

Baza de Date

“MagazinCumparare”

Realizat de Avrinte Teodor-Stefan

Grupa 243

Anul II

Anul universitar curent 2024-2025

Cuprins

1. Prezentati pe scurt baza de date (utilitatea ei).....	5
➤ Descrierea modelului real.....	5
➤ Utilitatea.....	5
➤ Funcționalități.....	5
2. Realizati diagrama entitatea-relatie(ERD): entitatile, relatiile si atributele trebuie definite in limba romana.....	6
3. Pornind de la diagrama entitate-relatie realizati diagrama conceptuala a modelului propus, integrand toate atributele necesare: entitatile, relatiile si atributele trebuie definite in limba romana.....	7
4. Implementati in Oracle diagrama conceptuala realizata: definiți toate tabelele, adaugand toate constrangerile de integritate necesare(chei primare, cheile externe etc).....	8
1.Tabela CLIENT :.....	8
2. Tabela COMANDA :.....	9
3. Tabela PRODUS :.....	10
4. Tabela DETALIICOMANDA :.....	11
5. Tabela JOB :.....	12
6. Tabela CONCEDIU :.....	13
7. Tabela ANGAJAT :.....	14
8. Tabela MAGAZIN :.....	15
9. Tabela PRODUSMAGAZIN :.....	16
10. Tabela TARA :	17
11. Tabela REGIZOR :	18
12. Tabela AUTOR	19
13. Tabela ARTIST	20
14. Tabela ALBUMMUZICAL	21
15. Tabela ALBUMARTIST	22
16. Tabela CARTE	23
17. Tabela CARTEAUTOR	24
18. Tabela FILM	25
19. Tabela FILMREGIZOR	26
5. Adauga informatii coerente in tabelele create (minim 5 inregistrari pentru fiecare entitate independenta; minim 10 inregistrari pentru fiecare tabela asociativa).....	27
1.Tabela CLIENT :.....	27

2. Tabela COMANDA :	28
3. Tabela PRODUS :	30
4. Tabela DETALIICOMANDA :	31
5. Tabela JOB :	33
6. Tabela CONCEDIU :	34
7. Tabela ANGAJAT :	35
8. Tabela MAGAZIN :	36
9. Tabela PRODUSMAGAZIN :	38
10. Tabela TARA :	40
11. Tabela REGIZOR :	42
12. Tabela AUTOR :	44
13. Tabela ARTIST	46
14. Tabela ALBUMMUZICAL	48
15. Tabela ALBUMARTIST	49
16. Tabela CARTE	50
17. Tabela CARTEAUTOR	51
18. Tabela FILM	52
19. Tabela FILMREGIZOR	53
6. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze toate cele 3 tipuri de colectii studiate. Apelati subprogramul	54
7. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celalalt cursor. Apelati subprogramul	58
8. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 dintre tabelele create. Tratati toate exceptiile care pot aparea, incluzand exceptiile predefinite NO_DATA_FOUND si TOO_MANY_ROWS. Apelati subprogramul astfel incat sa evidențiatati toate cazurile tratate	61
9. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa aiba minim 2 parametri si sa utilizeze intr-o singura comanda SQL 5 dintre tabelele create. Definiti minim 2 exceptii proprii, altele decat cele predefinite la nivel de sistem. Apelati subprogramul astfel incat sa evidențiatati toate cazurile definite si tratate	63
10. Definiti un trigger de tip LMD la nivel de comanda. Declansati trigger-ul.	66
11. Definiti un trigger de tip LMD la nivel de linie. Declansati trigger-ul.	68
12. Definiti un trigger de tip LDD. Declansati trigger-ul.	70

Introducere

Tema aleasa pentru proiect:

Tema proiectului acesta este o baza de date a unei librarii care vinde albume,carti si filme.Baza de date are scopul de a ajuta clientii sa cumpere produse din librarie.

Infrastructura utilizata:

Versiunea SGBD-ului: Oracle Database 19c

RAM Alocat: 4GB RAM

Sistem de Operare: Windows 11

Utilizarea sau nu a unei masini virtuale: Nu am folosit masina virtuala

1. Prezentati pe scurt baza de date (utilitatea ei).

Descrierea modelului real

Baza de date este facuta pentru a fi utilizata pentru a ajuta cumparatorul sa afle informatii despre produsul pe care il cumpara, de exemplu de cine este facut si informatii despre omul care a facut produsul pe care doreste sa si-l cumpere.

Entitatea Client stocheaza informatii despre un client cum ar fi numele sau, prenumele, numarul de telefon si alte informatii personale. Entitatea Comanda contine ID-ul clientului, Data in care a fost livrata comanda si date in care aceasta ajunge. Aceasta este legata printr-o tabela asociativa de Produs care are 3 subentitati AlbumMuzical, Carte si Film care reprezinta ce doreste clientul sa cumpere. Ele sunt ajutate de catre atribute ca sa descrie genul continutului si pretul cu care pot fi cumparate, dar si de alte entitati ca sa spuna detalii despre omul care a facut produsele.

Alta entitate care este folosita in baza de date este Magazin care este in relatie cu Produs, iar in fiecare magazin pot sa lucreze unul sau mai multi angajati. Entitatea Angajat contine informatii unice despre un angajat cum ar fi adresa de email, orasul in care locuieste, iar entitatea Job ne spune despre functia pe care o detine fiecare angajat si orele pe saptamana care sunt lucrate de catre acesta.

Utilitatea

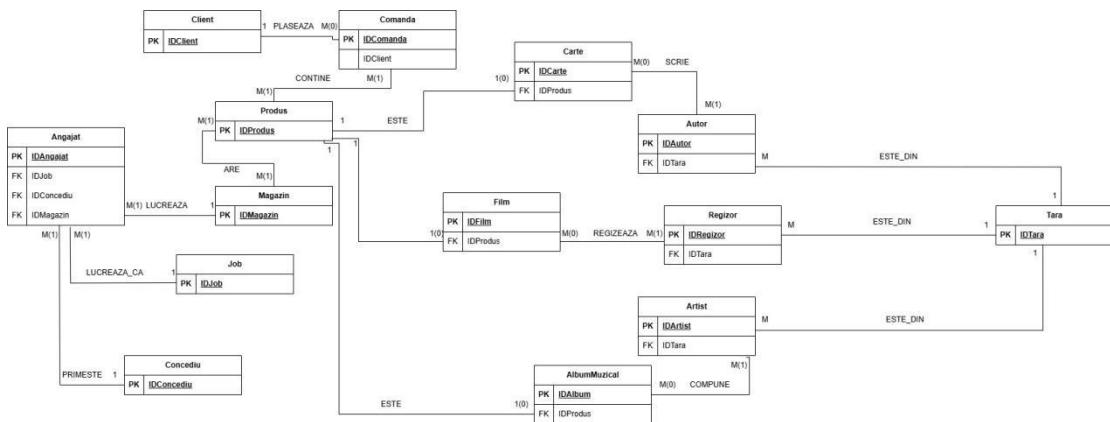
Utilitatea acestei baze de date este sa poata intregistra produsele achizitionate, cantitatea, pretul, data comenzii, metoda de plata. Aceste date sunt utile pentru analiza performantei magazinului.

Baza de date poate sa pastreze informatii despre clienti, cum ar fi numele, adresa, istoricul cumparaturilor. Acest lucru faciliteaza gestionarea relatiilor cu clientii.

