'), 'del',
'activ', 'BRD 134');
insert into cont
values(secventa_cont.nextval,'C006', 'ionelmili,to_date('May 13, 2021', 'Month dd, yyyy'), 'mili',
'activ', 'BRD 290');
insert into cont(id_cont, cod_client, nume, parola)
values(secventa_cont.nextval,'C007', 'alinaa', 'alina');
insert into cont
values(secventa_cont.nextval,'C008', 'stelian,to_date('June 3, 2020', 'Month dd, yyyy'),
'stelian00', 'activ', 'BT 009');
insert into cont
values(secventa_cont.nextval,'C009', 'miha,to_date('February 6, 2022', 'Month dd, yyyy'),
'dragomirm', 'activ', 'BRD 124');
insert into cont
values(secventa_cont.nextval,'C009', 'miha2,to_date('February 7, 2022', 'Month dd, yyyy'),
'dragomirm', 'inactiv', 'BRD 124');

```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Text, Tools, Window, Help. The left sidebar shows a tree view of database objects. The main workspace has tabs for 'Script Output', 'Query Result 1', 'Query Result 2', and 'Query Result 3'. The 'Query Result 1' tab contains the following SQL query:

```
select * from cont; --13 rez
```

The results pane displays a table with 13 rows of data:

ID_CONT	COD_CLIENT	NUME	DATA_CREARE	PAROLA	STATUS	DETALII_CARD
1	1C001	dariap	14-MAY-22	daria	inactiv	(null)
2	2C002	marini	09-SEP-17	parolamarini	activ	ING 002
3	3C002	marin2	18-FEB-20	parolamarini	activ	ING 002
4	4C003	andreipop	15-SEP-16	ada	activ	BRD 016
5	5C004	bibi3	14-MAY-22	anton	inactiv	BCR 098
6	6C005	marini15	14-MAY-20	anton	inactiv	ING 270
7	7C005	delia1	09-MAY-20	del	activ	BT 221
8	8C005	delia3	12-MAY-21	del	activ	BRD 134
9	9C006	ionelmili	13-MAY-21	mili	activ	BRD 290
10	10C007	alinaa	14-MAY-22	alina	inactiv	(null)
11	11C008	stelian	03-JUN-20	stelian00	activ	BT 009
12	12C009	miha	06-FEB-22	dragomirm	activ	BRD 124
13	13C009	miha2	07-FEB-22	dragomirm	inactiv	BRD 124

The status bar at the bottom shows the date as 5/16/2022, time as 7:48 PM, and system information including weather (25°C, Mostly cloudy).

```

insert into sezon
values(3, 1, 'Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig', null);

```

```
insert into sezony
values(3, 2, 'Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig, George Blagden',
null);
insert into sezony
values(3, 3, 'Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig', 'Ben Robson');
insert into sezony
values(3, 4, 'Travis Fimmel, Katheryn Winnick, Clive Standen', 'Moe Dunford');
insert into sezony
values(4, 1, 'Sean Bean, Marc Addy, Michelle Fairley, Lena Headey, Emilia Clarke', 'David Benioff');
insert into sezony
values(4, 2, 'Peter Dinklage, Lena Headey, Nikolaj Coster-Waldau, Michelle Fairley, Emilia Clarke',
'David Benioff');
insert into sezony
values(4, 3, 'Peter Dinklage, Lena Headey, Nikolaj Coster-Waldau, Harry Loyd, Emilia Clarke',
'David Benioff');
insert into sezony
values(4, 4, 'Peter Dinklage, Lena Headey, Aidan Gillen, Michelle Fairley, Kit Harington', 'David
Benioff');
insert into sezony
values(4, 5, 'Sean Bean, Lena Headey, Nikolaj Coster-Waldau, Michelle Fairley, Emilia Clarke',
'David Benioff');
insert into sezony
values(8, 1, 'David Duchovny, Gillian Anderson, Mitch Pilegg, Robbie Amell', 'Gillian Anderson');
insert into sezony
values(8, 2, 'David Duchovny, Gillian Anderson, Mitch Pilegg, Lauren Ambrose ', 'Gillian
Anderson');
insert into sezony
values(8, 4, 'David Duchovny, Gillian Anderson, Mitch Pilegg, Robbie Amell', 'David Duchovny');
insert into sezony
values(8, 7, 'David Duchovny, Gillian Anderson, Mitch Pilegg, Rhys Darby', 'David Duchovny');
insert into sezony
values(8, 8, 'David Duchovny, Gillian Anderson, Mitch Pilegg, Lauren Ambrose, Rhys Darby',
'Gillian Anderson');
insert into sezony
values(9, 1, 'Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matt LeBlanc, Matthew Perry, David
Schwimmer', ' David Crane');
insert into sezony
values(9, 3, 'Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matt LeBlanc, Matthew Perry, David
Schwimmer', ' David Crane');
insert into sezony
values(10, 1, 'Alex Lawther, Mackenzie Davis, Jessica Brows', 'Barney Reisz');
```

```

insert into sezon
values(10, 2, 'Mackenzie Davis, Alex Lawther, Hannah John-Kamen, Rhys Darby', 'Charlie
Brooker');

insert into sezon
values(10, 3, 'Alex Lawther, Mackenzie Davis, Hannah John-Kamen, Jessica Brown', 'Barney
Reisz');

```

ID_ARTICOL	NM_SEZON	ACTOR	DIRECTOR
1	3	1Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig, George Blagden	(null)
2	3	2Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig, George Blagden	(null)
3	3	3Travis Fimmel, Katheryn Winnick, Clive Standen, Jessalyn Gilsig	Ben Robson
4	3	4Travis Fimmel, Katheryn Winnick, Clive Standen	Moe Dunford
5	4	1Sean Bean, Marc Addy, Michelle Fairley, Lena Headey, Emilia Clarke	David Benioff
6	4	2Peter Dinklage, Lena Headey, Nikolai Coster-Waldau, Michelle Fairley, Emilia Clarke	David Benioff
7	4	3Peter Dinklage, Lena Headey, Nikolai Coster-Waldau, Harry Lvd, Emilia Clarke	David Benioff
8	4	4Peter Dinklage, Lena Headey, Aidan Gillen, Michelle Fairley, Kit Harington	David Benioff
9	4	5Sean Bean, Lena Headey, Nikolai Coster-Waldau, Michelle Fairley, Emilia Clarke	David Benioff
10	8	1David Duchovny, Gillian Anderson, Mitch Pileggi, Robbie Amell	Gillian Anderson
11	8	2David Duchovny, Gillian Anderson, Mitch Pileggi, Lauren Ambrose	Gillian Anderson
12	8	3David Duchovny, Gillian Anderson, Mitch Pileggi, Robbie Amell	David Duchovny
13	8	4David Duchovny, Gillian Anderson, Mitch Pileggi, Rhys Darby	David Duchovny
14	8	5David Duchovny, Gillian Anderson, Mitch Pileggi, Lauren Ambrose, Rhys Darby	Gillian Anderson
15	9	1Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matt LeBlanc, Matthew Perry, David Schwimmer	David Crane
16	9	2Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matt LeBlanc, Matthew Perry, David Schwimmer	David Crane
17	10	1Alex Lawther, Mackenzie Davis, Jessica Brown	Barney Reisz
18	10	2Mackenzie Davis, Alex Lawther, Hannah John-Kamen, Rhys Darby	Charlie Brooker