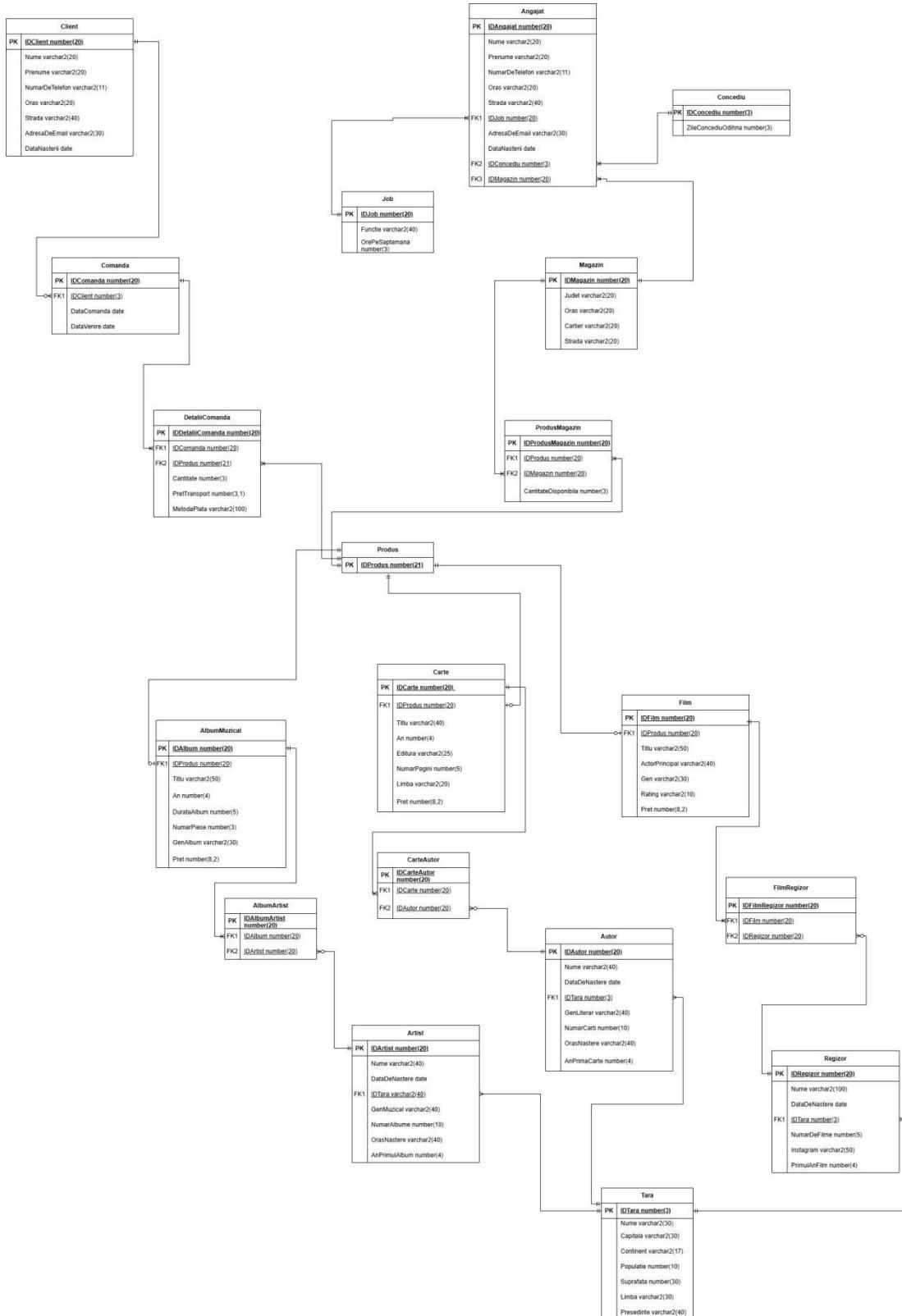
Functionalitati

Aceasta baza de date stocheaza informatii despre produsele disponibile in magazin, inclusiv detalii despre tipul produsului, anul in care a aparut produsul, omul care l-a facut, genul de continut. Acest lucru permite monitorizarea si gestionarea eficienta a inventarului.

2. Realizați diagrama entitate-relație (ERD)



3. Realizați diagrama conceptuală



4.Definiți toate tabelele

1. Tabela Client

```
create table Client  
(  
    IDClient number(20) PRIMARY KEY,  
    Nume varchar2(20),  
    Prenume varchar2(20),  
    NumarDeTelefon varchar2(11),  
    Oras varchar2(20),  
    Strada varchar2(40),  
    AdresaDeEmail varchar2(30),  
    DataNasterii date  
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script to create the 'Client' table. The code is identical to the one provided above. In the bottom-right pane, there is a 'Script Output' window. It displays the results of the table creation: 'Table CLIENT dropped.' followed by 'Table CLIENT created.' A status bar at the bottom of the window indicates 'Task completed in 0.028 seconds'. The interface includes standard database navigation icons (undo, redo, search, etc.) and tabs for 'Script' and 'Output'.

2. Tabela Comanda

```
create table Comanda
(
    IDComanda number(20) PRIMARY KEY,
    IDCClient number(20) REFERENCES Client(IDCClient),
    DataComanda date,
    DataVenire date
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the 'Comanda' table. The script includes columns for 'IDComanda' (primary key), 'IDCClient' (foreign key referencing 'Client'), and two date columns, 'DataComanda' and 'DataVenire'. The bottom-left pane, titled 'Script Output', displays the results of the execution: 'Table CLIENT dropped.', 'Table CLIENT created.', and 'Table COMANDA created.' A progress bar at the bottom indicates the task completed in 0.028 seconds.

```
create table Comanda
(
    IDComanda number(20) PRIMARY KEY,
    IDCClient number(20) REFERENCES Client(IDCClient),
    DataComanda date,
    DataVenire date
);

Script Output X | Task completed in 0.028 seconds
Table CLIENT dropped.
Table CLIENT created.
Table COMANDA created.
```

3. Tabela Produs

```
create table Produs
```

```
(
```

```
    IDProdus number(21) PRIMARY KEY
```

```
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the 'Produs' table. The script includes the table definition with a primary key constraint on the 'IDProdus' column. Below the code editor is a status bar indicating 'Task completed in 0.029 seconds'. The bottom pane is a 'Script Output' window displaying the results of the table creation. It shows four messages: 'Table CLIENT dropped.', 'Table CLIENT created.', 'Table COMANDA created.', and 'Table PRODUS created.'.

```
create table Produs
(
    IDProdus number(21) PRIMARY KEY
);

```

Script Output | Task completed in 0.029 seconds

Table CLIENT dropped.
Table CLIENT created.
Table COMANDA created.
Table PRODUS created.

4. Tabela DetaliiComanda

```
create table DetaliiComanda
(
    IDDetaliiComanda number(20) PRIMARY KEY,
    IDComanda number(20) REFERENCES Comanda(IDComanda),
    IDProdus number(21) REFERENCES Produs(IDProdus),
    Cantitate number(3),
    PretTransport number(3,1),
    MetodaPlata varchar(100),
    CONSTRAINT MetodaPlatacheck CHECK (MetodaPlata IN ('cash', 'card'))
);
```

The screenshot shows the Oracle SQL Developer interface with the 'Script Output' tab selected. The script area contains the SQL code for creating the 'DetaliiComanda' table. The constraint definition 'CONSTRAINT MetodaPlatacheck CHECK (MetodaPlata IN ('cash', 'card'))' is highlighted with a yellow background. The output pane below shows the message 'Table DETALIICOMANDA created.' and a note indicating the task completed in 0.05 seconds.

```
create table DetaliiComanda
(
    IDDetaliiComanda number(20) PRIMARY KEY,
    IDComanda number(20) REFERENCES Comanda(IDComanda),
    IDProdus number(21) REFERENCES Produs(IDProdus),
    Cantitate number(3),
    PretTransport number(3,1),
    MetodaPlata varchar(100),
    CONSTRAINT MetodaPlatacheck CHECK (MetodaPlata IN ('cash', 'card'))
);
```

Script Output | Task completed in 0.05 seconds

Table DETALIICOMANDA created.

5. Tabela Job

```
CREATE TABLE Job (
    IDJob NUMBER(20) PRIMARY KEY,
    Functie VARCHAR2(40),
    OrePeSaptamana NUMBER(3)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the 'Job' table. The script includes columns for 'IDJob' (NUMBER(20) PRIMARY KEY), 'Functie' (VARCHAR2(40)), and 'OrePeSaptamana' (NUMBER(3)). The bottom-right pane shows the 'Script Output' window, which displays the message 'Table JOB created.' and indicates the task was completed in 0.05 seconds.

```
CREATE TABLE Job (
    IDJob NUMBER(20) PRIMARY KEY,
    Functie VARCHAR2(40),
    OrePeSaptamana NUMBER(3)
);
```

Script Output X
Task completed in 0.05 seconds

Table JOB created.

6. Tabela Concediu

```
CREATE TABLE Concediu (
    IDConcediu NUMBER(3) PRIMARY KEY,
    ZileConcediuOdihna NUMBER(3)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor window containing the SQL script to create the 'Concediu' table. The script includes columns for 'IDConcediu' (NUMBER(3) PRIMARY KEY) and 'ZileConcediuOdihna' (NUMBER(3)). A closing parenthesis ')' is also present. The code editor has a light gray background with syntax highlighting. In the bottom-right pane, there is a 'Script Output' window. It displays the message 'Table CONCEDIU created.' and indicates that the task was completed in 0.05 seconds. The output window has a dark header bar with icons for redo, undo, and other database operations.

7. Tabel Magazin

```
create table Magazin
(
    IDMagazin number(20) PRIMARY KEY,
    Judet varchar2(20),
    Oras varchar2(20),
    Cartier varchar2(20),
    Strada varchar2(20)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the Magazin table. The script includes the table definition with columns IDMagazin, Judet, Oras, Cartier, and Strada, all defined with a length of 20. The primary key constraint is applied to the IDMagazin column. The code ends with a closing parenthesis and a semicolon. In the bottom-right pane, there is a 'Script Output' window. The window title is 'Script Output X'. Inside, there is a message bar at the top stating 'Task completed in 0.05 seconds'. Below the message bar, the text 'Table MAGAZIN created.' is displayed. The overall interface is light-colored with standard Windows-style icons.