```

insert into episod
values(3, 1, 1, 'Rites of Passage', to_date('March 3, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 1, 2, 'Wrath of the Northmen', to_date('March 7, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 1, 3, 'Dispossessed', to_date('March 11, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 1, 4, 'Trial', to_date('March 11, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 2, 1, 'Brothers War', to_date('April 1, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 2, 2, 'Invasion', to_date('April 5, 2013', 'Month dd, yyyy'), 0.7);

insert into episod
values(3, 2, 3, 'Treachery', to_date('April 1, 2013', 'Month dd, yyyy'), 0.7);

```

```
values(3, 3, 1, 'Mercenary', to_date('April 10, 2013', 'Month dd, yyyy'), 0.7);
insert into episod
values(3, 3, 2, 'The Wanderer', to_date('April 14, 2013', 'Month dd, yyyy'), 0.7);
insert into episod
values(3, 4, 1, NULL, to_date('May 2, 2013', 'Month dd, yyyy'), 0.7);
insert into episod
values(3, 4, 2, NULL, to_date('May 6, 2013', 'Month dd, yyyy'), 0.7);
insert into episod
values(3, 4, 3, NULL, to_date('May 11, 2013', 'Month dd, yyyy'), 0.7);
insert into episod
values(4, 1, 1, 'Iarna se apropie', to_date('April 17, 2011', 'Month dd, yyyy'), 1.3);
insert into episod
values(4, 1, 2, 'Drumul regelui', to_date('April 24, 2011', 'Month dd, yyyy'), 1.3);
insert into episod
values(4, 1, 3, 'Lordul Snow', to_date('May 1, 2011', 'Month dd, yyyy'), 1.4);
insert into episod
values(4, 1, 4, 'Infirmi, bastarzi si lucruri stricate', to_date('May 8, 2011', 'Month dd, yyyy'), 1.3);
insert into episod
values(4, 2, 1, 'Nordul isi aminteste', to_date('April 1, 2012', 'Month dd, yyyy'), 1.2);
insert into episod
values(4, 2, 2, 'Moartea', to_date('April 8, 2012', 'Month dd, yyyy'), 1.2);
insert into episod
values(4, 2, 3, 'Viclenie', to_date('April 15, 2012', 'Month dd, yyyy'), 1.2);
insert into episod
values(8, 1, 1, 'Providence', to_date('March 10, 2002', 'Month dd, yyyy'), 1);
insert into episod
values(8, 1, 2, 'Audrey Pauley', to_date('March 17, 2002', 'Month dd, yyyy'), 1);
insert into episod
values(8, 4, 1, 'Jump the Shark', to_date('April 21, 2003', 'Month dd, yyyy'), 1);
insert into episod
values(8, 7, 1, 'Sunshine Days', to_date('February 7, 2003', 'Month dd, yyyy'), 1.1);
insert into episod
values(8, 7, 2, 'The Truth', to_date('February 12, 2003', 'Month dd, yyyy'), 1);
insert into episod
values(10, 1, 1, 'The National Anthem', to_date('December 4, 2011', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 1, 2, 'Fifteen Million Merits', to_date('December 12, 2011', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 1, 3, 'The Entire History of You', to_date('December 20, 2011', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 2, 1, 'Be Right Back', to_date('January 13, 2012', 'Month dd, yyyy'), 0.75);
```

```
insert into episod
values(10, 2, 2, 'White Bear', to_date('January 16, 2012', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 3, 1, NULL, to_date('March 1, 2012', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 3, 2, 'Nosedive', to_date('March 4, 2012', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 3, 3, NULL, to_date('March 6, 2012', 'Month dd, yyyy'), 0.75);
insert into episod
values(10, 3, 4, 'Shut Up and Dance', to_date('March 10, 2012', 'Month dd, yyyy'), 0.75);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 1, 1, 'Going out', 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 1, 2, NULL, 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 1, 3, 'Grandfather Arrives', 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 3, 1, 'The One with the Princess', 1.0);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 3, 2, NULL, 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 3, 3, 'The One with the Jam', 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 3, 4, 'The One with the Tunnel', 0.9);
insert into episod(id_articol, nr_sezon, nr_episod, nume, durata)
values(9, 3, 5, 'The One with Frank Jr.', 0.9);
```

```

select * from episod; --40

```

ID_ARTICOL	NR_SEZON	NR_EPISOD	NAME	DATA_LANGA	DURATA
1	3	1	1Rites of Passage	03-MAR-13	0.7
2	3	1	2Wrath of the Northmen	07-MAR-13	0.7
3	3	1	3Dispossessed	11-MAR-13	0.7
4	3	1	4The Final Days	11-MAR-13	0.7
5	3	2	1Brothers War	01-APR-13	0.7
6	3	2	2Invasion	05-APR-13	0.7
7	3	2	3Treachery	01-APR-13	0.7
8	3	3	1Mercenary	10-APR-13	0.7
9	3	3	2The Wanderer	14-APR-13	0.7
10	3	4	1(null)	02-MAY-13	0.7
11	3	4	2(null)	06-MAY-13	0.7
12	3	4	3(null)	11-MAY-13	0.7
13	4	1	1Tarna se apropie	17-APR-11	1.3
14	4	1	2Drumul regelui	24-APR-11	1.3
15	4	1	3Lordul Snow	01-MAY-11	1.4
16	4	1	4Infini, bastarzi si lucruri stricante	01-MAY-11	1.3
17	4	2	1Lordul Isl aminteste	01-APR-12	1.2
18	4	2	2Moartea	08-APR-12	1.2

```

insert into categorie
values(1, 'Thriller', 4,'Categorie pentru tineri si adulti');

insert into categorie(id_categorie, denumire, nivel_popularitate)
values(2, 'Istoric', 3);

insert into categorie
values(3, 'SF', 5, 'Efecte speciale, povesti inedite');

insert into categorie
values(4, 'Romantic', 4, NULL);

insert into categorie
values(5, 'Comedie', 5, 'Pentru toate varstele');

insert into categorie
values(6, 'Realist', 2, NULL);

insert into categorie
values(7, 'Pentru copii', 5, NULL);

insert into categorie
values(8, 'Actiune', 4, NULL);

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of database objects. The central workspace contains a query window with the following SQL statement:

```
select * from categorie; --8 rez
```

The bottom pane displays the results of the query, which is a table named 'categorie' with columns 'ID_CATEGORIE', 'DENUMIRE', and 'NIVEL_POPULARITATE'. The data is as follows:

ID_CATEGORIE	DENUMIRE	NIVEL_POPULARITATE
1	Thriller	4
2	Istoric	3 (null)
3	SF	5
4	Romantic	4 (null)
5	Comedie	5
6	Realist	5 (null)
7	Pentru copii	5 (null)
8	Actiune	4 (null)

```
insert into trailer
values(1, 1, 'Trailer#1 Pe aripile vantului', 0.01, to_date('February 1, 2014', 'Month dd, yyyy'),
'Movie HD');
insert into trailer
values(2, 2, 'Trailer#1 Inception', 0.02, to_date('December 3, 2013', 'Month dd, yyyy'), NULL);
insert into trailer
values(3, 3, 'Trailer Vikingii', 0.01, to_date('February 14, 2021', 'Month dd, yyyy'), 'Netflix');
insert into trailer
values(4, 4, 'Trailer#1 Urzeala tronurilor', 0.01, to_date('July 18, 2016', 'Month dd, yyyy'), 'HBO');
insert into trailer
values(5, 5, 'Trailer#1 Diamantul sangeriu', 0.02, to_date('July 16, 2011', 'Month dd, yyyy'),
'MovieClips');
insert into trailer
values(6, 6, NULL, 0.01, to_date('December 13, 2011', 'Month dd, yyyy'), 'MovieClips');
insert into trailer
values(7, 7, 'Trailer#1 King Richard', 0.02, to_date('October 16, 2021', 'Month dd, yyyy'), NULL);
insert into trailer
values(8, 8, NULL, 0.01, to_date('June 10, 2014', 'Month dd, yyyy'), NULL);
insert into trailer
values(9, 9, 'Trailer original Friends', 0.01, to_date('July 14, 2009', 'Month dd, yyyy'), 'Steven
Carter');
insert into trailer
values(10, 10, 'Trailer#1 Oglinda neagra', 0.01, to_date('November 29, 2017', 'Month dd, yyyy'),
'Netflix');
insert into trailer
```

```
values(11, 11, 'Trailer Madagascar1', 0.06, to_date('March 2, 2019', 'Month dd, yyyy'),
'DreamWorks Animation');
```

ID_TRAILER	ID_MOVIE	title	director	year	resource
1	1	1Trailer#1 Pe aribile vantului	0.0101-FEB-14	Movie HD	
2	2	2Trailer#1 Inception	0.0203-DEC-13	(null)	
3	3	3Trailer Vikingii	0.0114-FEB-21	Netflix	
4	4	4Trailer#1 Urzeala tronurilor	0.0118-JUL-16	HBO	
5	5	5Trailer#1 Diamantul sanderiei	0.0216-JUL-11	MovieClips	
6	6	6 (null)	0.0114-JUL-11	MovieClips	
7	7	7Trailer#1 King Richard	0.0216-OCT-11	(null)	
8	8	8 (null)	0.0110-JUN-14	(null)	
9	9	9Trailer original Friends	0.0114-JUL-09	Steven Carter	
10	10	10Trailer#1 Oglinza neagra	0.0129-NOV-17	Netflix	
11	11	11Trailer Madagascar1	0.0602-MAR-19	DreamWorks Animation	

```
insert into apartine
values(1, 2);
insert into apartine
values(1, 4);
insert into apartine
values(1, 6);
insert into apartine
values(2, 1);
insert into apartine
values(2, 3);
insert into apartine
values(3, 1);
insert into apartine
values(3, 2);
insert into apartine
values(4, 3);
insert into apartine
values(4, 8);
insert into apartine
values(5, 8);
```

```

insert into apartine
values(5, 6);
insert into apartine
values(6, 4);
insert into apartine
values(6, 5);
insert into apartine
values(7, 6);
insert into apartine
values(7, 8);
insert into apartine
values(8, 1);
insert into apartine
values(9, 5);
insert into apartine
values(10, 1);
insert into apartine
values(10, 6);
insert into apartine
values(11, 7);

```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. A toolbar with various icons is visible above the main workspace. The central workspace has tabs for 'Script Output' and 'Query Result'. The 'Query Result' tab displays the output of the executed query:

ID_ARTICOL	ID_CATEGORY
3	1
4	2
5	2
6	3
7	2
8	4
9	4
10	5
11	5
12	6
13	6
14	7
15	7
16	8
17	8
18	10
19	10
20	11

The bottom status bar shows the weather as '25°C Mostly cloudy', system icons, and the date/time as '7:51 PM 5/16/2022'.

```

insert into abonament
values(secventa_abonament.nextval, 'standard', 25.90, NULL)

```

```

insert into abonament
values(secventa_abonament.nextval, 'standard familie', 30.90, 'Acces la categoriile comedie,
actiune, romantic, SF, oferte articole populare');
insert into abonament
values(secventa_abonament.nextval, 'premium tineri', 35.99, 'Acces la toate filmele pana in 16
ani, oferte de vizionari seriale populare');
insert into abonament
values(secventa_abonament.nextval, 'premium adulti', 38.50, 'Acces la toate articolele, oferte
vizionari articole populare');
insert into abonament
values (secventa_abonament.nextval, 'gold', 40.99, 'Acces in premiera la toate articolele,
reduceri de ziua de nastere');

```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there's a tree view of schemas and objects. The main area is a 'Worksheet' tab where the following SQL query is typed:

```
select * from abonament; --5 rez
```

Below the worksheet, the 'Script Output' tab displays the results of the query. The output shows five rows of data:

NR_ABONAMENT	NUME	PRET_LUMAN	FACULTATE
1	standard	25.9	(null)
2	standard familie	30.9	Acces la categoriile comedie, actiune,
3	premium tineri	35.99	Acces la toate filmele pana in 16 ani,
4	premium adulti	38.5	Acces la toate articolele, oferte vizionari articole populare
5	gold	40.99	Acces in premiera la toate articolele, reduceri de ziua de nastere

```

insert into reclama
values('R001', 1, 'Oferta1', 'Filme si seriale 2022', 600000);
insert into reclama
values('R002', 2, 'Oferta familie', 'Vezi acum toata colectia de filme pentru vacanta de vara!', 800000);
insert into reclama
values('R003', 2, 'Oferta familie2', 'Singur Acasa disponibil acum', 200000);
insert into reclama
values('R004', 3, 'Bonus', 'Plateste in avans si primesti bonus filmul Avatar2 in premiera!', 750000);

```

```

insert into reclama
values('R005', 3, NULL, 'Seriale noi pentru adolescenti', 700000);
insert into reclama
values('R006', 5, 'Oferta gold', 'Descopera cele mai noi filme!', 499900);
insert into reclama
values('R007', 4, 'Luna aventurilor', 'Vezi filme sau seriale din categoria Istorice si poti castiga un
cadou', 600000);
insert into reclama
values('R008', 1, NULL, 'Vrei sa iti schimbi abonamentul pentru o experienta mai interesanta?
Verific acum!', 100000);
insert into reclama
values('R009', 4, NULL, 'The Crown sezonul 5 apare anul acesta!!', 990000);
insert into reclama
values('R010', 3, 'Oferta Iunie', 'Pana la 1 iunie, poti castiga o luna de gratuitate
daca te afli printre top 100 conturi cu cele mai multe accesari', 800000);
insert into reclama
values('R011', 4, 'Oferta Iunie', 'Pana la 1 iunie, poti castiga o luna de gratuitate
daca te afli printre top 100 conturi cu cele mai multe accesari', 900000);

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of database objects. The main area contains a SQL editor window with the following query:

```
select * from reclama; -- rez
```

Below the editor, the results are displayed in a table format:

NR RECLAMA	NR_ABONAMENT	DETALII
R001	10oferta1	Filme si seriale 2022
R002	2Oferta familie	Vezi acutiu toata colectia de filme pentru vacanta de vară!
R003	2Oferta familie2	Sigur, Acum disponibil acum
R004	38nume	Plateste in avans si primesti bonus filmul Avatar2 in premiera!
R005	3(null)	Seriale noi pentru adolescenti
R006	5Oferta gold	Descopera cele mai noi filme!
R007	4Luna aventurilor	Vezi filme sau seriale din categoria Istorice si poti castiga un cadou
R008	1(null)	Vrei sa iti schimbi abonamentul pentru o experienta mai interesanta? Verific acum!
R009	4(null)	The Crown sezonul 5 apare anul acesta!!
R010	3Oferta Iunie	Pana la 1 iunie, poti castiga o luna de gratuitate
R011	4Oferta Iunie	Pana la 1 iunie, poti castiga o luna de gratuitate

The results show 11 rows. The last two rows (R010 and R011) have the same details as the previous ones but are listed again, likely due to a bug or a specific requirement.