8. Tabela Angajat

```
create table Angajat
```

```
(
```

```
    IDAngajat number(20) PRIMARY KEY,
```

```
    Nume varchar2(20),
```

```
    Prenume varchar2(20),
```

```
    NumarDeTelefon varchar2(11),
```

```
    Oras varchar2(20),
```

```
    Strada varchar2(40),
```

```
    IDJob number(20) REFERENCES Job(IDJob),
```

```
    AdresaDeEmail varchar2(30),
```

```
    DataNasterii date,
```

```
    IDConcediu number(3) REFERENCES Concediu(IDConcediu),
```

```
    IDMagazin number(20) REFERENCES Magazin(IDMagazin)
```

```
);
```

The screenshot shows the Oracle SQL Developer interface with the 'Script Output' tab selected. The script area contains the SQL code for creating the 'Angajat' table, which includes columns for ID, name, address, job, email, birthdate, and references to other tables. The execution message at the bottom indicates the task was completed in 0.05 seconds.

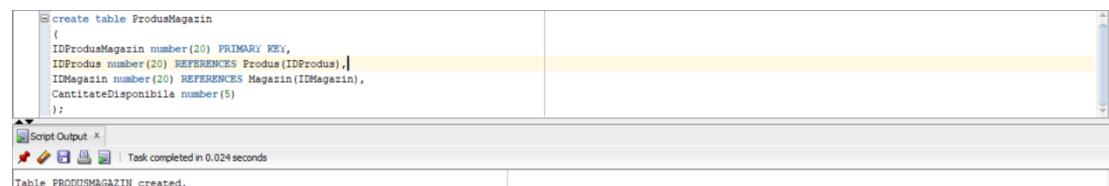
```
create table Angajat
(
    IDAngajat number(20) PRIMARY KEY,
    Nume varchar2(20),
    Prenume varchar2(20),
    NumarDeTelefon varchar2(11),
    Oras varchar2(20),
    Strada varchar2(40),
    IDJob number(20) REFERENCES Job(IDJob),
    AdresaDeEmail varchar2(30),
    DataNasterii date,
    IDConcediu number(3) REFERENCES Concediu(IDConcediu),
    IDMagazin number(20) REFERENCES Magazin(IDMagazin)
);
```

Script Output | Task completed in 0.05 seconds

Table ANGAJAT created.

9. Tabela ProdusMagazin

```
create table ProdusMagazin
(
    IDProdusMagazin number(20) PRIMARY KEY,
    IDProdus number(20) REFERENCES Produs(IDProdus),
    IDMagazin number(20) REFERENCES Magazin(IDMagazin),
    CantitateDisponibila number(5)
);
```



The screenshot shows the Oracle SQL Developer interface with the 'Script Output' tab selected. The script area contains the SQL code for creating the 'ProdusMagazin' table. The output area below shows the command executed and the confirmation 'Table PRODUSMAGAZIN created.'.

```
create table ProdusMagazin
(
    IDProdusMagazin number(20) PRIMARY KEY,
    IDProdus number(20) REFERENCES Produs(IDProdus),
    IDMagazin number(20) REFERENCES Magazin(IDMagazin),
    CantitateDisponibila number(5)
);
Table PRODUSMAGAZIN created.
```

10. Tabela Tara

```
create table Tara
(
    IDTara number(20) PRIMARY KEY,
    Nume varchar2(30),
    Capitala varchar2(30),
    Continent varchar2(17),
    Populatie number(10),
    Suprafata number(15),
    Limba varchar2(30),
    Presedinte varchar2(40)
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there is a small icon of a person. Below the code block, the status bar displays "Script Output" and "Table TARA created." The bottom status bar indicates "Task completed in 0.026 seconds".

```
create table Tara
(
    IDTara number(20) PRIMARY KEY,
    Nume varchar2(30),
    Capitala varchar2(30),
    Continent varchar2(17),
    Populatie number(10),
    Suprafata number(15),
    Limba varchar2(30),
    Presedinte varchar2(40)
);
```

11. Tabela Regizor

```
create table Regizor
(
    IDRegizor number(20) PRIMARY KEY,
    Nume varchar2(20),
    DataDeNastere date,
    IDTara number(20) REFERENCES Tara(IDTara),
    NumarDeFilme number(5),
    Instagram varchar2(50),
    PrimulAnFilm number(4)
);
```



The screenshot shows the Oracle SQL Developer interface with the SQL script for creating the 'Regizor' table. The code is identical to the one above. Below the code, the 'Script Output' tab is visible, showing the message 'Table REGIZOR created.' and a note that the task completed in 0.019 seconds.

```
create table Regizor
(
    IDRegizor number(20) PRIMARY KEY,
    Nume varchar2(20),
    DataDeNastere date,
    IDTara number(20) REFERENCES Tara(IDTara),
    NumarDeFilme number(5),
    Instagram varchar2(50),
    PrimulAnFilm number(4)
);
```

Script Output | Task completed in 0.019 seconds

Table REGIZOR created.

12. Tabela Artist

```
create table Artist
(
    IDArtist number(20) PRIMARY KEY,
    Nume varchar2(20),
    DataDeNastere date,
    IDTara number(20) REFERENCES Tara(IDTara),
    GenMuzical varchar2(40),
    NumarAlbume number(10),
    OrasNastere varchar2(40),
    AnPrimulAlbum number(4)
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the 'Artist' table. The script defines the table structure with columns for IDArtist (number(20), primary key), Nume (varchar2(20)), DataDeNastere (date), IDTara (number(20), foreign key referencing 'Tara'), GenMuzical (varchar2(40)), NumarAlbume (number(10)), OrasNastere (varchar2(40)), and AnPrimulAlbum (number(4)). The code ends with a closing parenthesis and a semicolon. In the bottom-left pane, there is a 'Script Output' window with a toolbar above it. The output window displays the message 'Table ARTIST created.' indicating the successful execution of the command.

```
create table Artist
(
    IDArtist number(20) PRIMARY KEY,
    Nume varchar2(20),
    DataDeNastere date,
    IDTara number(20) REFERENCES Tara(IDTara),
    GenMuzical varchar2(40),
    NumarAlbume number(10),
    OrasNastere varchar2(40),
    AnPrimulAlbum number(4)
);

Table ARTIST created.
```

13. Tabela Autor

```
create table Autor
(
    IDAutor number(7) PRIMARY KEY,
    Nume varchar2(20),
    DataDeNastere date,
    IDTara number(20) REFERENCES Tara(IDTara),
    GenLiterar varchar2(40),
    NumarCarti number(10),
    OrasNastere varchar2(40),
    AnPrimaCarte number(4)
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left corner, there is a small icon of a person with a gear, followed by the text "create table Autor". Below this is a code editor window containing the SQL code for creating the Autor table. The code defines the table structure with columns for IDAutor (number(7), primary key), Nume (varchar2(20)), DataDeNastere (date), IDTara (number(20), references Tara(IDTara)), GenLiterar (varchar2(40)), NumarCarti (number(10)), OrasNastere (varchar2(40)), and AnPrimaCarte (number(4)). The code ends with a closing parenthesis and a semicolon. To the right of the code editor is a vertical toolbar with several icons. At the bottom of the interface, there is a status bar with the text "Script Output" and "Table AUTOR created.", followed by a note indicating the task was completed in 0.028 seconds.

14. Tabel AlbumMuzical

```
create table AlbumMuzical
(
    IDAlbum number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(50),
    An number(4),
    DurataAlbum number(5),
    NumarPiese number(3),
    GenAlbum varchar2(30),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the 'AlbumMuzical' table. The code is identical to the one provided above. In the bottom-left pane, there is a 'Script Output' window with a toolbar. The output message 'Table ALBUMMUZICAL created.' is visible. The bottom status bar indicates the task was completed in 0.032 seconds.

```
create table AlbumMuzical
(
    IDAlbum number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(50),
    An number(4),
    DurataAlbum number(5),
    NumarPiese number(3),
    GenAlbum varchar2(30),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
```

Script Output | Task completed in 0.032 seconds

Table ALBUMMUZICAL created.

15. Tabel Carte

```
create table Carte
(
    IDCarte number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(40),
    An number(4),
    Editura varchar2(25),
    NumarPagini number(5),
    Limba varchar2(20),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the 'Carte' table. The code includes fields for book ID (primary key), product ID (unique), title, year, publisher, page count, language, price, and a foreign key reference to the 'Produs' table. The bottom-left pane is labeled 'Script Output' and shows the message 'Table CARTE created.' indicating the successful execution of the command. The bottom status bar indicates the task completed in 0.03 seconds.

```
create table Carte
(
    IDCarte number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(40),
    An number(4),
    Editura varchar2(25),
    NumarPagini number(5),
    Limba varchar2(20),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
Table CARTE created.
```

16. Tabela Film

```
create table Film
(
    IDFilm number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(50),
    ActorPrincipal varchar2(40),
    Gen varchar2(30),
    IDTara number(3) REFERENCES Tara(IDTara),
    Rating varchar2(10),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
```



The screenshot shows the Oracle SQL Developer interface with the 'Script Output' tab selected. The script area contains the SQL code for creating the 'Film' table, which includes columns for film ID (primary key), producer ID (unique), title, lead actor, genre, country ID (foreign key), rating, price, and a foreign key reference to the 'Produs' table. The execution message at the bottom indicates the task was completed in 0.02 seconds.

```
create table Film
(
    IDFilm number(20) PRIMARY KEY,
    IDProdus number(20) UNIQUE,
    Titlu varchar2(50),
    ActorPrincipal varchar2(40),
    Gen varchar2(30),
    IDTara number(3) REFERENCES Tara(IDTara),
    Rating varchar2(10),
    Pret number(8,2),
    FOREIGN KEY (IDProdus) REFERENCES Produs(IDProdus)
);
```

Script Output | Task completed in 0.02 seconds

Table FILM created.

17. Tabel AlbumArtist

```
create table AlbumArtist  
(IDAlbumArtist number(20) PRIMARY KEY,  
IDAlbum number(20),  
IDArtist number(20)  
);
```



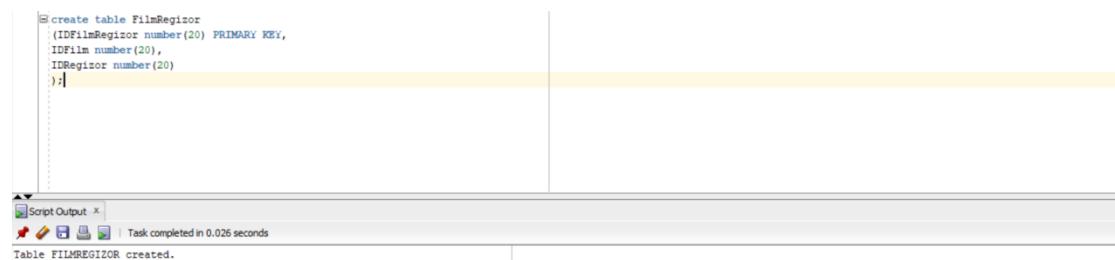
The screenshot shows the MySQL Workbench interface with a script editor window. The script contains the SQL code for creating the 'AlbumArtist' table. Below the script, the status bar indicates 'Task completed in 0.025 seconds' and 'Table ALBUMARTIST created.'

```
create table AlbumArtist  
(IDAlbumArtist number(20) PRIMARY KEY,  
IDAlbum number(20),  
IDArtist number(20)  
);
```

Script Output X | Task completed in 0.025 seconds
Table ALBUMARTIST created.

18. Tabela FilmRegizor

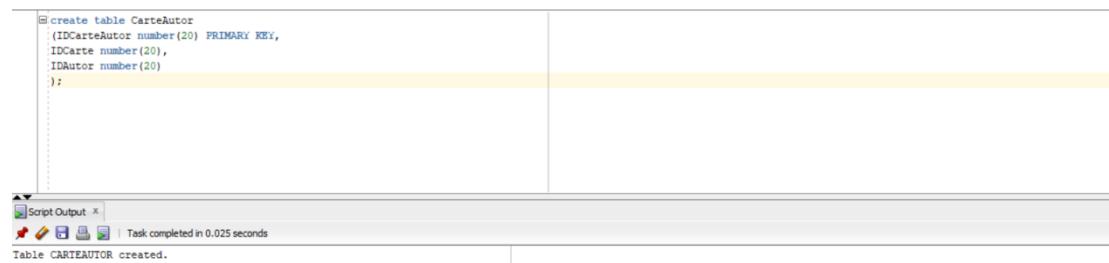
```
create table FilmRegizor  
    (IDFilmRegizor number(20) PRIMARY KEY,  
     IDFilm number(20),  
     IDRegizor number(20)  
);
```



The screenshot shows the Oracle SQL Developer interface with a script editor window. The code in the editor is identical to the one above. Below the editor, a status bar displays "Table FILMREGIZOR created." and "Task completed in 0.026 seconds". The bottom of the screen has a toolbar with various icons.

19. Tabela CarteAutor

```
create table CarteAutor  
  (IDCarteAutor number(20) PRIMARY KEY,  
   IDCarte number(20),  
   IDAutor number(20)  
);
```



The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a code editor containing the SQL script for creating the 'CarteAutor' table. The script includes three columns: 'IDCarteAutor' (number(20), primary key), 'IDCarte' (number(20)), and 'IDAutor' (number(20)). A semi-colon at the end of the script is highlighted with a yellow background. In the bottom-right pane, there is a 'Script Output' window. It displays the command 'Table CARTEAUTOR created.' and a note indicating the task completed in 0.025 seconds. The window has standard icons for closing, minimizing, and maximizing.

```
create table CarteAutor  
  (IDCarteAutor number(20) PRIMARY KEY,  
   IDCarte number(20),  
   IDAutor number(20)  
);
```

Script Output x
| Task completed in 0.025 seconds
Table CARTEAUTOR created.

5.Adăugăți informații coerente în tabelele create

1. Tabela Client

```
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (1, 'Avrinte', 'Teodor', '0743623136', 'Valcea', 'Strada I.L.Caragiale',  
'avrinterica@gmail.com', TO_DATE('2004-09-30', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (4, 'Mutuleasa', 'Andrei', '0734241632', 'Craiova', 'Strada Mihai Eminescu',  
'sorinbeta@gmail.com', TO_DATE('1990-07-08', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (7, 'Ivancu', 'David', '0721353212', 'Braila', 'Strada Ion Creanga',  
'omegamajoc@yahoo.com', TO_DATE('1998-05-20', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (2, 'Botez', 'Marcus', '0725612351', 'Galati', 'Strada Lucian Blaga',  
'ioscarn23@yahoo.com', TO_DATE('1996-04-03', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (6, 'Ciocan', 'Vlad', '0798312312', 'Timis', 'Strada Nichita Stanescu',  
'mamam1251@gmail.com', TO_DATE('2001-01-17', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (3, 'Nistor', 'Georgian', '0712351241', 'Cluj', 'Strada Nicolae Iorga',  
'jupiter3214@gmail.com', TO_DATE('2002-12-12', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)  
  
VALUES (8, 'Anghel', 'Dan', '0736224632', 'Iasi', 'Strada Nicolae Grigorescu',  
'asasic12@yahoo.com', TO_DATE('1999-07-03', 'YYYY-MM-DD'));  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)
```

```
VALUES (5, 'Deaconu', 'Mihnea', '0723152512', 'Constanta', 'Strada George Enescu',
'dadacaece5a@yahoo.com', TO_DATE('1986-09-21', 'YYYY-MM-DD'));
```

```
select * from Client order by IDClient;
```

```
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (1, 'Avrinte', 'Teodor', '0743623136', 'Valcea', 'Strada I.L.Caragiale', 'avrinterica@gmail.com', TO_DATE('2004-09-30', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (2, 'Botez', 'Marcus', '0725612351', 'Galati', 'Strada Lucian Blaga', 'ioscarn23@yahoo.com', TO_DATE('1996-04-03', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (3, 'Nistor', 'Georgian', '0712351241', 'Cluj', 'Strada Niculae Iorga', 'jupiter3214@gmail.com', TO_DATE('2001-01-17', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (4, 'Mutuleasa', 'Andrei', '0734241632', 'Craiova', 'Strada Mihai Eminescu', 'sorinbeta@gmail.com', TO_DATE('1990-07-08', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (5, 'Ivanu', 'David', '0721353212', 'Braila', 'Strada Ion Creanga', 'omegamajoc@yahoo.com', TO_DATE('1998-05-20', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (6, 'Ciocan', 'Vlad', '0798312312', 'Timis', 'Strada Nichita Stanescu', 'mamami251@gmail.com', TO_DATE('2002-12-12', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (7, 'Anghel', 'Dan', '0736224632', 'Iasi', 'Strada Nicolae Grigorescu', 'asasicl2@yahoo.com', TO_DATE('1999-07-03', 'YYYY-MM-DD'));
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (8, 'Deaconu', 'Mihnea', '0723152512', 'Constanta', 'Strada George Enescu', 'dadacaece5a@yahoo.com', TO_DATE('1986-09-21', 'YYYY-MM-DD'));
select * from Client order by IDClient;
```

IDCLIENT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	ADRESADEEMAIL	DATANASTERII
1	Avrinte	Teodor	0743623136	Valcea	Strada I.L.Caragiale	avrinterica@gmail.com	30-SEP-04
2	Botez	Marcus	0725612351	Galati	Strada Lucian Blaga	ioscarn23@yahoo.com	03-APR-96
3	Nistor	Georgian	0712351241	Cluj	Strada Niculae Iorga	jupiter3214@gmail.com	12-DEC-02
4	Mutuleasa	Andrei	0734241632	Craiova	Strada Mihai Eminescu	sorinbeta@gmail.com	08-JUL-90
5	Deaconu	Mihnea	0723152512	Constanta	Strada George Enescu	dadacaece5a@yahoo.com	21-SEP-86
6	Ciocan	Vlad	0798312312	Timis	Strada Nichita Stanescu	mamami251@gmail.com	17-JAN-01
7	Ivanu	David	0721353212	Braila	Strada Ion Creanga	omegamajoc@yahoo.com	20-MAY-98
8	Anghel	Dan	0736224632	Iasi	Strada Nicolae Grigorescu	asasicl2@yahoo.com	03-JUL-99

2. Tabela Comanda

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (1, 1, TO_DATE('2024-05-11', 'YYYY-MM-DD'), TO_DATE('2024-05-15', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (2, 3, TO_DATE('2024-03-12', 'YYYY-MM-DD'), TO_DATE('2024-03-19', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (3, 7, TO_DATE('2023-07-24', 'YYYY-MM-DD'), TO_DATE('2023-07-26', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (4, 1, TO_DATE('2022-09-05', 'YYYY-MM-DD'), TO_DATE('2022-09-09', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (5, 2, TO_DATE('2024-01-05', 'YYYY-MM-DD'), TO_DATE('2024-01-10', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (6, 4, TO_DATE('2020-04-23', 'YYYY-MM-DD'), TO_DATE('2020-04-28', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (7, 6, TO_DATE('2023-12-04', 'YYYY-MM-DD'), TO_DATE('2023-12-09', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (8, 1, TO_DATE('2022-03-29', 'YYYY-MM-DD'), TO_DATE('2022-04-03', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (9, 4, TO_DATE('2024-10-06', 'YYYY-MM-DD'), TO_DATE('2024-10-10', 'YYYY-MM-DD'));
```

```
INSERT INTO Comanda (IDComanda, IDClient, DataComanda, DataVenire)
```

```
VALUES (10, 5, TO_DATE('2024-09-09', 'YYYY-MM-DD'), TO_DATE('2024-09-16', 'YYYY-MM-DD'));
```

```
select * from Comanda order by IDComanda;
```