```

insert into plateste
values(2, 4, 'P1', 38.5, to_date('November 9, 2021', 'Month dd, yyyy'), to_date('December 9,
2021', 'Month dd, yyyy'));
insert into plateste

```

```
values(2, 4, 'P2', 38.5, to_date('May 3, 2022', 'Month dd, yyyy'), to_date('June 3, 2022', 'Month dd, yyyy'));
insert into plateste
values(3, 2, 'P3', 31, to_date('May 9, 2022', 'Month dd, yyyy'), to_date('June 9, 2022', 'Month dd, yyyy'));
insert into plateste
values(4, 3, 'P4', 36, to_date('April 27, 2021', 'Month dd, yyyy'), to_date('May 27, 2021', 'Month dd, yyyy'));
insert into plateste
values(4, 4, 'P5', 38, to_date('May 4, 2022', 'Month dd, yyyy'), to_date('June 4, 2022', 'Month dd, yyyy'));
insert into plateste
values(6, 5, 'P6', 41, to_date('May 14, 2022', 'Month dd, yyyy'), to_date('June 14, 2022', 'Month dd, yyyy'));
insert into plateste
values(7, 1, 'P7', 52, to_date('May 10, 2022', 'Month dd, yyyy'), to_date('July 10, 2022', 'Month dd, yyyy'));
insert into plateste
values(7, 2, 'P8', 93, to_date('March 22, 2022', 'Month dd, yyyy'), to_date('June 22, 2022', 'Month dd, yyyy'));
insert into plateste
values(9, 2, 'P9', 31, to_date('September 7, 2022', 'Month dd, yyyy'), to_date('November 7, 2022', 'Month dd, yyyy'));
insert into plateste
values(9, 2, 'P10', 31, to_date('March 3, 2022', 'Month dd, yyyy'), to_date('April 3, 2022', 'Month dd, yyyy'));
insert into plateste
values(9, 2, 'P11', 62, to_date('April 4, 2022', 'Month dd, yyyy'), to_date('June 4, 2022', 'Month dd, yyyy'));
insert into plateste
values(8, 5, 'P12', 41, to_date('May 15, 2022', 'Month dd, yyyy'), to_date('June 15, 2022', 'Month dd, yyyy'));
insert into plateste
values(11, 1, 'P13', 26, to_date('May 13, 2022', 'Month dd, yyyy'), to_date('June 19, 2022', 'Month dd, yyyy'));
insert into plateste
values(12, 3, 'P14', 36, to_date('May 16, 2022', 'Month dd, yyyy'), to_date('June 16, 2022', 'Month dd, yyyy'));
insert into plateste
values(13, 1, 'P15', 31, to_date('March 8, 2022', 'Month dd, yyyy'), to_date('April 8, 2022', 'Month dd, yyyy'));
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of schemas and objects. A query window titled '152_11.sql' contains the SQL command: 'select * from plateste; --15 rez'. Below the command, the results are displayed in a table with 15 rows. The table has columns: ID_CONT, NR_ABONAMENT, ID_PLATA, SUMA, DATA_PERIODE, and DATA_VIRARE. The data shows various transaction details. The bottom right corner of the screen shows a Windows taskbar with the date '5/16/2022' and time '7:52 PM'.

ID_CONT	NR_ABONAMENT	ID_PLATA	SUMA	DATA PERIODE	DATA_VIRARE
1	4	4P1	38,50	05-MAY-21	09-DEC-21
2	2	4P2	38,00	05-MAY-22	09-JUN-22
3	3	2P3	31,00	05-MAY-22	09-JUN-22
4	4	3P4	36,27	27-APR-21	27-MAY-21
5	4	4P5	38,04	04-MAY-22	04-JUN-22
6	6	5P6	41,14	04-MAY-22	14-JUN-22
7	7	1P7	52,00	22-APR-22	22-JUL-22
8	7	2P8	93,92	05-MAY-22	09-JUN-22
9	9	2P9	32,07	07-SEP-22	07-NOV-22
10	9	2P10	31,03	03-MAR-22	03-APR-22
11	9	2P11	62,04	04-APR-22	04-JUN-22
12	8	5P12	41,15	05-MAY-22	15-JUN-22
13	11	1P13	26,13	05-MAY-22	19-JUN-22
14	12	3P14	36,16	05-MAY-22	16-JUN-22
15	13	1P15	31,08	08-MAR-22	08-APR-22

insert into recenzie

```
values(1, 'R. Urzeala tronurilor', 'Printre cele mai bune seriale din lume', 5);
```

insert into recenzie

```
values(2, 'R. King Richard', NULL, 3);
```

insert into recenzie

```
values(3, 'R. Urzeala tronurilor', 'Prea multe efecte speciale', 2);
```

insert into recenzie

```
values(4, 'R. Friends', 'Razi cu lacrimi', 5);
```

insert into recenzie

```
values(5, 'R. Oglinda Neagra', 'Exagerat si nepotrivit pentru tineri', 1);
```

insert into recenzie

```
values(6, 'R. Inception', 'Ideea interesanta, filmul prea lung', 4);
```

insert into recenzie

```
values(7, 'R. Diamantul Sangeriu', 'Te tine in suspans', 4);
```

insert into recenzie

```
values(8, 'R. Vikingii', 'Frumos realizat, persoanele putin plăcătoare', 3);
```

insert into recenzie

```
values(9, 'R. Urzeala tronurilor', NULL, 5);
```

insert into recenzie

```
values(10, 'R. Mandrie si prejudecata', 'Amuzant, perfect pentru toate varstele', 5);
```

insert into recenzie

```
values(11, 'R. Friends', NULL, 3);
```

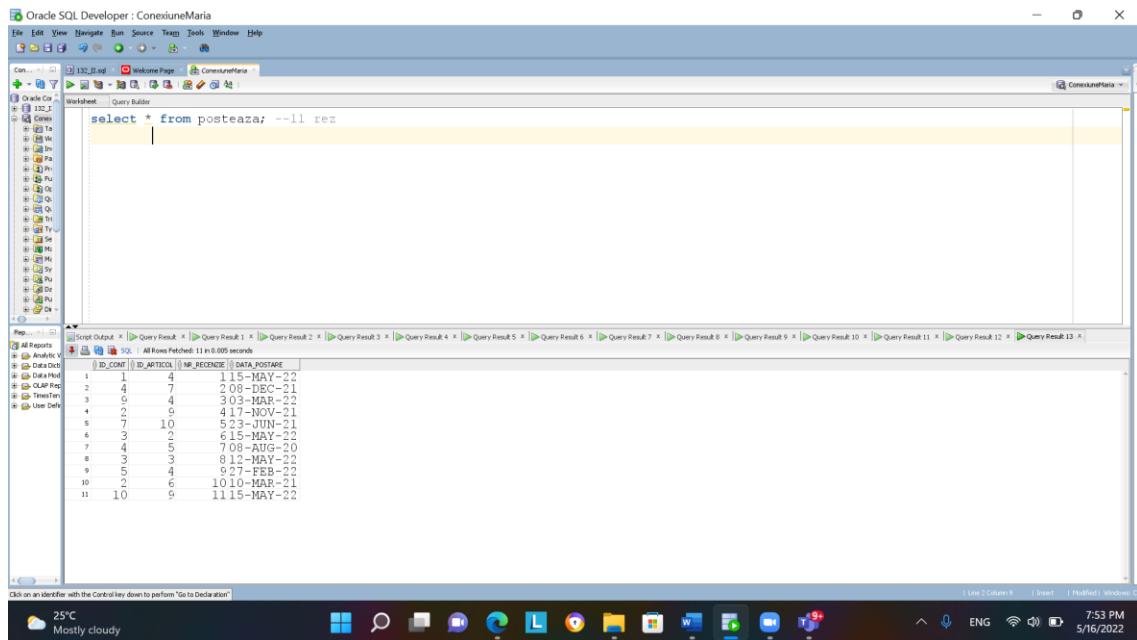
The screenshot shows the Oracle SQL Developer interface. In the central workspace, a query is being run:

```
select * from recenzie; --11 rez
```