The screenshot shows a database interface with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the SQL code for inserting 10 rows into the 'Comanda' table. The 'Query Result' tab displays a table with 10 rows of data, each consisting of four columns: IDComanda, IDClient, DataComanda, and DataVenire.

IDCOMANDA	IDCLIENT	DATACOMANDA	DATAVENIRE
1	1	11-MAI-24	15-MAY-24
2	2	3 12-MAR-24	19-MAR-24
3	3	7 24-JUL-23	26-JUL-23
4	4	1 05-SEP-22	09-SEP-22
5	5	2 05-JAN-24	10-JAN-24
6	6	4 23-APR-20	28-APR-20
7	7	6 04-DEC-23	09-DEC-23
8	8	1 29-MAR-22	03-APR-22
9	9	4 06-OCT-24	10-OCT-24
10	10	5 09-SEP-24	16-SEP-24

3. Tabela Produs

```
INSERT INTO Produs(IDProdus) VALUES(1);
INSERT INTO Produs(IDProdus) VALUES(2);
INSERT INTO Produs(IDProdus) VALUES(3);
INSERT INTO Produs(IDProdus) VALUES(4);
INSERT INTO Produs(IDProdus) VALUES(5);
INSERT INTO Produs(IDProdus) VALUES(6);
INSERT INTO Produs(IDProdus) VALUES(7);
INSERT INTO Produs(IDProdus) VALUES(8);
INSERT INTO Produs(IDProdus) VALUES(9);
INSERT INTO Produs(IDProdus) VALUES(10);
INSERT INTO Produs(IDProdus) VALUES(11);
INSERT INTO Produs(IDProdus) VALUES(12);
INSERT INTO Produs(IDProdus) VALUES(13);
INSERT INTO Produs(IDProdus) VALUES(14);
INSERT INTO Produs(IDProdus) VALUES(15);
```

The screenshot shows the Oracle SQL Developer interface. The top part displays the 15 INSERT statements. Below the statements is a results grid titled 'IDPRODUS' with 15 rows, each containing a value from 1 to 15. The bottom status bar indicates 'All Rows Fetched: 15 in 0.007 seconds'.

IDPRODUS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1														
2	2														
3		3													
4		4													
5			5												
6				6											
7					7										
8						8									
9							9								
10								10							
11									11						
12										12					
13											13				
14												14			
15													15		

4. Tabela DetaliiComanda

```
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (1, 1, 1, 2, 7.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (2, 3, 4, 3, 7.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (3, 7, 2, 3, 8.5, 'card');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (4, 4, 10, 2, 9.5, 'card');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (5, 2, 6, 5, 7.5, 'card');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (6, 10, 13, 2, 5.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (7, 8, 6, 2, 5.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (8, 9, 15, 3, 7.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (9, 6, 3, 6, 8.5, 'cash');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)

VALUES (10, 5, 2, 4, 9.5, 'card');

INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus,
Cantitate,PretTransport,MetodaPlata)
```

```
VALUES (11, 3, 14, 2, 7.5, 'card');
```

```
select * from DetaliiComanda order by IDDetaliiComanda;
```

```
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (1, 1, 1, 2, 7.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (2, 3, 4, 3, 7.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (3, 7, 2, 3, 8.5, 'card');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (4, 4, 10, 2, 9.5, 'card');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (5, 2, 6, 5, 7.5, 'card');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (6, 10, 13, 2, 5.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (7, 8, 6, 2, 5.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (8, 9, 15, 3, 7.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (9, 6, 3, 6, 8.5, 'cash');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (10, 5, 2, 4, 9.5, 'card');
INSERT INTO DetaliiComanda (IDDetaliiComanda, IDComanda, IDProdus, Cantitate,PretTransport,MetodaPlata)
VALUES (11, 3, 14, 2, 7.5, 'card');
select * from DetaliiComanda order by IDDetaliiComanda;
```

IDDETALII COMANDA	IDCOMANDA	IDPRODUS	CANTITATE	PRETTRANSPORT	METODAPLATA
1	1	1	1	2	7.5 cash
2	2	3	4	3	7.5 cash
3	3	7	2	3	8.5 card
4	4	4	10	2	9.5 card
5	5	2	6	5	7.5 card
6	6	10	13	2	5.5 cash
7	7	8	6	2	5.5 cash
8	8	9	15	3	7.5 cash
9	9	6	3	6	8.5 cash
10	10	5	2	4	9.5 card
11	11	3	14	2	7.5 card

5. Tabela Job

```
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (1, 'Casier', 40);

INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (2, 'Manager', 45);

INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (3, 'Receptioner', 35);

INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (4, 'Specialist in Achizitii', 42);

INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (5, 'Asistent de vanzari', 38);

INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (6, 'Grafician', 35);
```

select * from Job order by IDJob;

The screenshot shows a database interface with two panes. The left pane contains the SQL script used to insert data into the 'Job' table. The right pane displays the resulting table 'Job' with six rows of data.

```
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (1, 'Casier', 40);
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (2, 'Manager', 45);
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (3, 'Receptioner', 35);
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (4, 'Specialist in Achizitii', 42);
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (5, 'Asistent de vanzari', 38);
INSERT INTO Job (IDJob, Functie, OrePeSaptamana)
VALUES (6, 'Grafician', 35);
select * from Job order by IDJob;
```

IDJOB	FUNCTIE	OREPESAPTAMANA
1	Casier	40
2	Manager	45
3	Receptioner	35
4	Specialist in Achizitii	42
5	Asistent de vanzari	38
6	Grafician	35

6. Tabela Concediu

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(1,25);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(2,28);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(3,33);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(4,24);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(5,26);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(6,29);
```

```
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(1,25);
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(2,28);
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(3,33);
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(4,24);
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(5,26);
INSERT INTO Concediu(IDConcediu,ZileConcediuOdihna) VALUES(6,29);
select * from Concediu order by IDConcediu;
```

Script Output			Query Result	
			SQL	All Rows Fetched: 6 in 0.004 seconds
IDCONCEDIU	ZILECONCEDIUODIHNA			
1	1	25		
2	2	28		
3	3	33		
4	4	24		
5	5	26		
6	6	29		

7. Tabela Magazin

```
INSERT INTO Magazin (IDMagazin, Judet, Oras, Cartier, Strada)
VALUES (1, 'Bucuresti', 'Bucuresti', 'Sector 1', 'Strada Victoriei');

INSERT INTO Magazin (IDMagazin, Judet, Oras, Cartier, Strada)
VALUES (2, 'Cluj', 'Cluj-Napoca', 'Manastur', 'Strada Plopilor');

INSERT INTO Magazin (IDMagazin, Judet, Oras, Cartier, Strada)
VALUES (3, 'Timis', 'Timisoara', 'Iosefin', 'Strada Libertatii');

INSERT INTO Magazin (IDMagazin, Judet, Oras, Cartier, Strada)
VALUES (4, 'Constanta', 'Constanta', 'Centru', 'Strada Decebal');

INSERT INTO Magazin (IDMagazin, Judet, Oras, Cartier, Strada)
VALUES (5, 'Brasov', 'Brasov', 'Schei', 'Strada Republicii');
```

The screenshot shows a database management tool window with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the SQL code for inserting five rows into the 'Magazin' table. The 'Query Result' tab displays the resulting table with columns: IDMAZIN, JUDET, ORAS, CARTIER, and STRADA. The data is as follows:

IDMAZIN	JUDET	ORAS	CARTIER	STRADA
1	1 Bucuresti	Bucuresti	Sector 1	Strada Victoriei
2	2 Cluj	Cluj-Napoca	Manastur	Strada Plopilor
3	3 Timis	Timisoara	Iosefin	Strada Libertatii
4	4 Constanta	Constanta	Centru	Strada Decebal
5	5 Brasov	Brasov	Schei	Strada Republicii

8. Tabela Angajat

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (1, 'Popescu', 'Ion', '0712345678', 'Bucuresti', 'Str Libertatii', 1,  
'popescu.ion@gmail.com', TO_DATE('1990-05-15', 'YYYY-MM-DD'), 3, 1);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (2, 'Ionescu', 'Maria', '0723456789', 'Cluj-Napoca', 'Str Unirii', 2,  
'im198525@gmail.com', TO_DATE('1985-08-25', 'YYYY-MM-DD'), 6, 5);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (3, 'Nedelcu', 'Andrei', '0734567890', 'Timisoara', 'Str Victoriei', 4,  
'hahaha313@yahoo.com', TO_DATE('1993-11-10', 'YYYY-MM-DD'), 4, 2);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (4, 'Pana', 'Elena', '0745678901', 'iasi', 'Str Stefan cel Mare', 6,  
'wow234wow@gmail.com', TO_DATE('1988-04-20', 'YYYY-MM-DD'), 1, 4);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (5, 'Cotoarba', 'Cristian', '0756789012', 'Constanta', 'Str Mihai Viteazu', 5,  
'coticristi432@gmail.com', TO_DATE('1995-09-05', 'YYYY-MM-DD'), 2, 3);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (6, 'Georgescu', 'Ana', '0767890123', 'Brasov', 'Str Brasovului', 3,  
'georgeana221@yahoo.com', TO_DATE('1991-02-15', 'YYYY-MM-DD'), 3, 2);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (7, 'Stan', 'Mihai', '0778901234', 'Ploiesti', 'Str Mihai Eminescu', 2,  
'cemaamec00@yahoo.com', TO_DATE('1987-07-30', 'YYYY-MM-DD'), 1, 3);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

```
VALUES (8, 'Potcovariu', 'Mihaela', '0789012345', 'Sibiu', 'Str Aviatorilor', 1,  
'cevavcads222@gmail.com', TO_DATE('1994-12-10', 'YYYY-MM-DD'), 4, 1);
```

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob,  
AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
```

VALUES (9, 'Stoica', 'Catalin', '0790123456', 'Galati', 'Str Eroilor', 3, 'ma123@gmail.com', TO_DATE('1990-03-25', 'YYYY-MM-DD'), 6, 5);

INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)

VALUES (10, 'Florea', 'Ana-Maria', '0752303443', 'Bacau', 'Str Republicii', 1, 'dsasd111@gmail.com', TO_DATE('1992-06-20', 'YYYY-MM-DD'), 2, 4);

```
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (1, 'Popescu', 'Ion', '0712345678', 'Bucuresti', 'Str Libertatii', 1, 'popescu.ion@gmail.com', TO_DATE('1990-05-15', 'YYYY-MM-DD'), 3, 1);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (2, 'Ionescu', 'Maria', '0723456789', 'Cluj-Napoca', 'Str Unirii', 2, 'im198525@gmail.com', TO_DATE('1985-08-25', 'YYYY-MM-DD'), 6, 5);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (3, 'Nedelcu', 'Andrei', '0734567890', 'Timisoara', 'Str Victoriei', 4, 'hahahaha313@yahoo.com', TO_DATE('1993-11-10', 'YYYY-MM-DD'), 4, 2);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (4, 'Pana', 'Elena', '0745678901', 'Iasi', 'Str Stefan cel Mare', 6, 'www234www@gmail.com', TO_DATE('1988-04-20', 'YYYY-MM-DD'), 1, 4);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (5, 'Cotoarba', 'Cristian', '0756789012', 'Constanta', 'Str Mihai Viteazu', 5, 'coticrist1432@gmail.com', TO_DATE('1995-09-05', 'YYYY-MM-DD'), 2, 3);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (6, 'Georgescu', 'Ana', '0767890123', 'Brasov', 'Str Brasovului', 3, 'georgeana221@yahoo.com', TO_DATE('1991-02-15', 'YYYY-MM-DD'), 3, 2);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (7, 'Stan', 'Mihai', '0778901234', 'Ploiesti', 'Str Mihai Eminescu', 2, 'cemaamec00@yahoo.com', TO_DATE('1987-07-30', 'YYYY-MM-DD'), 1, 3);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (8, 'Potcovariu', 'Mihaela', '0789012345', 'Sibiu', 'Str Aviatorilor', 1, 'cevavcadas222@gmail.com', TO_DATE('1994-12-10', 'YYYY-MM-DD'), 4, 1);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (9, 'Stoica', 'Catalin', '0790123456', 'Galati', 'Str Eroilor', 3, 'mal23@gmail.com', TO_DATE('1990-03-25', 'YYYY-MM-DD'), 6, 5);
INSERT INTO Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataNasterii, IDConcediu, IDMagazin)
VALUES (10, 'Florea', 'Ana-Maria', '0752303443', 'Bacau', 'Str Republicii', 1, 'dsasd111@gmail.com', TO_DATE('1992-06-20', 'YYYY-MM-DD'), 2, 4);
select * from Angajat order by IDAngajat;
```

IDANGAJAT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	IDJOB	ADRESADEEMAIL	DATANASTERII	IDCONCEDIU	IDMAGAZIN
1	Popescu	Ion	0712345678	Bucuresti	Str Libertatii	1	popescu.ion@gmail.com	15-MAY-90	3	1
2	Ionescu	Maria	0723456789	Cluj-Napoca	Str Unirii	2	im198525@gmail.com	25-AUG-85	6	5
3	Nedelcu	Andrei	0734567890	Timisoara	Str Victoriei	4	hahahaha313@yahoo.com	10-NOV-93	4	2
4	Pana	Elena	0745678901	Iasi	Str Stefan cel Mare	6	www234www@gmail.com	20-APR-88	1	4
5	Cotoarba	Cristian	0756789012	Constanta	Str Mihai Viteazu	5	coticrist1432@gmail.com	05-SEP-95	2	3
6	Georgescu	Ana	0767890123	Brasov	Str Brasovului	3	georgeana221@yahoo.com	15-FEB-91	3	2
7	Stan	Mihai	0778901234	Ploiesti	Str Mihai Eminescu	2	cemaamec00@yahoo.com	30-JUL-87	1	3
8	Potcovariu	Mihaela	0789012345	Sibiu	Str Aviatorilor	1	cevavcadas222@gmail.com	10-DEC-94	4	1
9	Stoica	Catalin	0790123456	Galati	Str Eroilor	3	mal23@gmail.com	25-MAR-90	6	5
10	Florea	Ana-Maria	0752303443	Bacau	Str Republicii	1	dsasd111@gmail.com	20-JUN-92	2	4

9. Tabela ProdusMagazin

```
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (1, 1, 1, 10);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (2, 5, 3, 26);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (3, 7, 2, 13);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (4, 3, 5, 45);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (5, 4, 4, 7);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (6, 8, 4, 21);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (7, 15, 4, 45);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (8, 12, 3, 23);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (9, 6, 5, 65);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (10, 13, 1, 12);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (11, 14, 2, 52);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (12, 11, 5, 13);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (13, 10, 4, 34);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (14, 3, 2, 23);

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin,
CantitateDisponibila) VALUES (15, 5, 1, 43);

select * from ProdusMagazin order by IDProdusMagazin;
```

```

INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (1, 1, 1, 10);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (2, 5, 3, 26);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (3, 7, 2, 13);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (4, 3, 5, 45);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (5, 4, 4, 7);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (6, 8, 4, 21);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (7, 15, 4, 45);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (8, 12, 3, 23);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (9, 6, 5, 65);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (10, 13, 1, 12);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (11, 14, 2, 52);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (12, 11, 5, 13);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (13, 10, 4, 34);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (14, 3, 2, 23);
INSERT INTO ProdusMagazin (IDProdusMagazin, IDProdus, IDMagazin, CantitateDisponibila)
VALUES (15, 5, 1, 43);
select * from ProdusMagazin order by IDProdusMagazin;

```

	IDPRODUSMAGAZIN	IDPRODUS	IDMAGAZIN	CANTITATEDISPONIBILA
1	1	1	1	10
2	2	5	3	26
3	3	7	2	13
4	4	3	5	45
5	5	4	4	7
6	6	8	4	21
7	7	15	4	45
8	8	12	3	23
9	9	6	5	65
10	10	13	1	12
11	11	14	2	52
12	12	11	5	13
13	13	10	4	34
14	14	3	2	23
15	15	5	1	43

10. Tabela Tara

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (1, 'Romania', 'Bucuresti', 'Europa', 19500000, 238397, 'Romana', 'Klaus Iohannis');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (2, 'Germania', 'Berlin', 'Europa', 83019200, 357386, 'Germana', 'Frank-Walter Steinmeier');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (3, 'Franta', 'Paris', 'Europa', 66991000, 551695, 'Franceza', 'Emmanuel Macron');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (4, 'SUA', 'Washington, D.C.', 'America de Nord', 331449281, 9833517, 'Engleza', 'Joe Biden');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (5, 'China', 'Beijing', 'Asia', 1444216107, 9596960, 'Chineza', 'Xi Jinping');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (6, 'Japonia', 'Tokyo', 'Asia', 126476461, 377975, 'Japoneza', 'Naruhito');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (7, 'India', 'New Delhi', 'Asia', 1380004385, 3287240, 'Hindi', 'Ram Nath Kovind');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (8, 'Brazilia', 'Brasilia', 'America de Sud', 213993437, 8515767, 'Portugheza', 'Jair Bolsonaro');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (9, 'Rusia', 'Moscova', 'Europa', 146748590, 17098242, 'Rusa', 'Vladimir Putin');
```