The results are displayed in a table format:

NR_RECENZIE	TITLU	COMENTARIU	RATING
1	1.R. Urzeala tronurilor	Printre cele mai bune seriale din lume	5
2	2.R. King Richard	(null)	3
3	3.R. Urzeala tronurilor	Preg multe efecte speciale	5
4	4.R. Friends	Fazi cu lacrimi	3
5	5.R. Oglinzile Neagra	Exagerat si nepotrivit pentru tineri	1
6	6.R. Inception	Idea interesanta, filmul prea lung	4
7	7.R. Diamantul Sanderiei	Te tine in suspans	4
8	8.R. Merlin	Film realizat, persoanele putin plăcute	2
9	9.R. Urzeala tronurilor	(null)	5
10	10.R. Mandrie si prejudecata	Amuzant, perfect pentru toate varstele	5
11	11.R. Friends	(null)	3

```
insert into posteaza(id_cont, id_articol, nr_recenzie)
values(1, 4, 1);
insert into posteaza
values(4, 7, 2, to_date('December 8, 2021', 'Month dd, yyyy'));
insert into posteaza
values(9, 4, 3, to_date('March 3, 2022', 'Month dd, yyyy'));
insert into posteaza
values(2, 9, 4, to_date('November 17, 2021', 'Month dd, yyyy'));
insert into posteaza
values(7, 10, 5, to_date('June 23, 2021', 'Month dd, yyyy'));
insert into posteaza(id_cont, id_articol, nr_recenzie)
values(3, 2, 6);
insert into posteaza
values(4, 5, 7, to_date('August 8, 2020', 'Month dd, yyyy'));
insert into posteaza
values(3, 3, 8, to_date('May 12, 2022', 'Month dd, yyyy'));
insert into posteaza
values(5, 4, 9, to_date('February 27, 2022', 'Month dd, yyyy'));
insert into posteaza
values(2, 6, 10, to_date('March 10, 2021', 'Month dd, yyyy'));
insert into posteaza(id_cont, id_articol, nr_recenzie)
values(10, 9, 11);
```



```

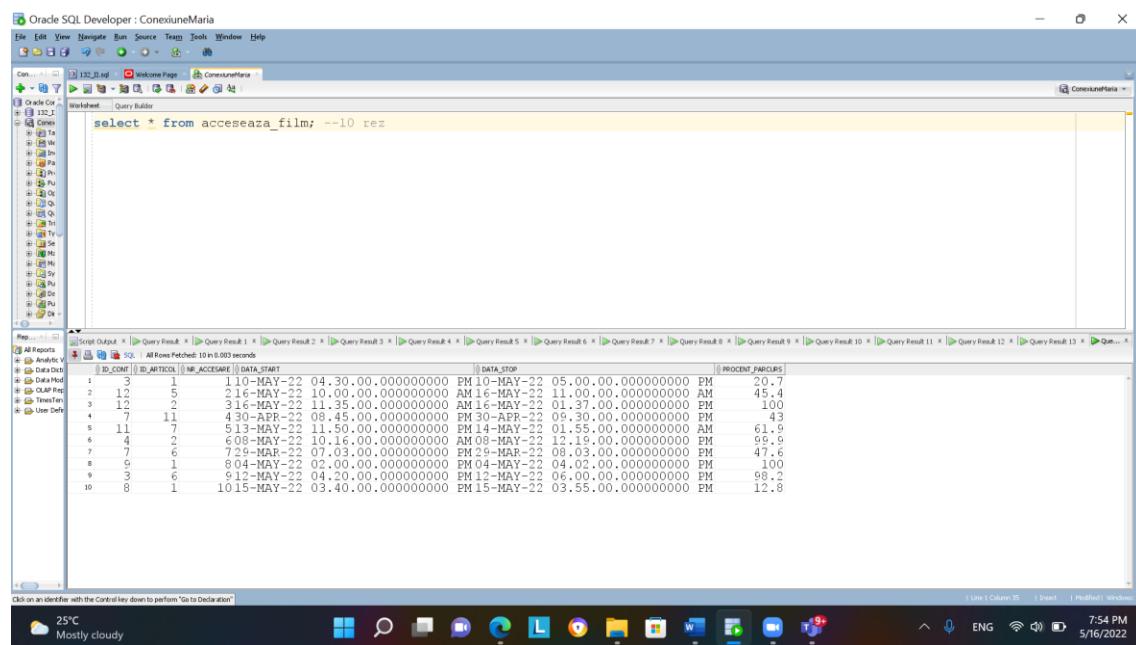
insert into accesearaza_film
values(3, 1, secventa_accesearaza.nextval, to_timestamp('May 10, 2022 16:30', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 10, 2022 17:00', 'Month dd, yyyy hh24:mi'), 20.7);
insert into accesearaza_film
values(12, 5, secventa_accesearaza.nextval, to_timestamp('May 16, 2022 10:00', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 16, 2022 11:00', 'Month dd, yyyy hh24:mi'), 45.4);
insert into accesearaza_film
values(12, 2, secventa_accesearaza.nextval, to_timestamp('May 16, 2022 11:35', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 16, 2022 13:37', 'Month dd, yyyy hh24:mi'), 100);
insert into accesearaza_film
values(7, 11, secventa_accesearaza.nextval, to_timestamp('April 30, 2022 20:45', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('April 30, 2022 21:30', 'Month dd, yyyy hh24:mi'), 43);
insert into accesearaza_film
values(11, 7, secventa_accesearaza.nextval, to_timestamp('May 13, 2022 23:50', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 14, 2022 01:55', 'Month dd, yyyy hh24:mi'), 61.9);
insert into accesearaza_film
values(4, 2, secventa_accesearaza.nextval, to_timestamp('May 8, 2022 10:16', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 8, 2022 10:16', 'Month dd, yyyy hh24:mi'), 61.9);

```

```

    to_timestamp('May 8, 2022 12:19', 'Month dd, yyyy hh24:mi'), 99.9);
insert into acceseaza_film
values(7, 6, seventa_acceseaza.nextval, to_timestamp('March 29, 2022 19:03', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('March 29, 2022 20:03', 'Month dd, yyyy hh24:mi'), 47.6);
insert into acceseaza_film
values(9, 1, seventa_acceseaza.nextval, to_timestamp('May 4, 2022 14:00', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 4, 2022 16:02', 'Month dd, yyyy hh24:mi'), 100);
insert into acceseaza_film
values(3, 6, seventa_acceseaza.nextval, to_timestamp('May 12, 2022 16:20', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 12, 2022 18:00', 'Month dd, yyyy hh24:mi'), 98.2);
insert into acceseaza_film
values(8, 1, seventa_acceseaza.nextval, to_timestamp('May 15, 2022 15:40', 'Month dd, yyyy
hh24:mi'),
      to_timestamp('May 15, 2022 15:55', 'Month dd, yyyy hh24:mi'), 12.8);

```



The screenshot shows the Oracle SQL Developer interface with a query window open. The query is:

```
select * from acceseaza_film; --10 rez
```

The results are displayed in a table:

ID_CONT	ID_ARTICOL	NR_ACCESEAZA	DATA_STAMF	DATA_STOP	PROCENT_PACURS
1	3	1	210-MAY-22 04:30:00,000000000	PM 10-MAY-22 05:00:00,000000000	PM 20,4
2	12	5	216-MAY-22 10:00:00,000000000	AM 16-MAY-22 11:00:00,000000000	AM 45,4
3	12	2	316-MAY-22 11:35:00,000000000	AM 16-MAY-22 01:37:00,000000000	PM 100
4	7	11	4 30-APR-22 08:45:00,000000000	PM 30-APR-22 09:30:00,000000000	PM 43
5	11	7	5 13-MAY-22 09:50:00,000000000	AM 04-MAY-22 01:55:00,000000000	AM 61,6
6	4	2	6 29-MAY-22 10:00:00,000000000	AM 03-MAY-22 11:00:00,000000000	AM 66,6
7	7	6	7 29-MAR-22 07:03:00,000000000	PM 29-MAR-22 08:03:00,000000000	PM 47,6
8	9	1	8 04-MAY-22 02:00:00,000000000	PM 04-MAY-22 04:02:00,000000000	PM 100
9	3	6	9 12-MAY-22 04:20:00,000000000	PM 12-MAY-22 06:00:00,000000000	PM 98,2
10	8	1	10 15-MAY-22 03:40:00,000000000	PM 15-MAY-22 03:55:00,000000000	PM 12,8

```

insert into acceseaza_serial
values(2, 3, 1, 1, seventa_acceseaza.nextval, to_timestamp('May 7, 2022 21:40', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 7, 2022 22:30', 'Month dd, yyyy hh24:mi'), 100, 0.3);
insert into acceseaza_serial

```

```

values(2, 3, 1, 2, secventa_acceseaza.nextval, to_timestamp('May 7, 2022 22:40', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 7, 2022 23:35', 'Month dd, yyyy hh24:mi'), 95.6, 0.59);
insert into accesearza_serial
values(8, 3, 2, 3, secventa_acceseaza.nextval, to_timestamp('May 15, 2022 9:10', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 15, 2022 10:12', 'Month dd, yyyy hh24:mi'), 99, 0.3);
insert into accesearza_serial
values(4, 4, 1, 1, secventa_acceseaza.nextval, to_timestamp('May 6, 2022 16:20', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 6, 2022 16:59', 'Month dd, yyyy hh24:mi'), 80.9, 0.01);
insert into accesearza_serial
values(4, 4, 1, 2, secventa_acceseaza.nextval, to_timestamp('May 8, 2022 17:25', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 8, 2022 18:19', 'Month dd, yyyy hh24:mi'), 90.4, 0.1);
insert into accesearza_serial
values(4, 4, 1, 3, secventa_acceseaza.nextval, to_timestamp('May 9, 2022 17:25', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 9, 2022 17:36', 'Month dd, yyyy hh24:mi'), 20.5, 0.19);
insert into accesearza_serial
values(2, 4, 1, 2, secventa_acceseaza.nextval, to_timestamp('May 8, 2022 19:30', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 8, 2022 20:00', 'Month dd, yyyy hh24:mi'), 55, 0.01);
insert into accesearza_serial
values(7, 8, 7, 1, secventa_acceseaza.nextval, to_timestamp('April 14, 2022 17:40', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 9, 2022 18:30', 'Month dd, yyyy hh24:mi'), 100, 0.2);
insert into accesearza_serial
values(12, 10, 1, 1, secventa_acceseaza.nextval, to_timestamp('May 16, 2022 14:10', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 9, 2022 15:08', 'Month dd, yyyy hh24:mi'), 99, 0.2);
insert into accesearza_serial
values(12, 10, 1, 3, secventa_acceseaza.nextval, to_timestamp('May 16, 2022 15:10', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 9, 2022 16:06', 'Month dd, yyyy hh24:mi'), 98.9, 0.4);
insert into accesearza_serial
values(8, 10, 2, 1, secventa_acceseaza.nextval, to_timestamp('May 15, 2022 22:30', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 15, 2022 23:10', 'Month dd, yyyy hh24:mi'), 79.9, 0.18);
insert into accesearza_serial

```

```

values(11, 9, 3, 5, secventa_acceseaza.nextval, to_timestamp('May 15, 2022 21:30', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 15, 2022 22:25', 'Month dd, yyyy hh24:mi'), 97, 0.01);
insert into accesazea_serial
values(6, 9, 1, 1, secventa_acceseaza.nextval, to_timestamp('May 14, 2022 23:30', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 15, 2022 01:10', 'Month dd, yyyy hh24:mi'), 79.9, 0.18);
insert into accesazea_serial
values(7, 10, 2, 1, secventa_acceseaza.nextval, to_timestamp('May 01, 2022 12:00', 'Month dd,
yyyy hh24:mi'),
      to_timestamp('May 01, 2022 12:40', 'Month dd, yyyy hh24:mi'), 78, 0);

```

The screenshot shows the Oracle SQL Developer interface with a query window open. The query is:

```
select * from accesazea_serial; --12 rez
```

The results are displayed in a table format:

ID_CONT	ID_ARTICOL	NR_SEZON	NR_EPSOZO	NR_ACCESEAZA	DATA_START	DATA_STOP	PROCENT_PARCURS_EPSOZO	PROCENT_PARCURS_SERIAL
1	2	3	1	1	11.07-MAY-22	09.40.00.000000000	FM 07-MAY-22 10.30.00.000000000	PM 100 0.3
2	2	3	1	2	12.07-MAY-22	10.40.00.000000000	FM 07-MAY-22 11.35.00.000000000	PM 95,6 0.59
3	8	3	2	3	13.07-MAY-22	10.50.00.000000000	FM 07-MAY-22 12.40.00.000000000	PM 96 0.3
4	4	4	1	1	14.06-MAY-22	04.20.00.000000000	FM 06-MAY-22 04.55.00.000000000	PM 80,9 0.01
5	4	4	1	2	15.08-MAY-22	05.25.00.000000000	FM 08-MAY-22 06.19.00.000000000	PM 90,4 0.1
6	4	4	1	3	16.09-MAY-22	05.25.00.000000000	FM 09-MAY-22 05.36.00.000000000	PM 20,5 0.19
7	2	4	1	2	17.08-MAY-22	07.30.00.000000000	FM 08-MAY-22 08.00.00.000000000	PM 55 0.01
8	7	8	7	1	18.14-APR-22	05.40.00.000000000	FM 09-MAY-22 06.30.00.000000000	PM 100 0.2
9	8	10	2	1	21.15-MAY-22	10.30.00.000000000	FM 15-MAY-22 11.10.00.000000000	PM 76,9 0.18
10	11	9	3	5	22.15-MAY-22	09.30.00.000000000	FM 15-MAY-22 10.25.00.000000000	PM 87 0.01
11	6	9	1	1	23.14-MAY-22	11.30.00.000000000	FM 15-MAY-22 01.10.00.000000000	PM 79,9 0.18
12	7	10	2	1	24.01-MAY-22	12.00.00.000000000	FM 01-MAY-22 12.40.00.000000000	PM 78 0

12. Cereri SQL

12.1) Să se afișeze numărul de episoade vizionate în mai 2022 de pe fiecare cont, episoade ce aparțin de seriale din categoriile *thriller* și *istoric*. Se va afișa și numele contului.

Se folosesc:

- JOIN pe 4 tabele: CONT, ACCESEAZA_SERIAL, APARTINE și CATEGORIE
- Filtrare la nivel de linii cu clauza WHERE

- Grupare după cont cu GROUP BY
- Funcție pentru siruri de caractere: transformare cu LOWER

```
select c.nume, count(a.nr_episod)
from cont c join accesaza_serial a
on c.id_cont=a.id_cont
join apartine ap
on a.id_articol=ap.id_articol
join categorie ca
on ap.id_categorie=ca.id_categorie
where lower(ca.denumire) in('thriller', 'istoric')
and to_char(a.data_start, 'mm')='05'
and to_char(a.data_start, 'yyyy')='2022'
group by c.nume;
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there's a tree view of database objects. The main area is a 'Worksheet' tab where the SQL query is typed. Below it is a 'Query Result' tab showing the execution output:

NAME	COUNT(A.NR_EPISOD)
delia1	1
delia3	3
marini	4

The status bar at the bottom indicates the system is at 23°C, sunny, and the time is 10:30 AM on 5/17/2022.

12.2) Se vor determina cele mai populare tipuri de abonamente în rândul adulților.

Să se afișeze în ordine alfabetică tipurile de abonamente și valoarea plătită pe ele pe ultimul an pentru abonamentele care îndeplinesc condițiile: plata s-a realizat de către un adult, iar suma plătită este cel puțin egală cu media sumelor plătite de adulți.

Se folosesc :

- Subcerere nesincronizată în care intervin 3 tabele (PLATESTE, CONT și CLIENT), folosită pentru a determina suma medie a plășilor realizate de adulți
- Funcții pe date calendaristice: MONTHS_BETWEEN și TRUNC
- Ordonare alfabetică după tipul de abonament cu ORDER BY

```
select a.tip, sum(p.suma)
from abonament a join plateste p
on a.nr_abonament=p.nr_abonament
join cont c
on p.id_cont=c.id_cont
join client cl
on c.cod_client=cl.cod_client
where months_between(sysdate, data_nastere)>=18*12
and to_char(trunc(p.data_realizare, 'yyyy'))='01-JAN-22'
and p.suma>=(select avg(p1.suma)
               from plateste p1 join cont c1
               on p1.id_cont=c1.id_cont
               join client cl1
               on c1.cod_client=cl1.cod_client
               where months_between(sysdate, cl1.data_nastere)>=18*12)
group by a.tip
order by a.tip;
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query with syntax highlighting. The bottom part shows the results of the query execution. The results table has two columns: 'Tip' and 'SUM(p.suma)'. The data is as follows:

Tip	SUM(p.suma)
standard	52
standard familie	155

12.3) Să se determine sezoanele incomplete (cu cel mult 2 episoade).

Să se afișeze sezonul (și serialul de care aparține), recomandarea în funcție de categoria de vârstă și numărul de episoade pentru sezoanele incomplete.

Se folosesc:

- Funcția DECODE pentru afișarea categoriei de vârstă asociată unor valori
- Filtrare la nivel de grupare cu clauza HAVING
- Funcție (de COUNT) de filtrare la nivel de grupuri
- Funcția CONCAT pentru concatenare pe siruri de caractere: sirul cu numele sezonului și cel cu numărul episodului

```
select concat(concat(a.nume, ' Sezon'), to_char(s.nr_sezon)) as "Sezon",
decode(a.varsta_min_recomandata, 12, 'pentru copii',
       16, 'pentru adolescenti',
       18, 'pentru adulți',
       'nespecificat') as "Recomandare",
count(e.nr_episod) as "Nr episoade"
from articol a join sezon s
on a.id_articol=s.id_articol
left join episod e
on s.nr_sezon=e.nr_sezon
and s.id_articol=e.id_articol
group by a.nume, a.varsta_min_recomandata, s.nr_sezon
having count(e.nr_episod)<=2;
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the SQL query in the 'Query Builder' window. The bottom part shows the 'Query Result' window displaying the output of the query. The output is a table with three columns: Sezon, Recomandare, and Nr episode. The data is as follows:

Sezon	Recomandare	Nr episode
1 Vikingii Sezon 3	pentru adolescenti	2
2 Dosarele X Sezon 1	pentru adolescenti	2
3 Dosarele X Sezon 7	pentru adolescenti	2
4 Orlinda neagră Sezon 2	pentru adulți	2
5 Dosarele X Sezon 4	pentru adolescenti	0
6 Urzelaia tronurilor Sezon 4	pentru copii	0
7 Dosarele X Sezon 2	pentru adolescenti	0
8 Urzelaia tronurilor Sezon 5	pentru copii	0
9 Urzelaia tronurilor Sezon 3	pentru copii	0

12.4) Să se afișeze recenziile pozitive, adică cele cu ratingul mai mare decât media pe acel articol.

Pentru acestea, se afișează numele articolului recenzat, ratingul și comentariul (dacă nu există, se afișează mesajul *Fără comentariu*).

```
select a.nume, r.rating, nvl(r.comentariu, 'Fara comentariu') as "Comentariu"
```

Se folosește:

- Subcerere sincronizată pe 3 tabele (RECENZIE, POSTEAZA și ARTICOL) pentru a determina valoarea medie a ratingului pe **acele** articolele care îndeplinesc condițiile din enunț

```
from articol a
join posteaza p
on a.id_articol=p.id_articol
join recenzie r
on p.nr_recenzie=r.nr_recenzie
where r.rating>=(select avg(r1.rating)
                  from recenzie r1
                  join posteaza p1
                  on r1.nr_recenzie=p1.nr_recenzie
                  where p1.id_articol=a.id_articol);
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Query Editor:** Contains the SQL query for selecting positive reviews based on the average rating of the article.
- Results:** A table titled "rez" displays the results of the query, showing 9 rows of data.
- Table Data:**

NUME	RATING	Comentariu
Inception	4	Ideea interesanta, filmul prea lung
Vikingii	3	Frumos realizat, persoanele putin plăcătisoare
Urzaala tronurilor	5	Printre cele mai bune seriale din lume
Urzaala tronurilor	5	Fara comentariu
Dreamland sanguin	4	tine în suspans
Mandrie si prejudecata	5	Amuzant, perfect pentru toate varstele
Kind Richard	3	Fara comentariu
Friends	5	Razi cu lacrimi
Oglinda neagra	1	Exagerat si nepotrivit pentru tineri

12.5) Să se determine lista filmelor care au cel puțin o vizionare în procent mai mare de 50%.

Pentru acestea, se va afișa pe rând categoria, denumirea, nivelul de popularitate, id-ul și numele trailerului (sau *Trailer*, dacă nu există).

Se folosesc:

- Clauza WITH
- Expresia CASE pentru a afișa nivelul de popularitate al categoriilor în funcție de anumite valori de rating
- Funcție NVL pentru numele nule de trailere

with f as

```
(select a.nume, a.id_articol
from articol a join acceseara_film af
on a.id_articol=af.id_articol
where af.procent_parcurs>=50)
select c.denumire, f.nume, case c.nivel_popularitate when 1 then 'nepopular'
when 2 then 'nepopular'
when 3 then 'mediu'
when 4 then 'popular'
when 5 then 'popular'
end as "Popularitate",
t.id_trailer, nvl(t.nume, 'Trailer') as "Trailer"
from f join apartine ap
on f.id_articol=ap.id_articol
join categorie c
on ap.id_categorie=c.id_categorie
join trailer t
on f.id_articol=t.id_articol
order by c.denumire, f.nume;
```

```

with f as
  (select a.nume, a.id_articol
   from articol a join accesearaza_film af
   on a.id_articol=af.id_articol
   where af.procent_parcurs>=50)
select c.denumire, f.nume, case c.nivel_popularitate when 1 then 'nepopular'
                                              when 2 then 'nepopular'
                                              when 3 then 'mediu'
                                              when 4 then 'popular'
                                              when 5 then 'popular'
                                              end as "Popularitate",
      t.id_trailer, nvl(t.nume, 'Trailer') as "Trailer"
  from f join apartine ap
  on f.id_articol=ap.id_articol
  join categorie c
  on ap.id_categorie=c.id_categorie
  join trailer t
  on f.id_articol=t.id_articol
 order by c.denumire, f.nume,-1 rez