```
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte) VALUES (10, 'Australia', 'Canberra', 'Australia', 25687041, 7692024, 'Engleza', 'Scott Morrison');
```

```
select * from Tara order by IDTara;
```

```

INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (1, 'Romania', 'Bucuresti', 'Europa', 19500000, 238397, 'Romana', 'Klaus Iohannis');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (2, 'Germania', 'Berlin', 'Europa', 83019200, 357386, 'Germana', 'Frank-Walter Steinmeier');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (3, 'Franta', 'Paris', 'Europa', 66991000, 551695, 'Franceza', 'Emmanuel Macron');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (4, 'SUA', 'Washington D.C.', 'America de Nord', 331449281, 9833517, 'Engleza', 'Joe Biden');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (5, 'China', 'Beijing', 'Asia', 1444216107, 9596960, 'Chineza', 'Xi Jinping');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (6, 'Japonia', 'Tokyo', 'Asia', 126476461, 377975, 'Japoneza', 'Naruhito');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (7, 'India', 'New Delhi', 'Asia', 1380004385, 3287240, 'Hindi', 'Ram Nath Kovind');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (8, 'Brazilia', 'Brasilia', 'America de Sud', 213993437, 8515767, 'Portugheza', 'Jair Bolsonaro');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (9, 'Rusia', 'Moscova', 'Europa', 146748590, 17098242, 'Rusa', 'Vladimir Putin');
INSERT INTO Tara (IDTara, Nume, Capitala, Continent, Populatie, Suprafata, Limba, Presedinte)
VALUES (10, 'Australia', 'Canberra', 'Australia', 25687041, 7692024, 'Engleza', 'Scott Morrison');

select * from Tara order by IDTara;

```

Script Output x | Query Result x

SQL | All Rows Fetched: 10 in 0.000 seconds

IDTARA	NUME	CAPITALA	CONTINENT	POPULATIE	SUPRAFATA	LIMBA	PRESEDINTE
1	Romania	Bucuresti	Europa	19500000	238397	Romana	Klaus Iohannis
2	Germania	Berlin	Europa	83019200	357386	Germana	Frank-Walter Steinmeier
3	Franta	Paris	Europa	66991000	551695	Franceza	Emmanuel Macron
4	SUA	Washington, D.C.	America de Nord	331449281	9833517	Engleza	Joe Biden
5	China	Beijing	Asia	1444216107	9596960	Chineza	Xi Jinping
6	Japonia	Tokyo	Asia	126476461	377975	Japoneza	Naruhito
7	India	New Delhi	Asia	1380004385	3287240	Hindi	Ram Nath Kovind
8	Brazilia	Brasilia	America de Sud	213993437	8515767	Portugheza	Jair Bolsonaro
9	Rusia	Moscova	Europa	146748590	17098242	Rusa	Vladimir Putin
10	Australia	Canberra	Australia	25687041	7692024	Engleza	Scott Morrison

11. Tabela Regizor

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (1, 'Steven Spielberg', TO_DATE('1946-12-18', 'YYYY-MM-DD'), 4, 34, 'spielbergofficial', 1971);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (2, 'Martin Scorsese', TO_DATE('1942-11-17', 'YYYY-MM-DD'), 4, 26, 'martinscorsese_', 1967);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (3, 'Quentin Tarantino', TO_DATE('1963-03-27', 'YYYY-MM-DD'), 4, 10, 'tarantinoxx', 1992);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (4, 'Spike Lee', TO_DATE('1957-03-20', 'YYYY-MM-DD'), 4, 35, 'officialspikelee', 1986);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (5, 'Hayao Miyazaki', TO_DATE('1941-01-05', 'YYYY-MM-DD'), 6, 11, 'miyazaki_hayao', 1979);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (6, 'Akira Kurosawa', TO_DATE('1910-03-23', 'YYYY-MM-DD'), 6, 30, 'akira_kurosawa', 1943);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (7, 'Cristian Mungiu', TO_DATE('1968-04-27', 'YYYY-MM-DD'), 1, 10, 'cristianmungiu', 2002);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (8, 'Cristi Puiu', TO_DATE('1967-04-03', 'YYYY-MM-DD'), 1, 7, 'cristipuiu', 2001);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (9, 'Jean-Pierre Jeunet', TO_DATE('1953-09-03', 'YYYY-MM-DD'), 3, 9, 'jeunet_jean_pierre', 1991);

INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)

VALUES (10, 'Luc Besson', TO_DATE('1959-03-18', 'YYYY-MM-DD'), 3, 22, 'luchessonofficial', 1983);

select * from Regizor order by IDRegizor;

```
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (1, 'Steven Spielberg', TO_DATE('1946-12-18', 'YYYY-MM-DD'), 4, 34, 'spielbergofficial', 1971);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (2, 'Martin Scorsese', TO_DATE('1942-11-17', 'YYYY-MM-DD'), 4, 26, 'martinscorseese_', 1967);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (3, 'Quentin Tarantino', TO_DATE('1963-03-27', 'YYYY-MM-DD'), 4, 10, 'tarantinnox', 1992);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (4, 'Spike Lee', TO_DATE('1957-03-20', 'YYYY-MM-DD'), 4, 35, 'officialspikelee', 1986);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (5, 'Hayao Miyazaki', TO_DATE('1941-01-05', 'YYYY-MM-DD'), 6, 11, 'miyazaki_hayao', 1979);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (6, 'Akira Kurosawa', TO_DATE('1910-03-23', 'YYYY-MM-DD'), 6, 30, 'akira_kurosawa', 1943);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (7, 'Cristian Mungiu', TO_DATE('1968-04-27', 'YYYY-MM-DD'), 1, 10, 'cristianmungiu', 2002);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (8, 'Cristi Puiu', TO_DATE('1967-04-03', 'YYYY-MM-DD'), 1, 7, 'cristipuiu', 2001);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (9, 'Jean-Pierre Jeunet', TO_DATE('1953-09-03', 'YYYY-MM-DD'), 3, 9, 'jeunet_jean_pierre', 1991);
INSERT INTO Regizor (IDRegizor, Nume, DataDeNastere, IDTara, NumarDeFilme, Instagram, PrimulAnFilm)
VALUES (10, 'Luc Besson', TO_DATE('1959-03-18', 'YYYY-MM-DD'), 3, 22, 'luchessonofficial', 1983);
select * from Regizor order by IDRegizor;
```

Script Output | All Rows Fetched: 10 in 0.005 seconds

IDREGIZOR	NUME	DATADENASTERE	IDTARA	NUMARDEFILME	INSTAGRAM	PRIMULANFILM
1	1 Steven Spielberg	18-DEC-46	4	34	spielbergofficial	1971
2	2 Martin Scorsese	17-NOV-42	4	26	martinscorseese_	1967
3	3 Quentin Tarantino	27-MAR-63	4	10	tarantinnox	1992
4	4 Spike Lee	20-MAR-57	4	35	officialspikelee	1986
5	5 Hayao Miyazaki	05-JAN-41	6	11	miyazaki_hayao	1979
6	6 Akira Kurosawa	23-MAR-10	6	30	akira_kurosawa	1943
7	7 Cristian Mungiu	27-APR-68	1	10	cristianmungiu	2002
8	8 Cristi Puiu	03-APR-67	1	7	cristipuiu	2001
9	9 Jean-Pierre Jeunet	03-SEP-53	3	9	jeunet_jean_pierre	1991
10	10 Luc Besson	18-MAR-59	3	22	luchessonofficial	1983

12. Tabela Artist

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (1, 'Antonio Carlos Jobim', TO_DATE('1927-01-25', 'YYYY-MM-DD'), 8, 'Bossa Nova', 18, 'Rio de Janeiro', 1953);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (2, 'Gilberto Gil', TO_DATE('1942-06-26', 'YYYY-MM-DD'), 8, 'MPB', 40, 'Salvador', 1967);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (3, 'Marisa Monte', TO_DATE('1967-07-01', 'YYYY-MM-DD'), 8, 'MPB', 15, 'Rio de Janeiro', 1989);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (4, 'Kylie Minogue', TO_DATE('1968-05-28', 'YYYY-MM-DD'), 10, 'Pop', 14, 'Melbourne', 1988);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (5, 'Sia', TO_DATE('1975-12-18', 'YYYY-MM-DD'), 10, 'Pop', 9, 'Adelaide', 1997);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (6, 'Keith Urban', TO_DATE('1967-10-26', 'YYYY-MM-DD'), 10, 'Country', 11, 'Whangarei', 1991);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (7, 'Michael Jackson', TO_DATE('1958-08-29', 'YYYY-MM-DD'), 4, 'Pop', 10, 'Gary', 1972);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (8, 'Madonna', TO_DATE('1958-08-16', 'YYYY-MM-DD'), 4, 'Pop', 14, 'Bay City', 1983);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (9, 'Eminem', TO_DATE('1972-10-17', 'YYYY-MM-DD'), 4, 'Rap', 11, 'Saint Joseph', 1996);
```

```
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMuzical, NumarAlbum, OrasNastere, AnPrimulAlbum) VALUES (10, 'Whitney Houston', TO_DATE('1963-08-09', 'YYYY-MM-DD'), 4, 'Pop', 7, 'Newark', 1985);
```

```
select * from Artist order by IDArtist;
```

```

INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (1, 'Antonio Carlos Jobim', TO_DATE('1927-01-25', 'YYYY-MM-DD'), 8, 'Bossa Nova', 18, 'Rio de Janeiro', 1953);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (2, 'Gilberto Gil', TO_DATE('1942-06-26', 'YYYY-MM-DD'), 8, 'MPB', 40, 'Salvador', 1967);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (3, 'Marisa Monte', TO_DATE('1967-07-01', 'YYYY-MM-DD'), 8, 'MPB', 15, 'Rio de Janeiro', 1989);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (4, 'Kylie Minogue', TO_DATE('1968-05-28', 'YYYY-MM-DD'), 10, 'Pop', 14, 'Melbourne', 1988);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (5, 'Sia', TO_DATE('1978-12-18', 'YYYY-MM-DD'), 10, 'Pop', 9, 'Adelaide', 1997);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (6, 'Keith Urban', TO_DATE('1967-10-26', 'YYYY-MM-DD'), 10, 'Country', 11, 'Whangarei', 1991);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (7, 'Michael Jackson', TO_DATE('1958-08-29', 'YYYY-MM-DD'), 4, 'Pop', 10, 'Gary', 1972);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (8, 'Madonna', TO_DATE('1958-08-16', 'YYYY-MM-DD'), 4, 'Pop', 14, 'Bay City', 1983);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (9, 'Eminem', TO_DATE('1972-10-17', 'YYYY-MM-DD'), 4, 'Rap', 11, 'Saint Joseph', 1996);
INSERT INTO Artist (IDArtist, Nume, DataDeNastere, IDTara, GenMusical, NumarAlbum, OrasNastere, AnPrimulAlbum)
VALUES (10, 'Whitney Houston', TO_DATE('1963-08-05', 'YYYY-MM-DD'), 4, 'Pop', 7, 'Newark', 1985);