```

The screenshot shows the Oracle SQL Developer interface. The 'Worksheet' tab contains a complex SQL query that joins multiple tables including 'articol', 'accesearaza_film', 'categorie', and 'trailer'. The 'Results' tab displays the output of the query, which consists of four columns: 'denumire' (category name), 'nume' (film name), 'Popularitate' (popularity level), and 'ID_TRAILER' (trailer ID). The popularity levels are categorized as 'nepopular', 'mediu', or 'popular' based on the count of views. The trailer information includes the trailer ID and its name.

denumire	nume	Popularitate	ID_TRAILER
'Actiune'	King Richard	popular	7Trailer#1 King Richard
'Comedie'	Mandrie si prejudecata	popular	6Trailer#1 Pe aripile vantului
'Istoric'	Pe aripile vantului	mediu	1Trailer#1 Pe aripile vantului
'Realist'	King Richard	nepopular	7Trailer#1 King Richard
'Romantic'	Pe aripile vantului	nepopular	1Trailer#1 Pe aripile vantului
'Romantic'	Pe aripile vantului	popular	6Trailer#1 Pe aripile vantului
'Romantic'	Pe aripile vantului	popular	1Trailer#1 Pe aripile vantului

13. Operării

13.1) Să se adauge cadou o lună gratuită de acces la abonament pentru toți clienții născuți în luna octombrie.

update plateste p

```

set p.data_expirare=add_months(p.data_expirare, 1)
where p.id_cont in (select c.id_cont from cont c join client cl
  on c.cod_client=cl.cod_client
  where to_char(cl.data_nastere,'mm')='10');

```

```

update plateste p
set p.data_expirare=add_months(p.data_expirare, 1)
where p.id_cont in (select c.id_cont from cont c join client cl
on c.cod_client=cl.cod_client
where to_char(cl.data_nastere,'mm')='10');--3 res

```

Script Output: Task completed in 0.033 seconds

3 rows updated.

13.2) Să se adauge 500 de lei la câștigul obținut din reclamele ce aparțin de abonamente pe care s-au făcut mai mult de 2 plăți.

```

update reclama
set castig=castig+500
where nr_abonament in (select p.nr_abonament
from plateste p
group by p.nr_abonament
having count(p.id_plata)>2);

```

```

update reclama
set castig=castig+500
where nr_abonament in (select p.nr_abonament
                        from plateste p
                        group by p.nr_abonament
                        having count(p.id_plata)>2);--7 rez

```

7 rows updated.

13. 3) Pentru serialul *Dosarele X*, să se șteargă sezoanele pentru care nu există niciun episod în baza de date.

```

delete from sezon
where id_articol= (select a.id_articol from articol a
                   where lower(a.nume)='dosarele x')
and nr_sezon in (select s.nr_sezon from sezon s
                  join articol a on s.id_articol=a.id_articol
                  left join episod e
                  on s.nr_sezon=e.nr_sezon
                  where e.nr_episod is null and
                  lower(a.nume)='dosarele x');

```

```

delete from sezon
where id_articol= (select a.id_articol from articol a
                     where lower(a.numar)='dosarele x')
and nr_sezon in (select s.nr_sezon from sezon s
                  join articol a on s.id_articol=a.id_articol
                  left join episod e
                  on s.nr_sezon=e.nr_sezon
                  where e.nr_episod is null and
                        lower(a.numar)='dosarele x');--l rez

```

Script Output: Task completed in 0.056 seconds

1 row deleted.

16. Cereri SQL

16.1) Pentru fiecare client, să se afișeze numărul de categorii de filme accesate.

```

select cl.nume||' '||cl.prenume as "Client", count(ca.id_categorie) as "Nr categorii"
from client cl left outer join cont c
on cl.cod_client=c.cod_client
left outer join acceseaza_film af
on c.id_cont=af.id_cont
left outer join apartine a
on af.id_articol=a.id_articol
left outer join categorie ca
on a.id_categorie=ca.id_categorie
group by cl.nume||' '||cl.prenume
order by 1;

```

The screenshot shows the Oracle SQL Developer interface. In the top-left corner, it says "Oracle SQL Developer : ConexiuneMaria". The menu bar includes File, Edit, View, Navigate, Run, Source, Test, Tools, Window, Help. Below the menu is a toolbar with various icons. The main area has a "Worksheet - Query Builder" tab open. The code in the worksheet is:

```

select cl.nume||' '||cl.prenume as "Client", count(ca.id_categorie) as "Nr categorii"
from client cl left outer join cont c
on cl.cod_client=c.cod_client
left outer join accesseaza_film af
on c.id_cont=af.id_cont
left outer join apartine a
on af.id_articol=a.id_articol
left outer join categorie ca
on a.id_categorie=ca.id_categorie
group by cl.nume||' '||cl.prenume
order by 1;--9 rez

```

Below the code is a "Query Result" window titled "All Reports" with a "SQL" icon. It shows the results of the query:

Client	Nr categorii
1 Anton Bianca	0
2 Barbulescu Stelian	2
3 Dragomir Mihaela	4
4 Dulgheru Delia	6
5 Gheorghiu Marin	5
6 Militaru Daniel	3
7 Neacsu Alina	0
8 Popescu Daria	0
9 Popovici Andrei	2

The status bar at the bottom right shows "Line 11 Column 19 | Insert | Modified | Windows | 6:33 PM 5/20/2022".

16.2) Să se afișeze toate articolele care aparțin de categoria *acțiune*.

```

select distinct a.nume
from articol a
where not exists
(select 1 from categorie c
 where lower(c.denumire)='acțiune'
 and not exists(select 1 from apartine ap
 where c.id_categorie=ap.id_categorie and a.id_articol=ap.id_articol));

```

```

select distinct a.nume
from articol a
where not exists
    (select l from categorie c
     where lower(c.denumire)='actiune'
     and not exists(select l from apartine ap
                   where c.id_categorie=ap.id_categorie and a.id_articol=ap.id_articol))--3 rez

```

Query Result
All Rows Patched: 3 in 0.017 seconds
! Diamantul sangeriu
! King Richard
! Urzeala tronurilor

16. 3) Să se afișeze toate id-urile și numele pentru toate conturile care încep cu litera *d* sau *i* și care au adăugat recenzii pentru serialul *Urzeala tronurilor*.

```

select c.id_cont, c.nume
from cont c
where lower(c.nume) like('d%') or lower(c.nume) like('i%')
except
(select c.id_cont, c.nume
from cont c left outer join posteaza p
on c.id_cont=p.id_cont
left outer join articol a
on p.id_articol=a.id_articol
where p.nr_recenzie is null--se elimina si conturile care nu au postat recenzii
or lower(a.nume)!='urzeala tronurilor');

```

Oracle SQL Developer : ConexiuneMaria

File Edit View Navigate Run Source Text Tool Window Help

Con... 132_1.sql welcome Page ConexiuneMaria

Worksheet - Query Builder

```
select c.id_cont, c.numere
from cont c
where lower(c.numere) like('d%') or lower(c.numere) like('i%')
except
(select c.id_cont, c.numere
from cont c left outer join posteaza p
on c.id_cont=p.id_cont
left outer join articol a
on p.id_articol=a.id_articol
where p.nr_recenzie is null--se elimina si conturile care nu au postat recenzie
or lower(a.numere)!='urzeala tronurilor');--2 res|
```

Query Result: A

ID_CONT	NUMERE
1	dariap
2	ionelmili

All Reports Analytic V Data Migr Data Prof OLAP Rep Testben User Defn

Click on an identifier with the Control key down to perform "Go to Declaration".

27°C Mostly sunny

1:03 PM 5/21/2022