```

Query Result | Script Output | Query Result | All Rows Fetched: 10 in 0.002 seconds

IDARTIST	NUME	DATADENASTERE	IDTARA	GENMUSICAL	NUMARALBUM	ORASNASTERE	ANPRIMULALBUM
1	Antonio Carlos Jobim	25-JAN-27	8 Bossa Nova	18 Rio de Janeiro	153		
2	Gilberto Gil	26-JUN-42	8 MPB	40 Salvador	1967		
3	Marisa Monte	01-JUL-67	8 MPB	15 Rio de Janeiro	1989		
4	Kylie Minogue	28-MAY-68	10 Pop	14 Melbourne	1988		
5	Sia	18-DEC-75	10 Pop	9 Adelaide	1997		
6	Keith Urban	26-OCT-67	10 Country	11 Whangarei	1991		
7	Michael Jackson	29-AUG-58	4 Pop	10 Gary	1972		
8	Madonna	16-AUG-58	4 Pop	14 Bay City	1983		
9	Eminem	17-OCT-72	4 Rap	11 Saint Joseph	1996		
10	Whitney Houston	05-AUG-85	4 Pop	7 Newark	1985		

13. Tabela Autor

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (1, 'Günter Grass', TO_DATE('1927-10-16', 'YYYY-MM-DD'), 2, 'Fictiune', 15, 'Danzig', 1959);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (2, 'Heinrich Böll', TO_DATE('1917-12-21', 'YYYY-MM-DD'), 2, 'Fictiune', 28, 'Cologne', 1947);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (3, 'Mo Yan', TO_DATE('1955-02-17', 'YYYY-MM-DD'), 5, 'Fictiune', 15, 'Gaomi', 1981);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (4, 'Jin Yong', TO_DATE('1924-03-06', 'YYYY-MM-DD'), 5, 'Fictiune', 15, 'Haining', 1955);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (5, 'Vikram Seth', TO_DATE('1952-06-20', 'YYYY-MM-DD'), 7, 'Fictiune', 6, 'Kolkata', 1986);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (6, 'Arundhati Roy', TO_DATE('1961-11-24', 'YYYY-MM-DD'), 7, 'Fictiune', 2, 'Shillong', 1997);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (7, 'Lev Razgon', TO_DATE('1908-03-10', 'YYYY-MM-DD'), 9, 'Proza', 5, 'Moskva', 1970);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (8, 'Aleksandr Soljenitin', TO_DATE('1918-12-11', 'YYYY-MM-DD'), 9, 'Literatura politica', 13, 'Kislovodsk', 1962);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (9, 'Isaac Asimov', TO_DATE('1920-01-02', 'YYYY-MM-DD'), 9, 'Fictiune stiintifica', 500, 'Petrovichi', 1950);
```

```
INSERT INTO Autor (IDAutor, Nume, DataDeNastere, IDTara, GenLiterar, NumarCarti, OrasNastere, AnPrimaCarte) VALUES (10, 'Vladimir Bukovski', TO_DATE('1942-12-30', 'YYYY-MM-DD'), 9, 'Non-fictiune', 12, 'Belebei', 1963);
```

```
select * from Autor order by IDAutor;
```

```

INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (1, 'Günter Grass', TO_DATE('1927-10-16', 'YYYY-MM-DD'), 2, 'Fictiune', 15, 'Danzig', 1959);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (2, 'Heinrich Böll', TO_DATE('1917-12-21', 'YYYY-MM-DD'), 2, 'Fictiune', 25, 'Cologne', 1947);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (3, 'Mo Yan', TO_DATE('1955-02-17', 'YYYY-MM-DD'), 5, 'Fictiune', 15, 'Gaomi', 1981);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (4, 'Jin Yong', TO_DATE('1924-03-06', 'YYYY-MM-DD'), 5, 'Fictiune', 15, 'Haining', 1955);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (5, 'Vikram Seth', TO_DATE('1950-06-29', 'YYYY-MM-DD'), 7, 'Fictiune', 6, 'Kolkata', 1996);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (6, 'Arundhati Roy', TO_DATE('1961-11-24', 'YYYY-MM-DD'), 7, 'Fictiune', 15, 'Shillong', 1997);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (7, 'Lev Razgon', TO_DATE('1908-03-10', 'YYYY-MM-DD'), 9, 'Proza', 5, 'Moskva', 1970);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (8, 'Aleksandr Soljenitin', TO_DATE('1918-12-11', 'YYYY-MM-DD'), 9, 'Literatura politica', 13, 'Kislovodsk', 1962);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (9, 'Isaac Asimov', TO_DATE('1920-01-02', 'YYYY-MM-DD'), 9, 'Fictiune stiintifica', 500, 'Petrovichi', 1950);
INSERT INTO Autor (IDAutor, Name, DataDeNastere, IDTara, GenLiterat, NumarCarti, OrasNastere, AnPrimaCarte)
VALUES (10, 'Vladimir Bukshevici', TO_DATE('1942-12-30', 'YYYY-MM-DD'), 9, 'Non-fictiune', 12, 'Bielehei', 1963);

```

Script Output | Query Results | All Rows Fetched: 10 in 0.004 seconds

IDAUTOR	NAME	IDDATENASTERE	IDTARA	GENLITERAT	IDNUMARCARTE	IDORASNASTERE	IDANPRIMACARTE
1	Günter Grass	16-OCT-27	2	Fictiune	15	Danzig	1959
2	Heinrich Böll	21-DEC-17	2	Fictiune	25	Cologne	1947
3	Mo Yan	17-TEB-55	5	Fictiune	15	Gaomi	1981
4	Jin Yong	06-MAR-24	5	Fictiune	15	Haining	1955
5	Vikram Seth	20-JUN-52	7	Fictiune	6	Kolkata	1996
6	Arundhati Roy	24-NOV-61	7	Fictiune	2	Shillong	1997
7	Lev Razgon	10-MAR-08	9	Proza	5	Moskva	1970
8	Aleksandr Soljenitin	11-DEC-18	9	Literatura politica	13	Kislovodsk	1962
9	Isaac Asimov	02-JAN-20	9	Fictiune stiintifica	500	Petrovichi	1950
10	Vladimir Bukshevici	30-DEC-42	9	Non-fictiune	12	Bielehei	1963

14. Tabel AlbumMuzical

```
INSERT INTO AlbumMuzical (IDAlbum, IDProdus, Titlu, An, DurataAlbum, NumarPiese,  
GenAlbum, Pret)
```

```
VALUES (1, 1, 'Mais', 1991, 42, 14, 'MPB', 19.99);
```

```
INSERT INTO AlbumMuzical (IDAlbum, IDProdus, Titlu, An, DurataAlbum, NumarPiese,  
GenAlbum, Pret)
```

```
VALUES (2, 2, 'Kylie', 1988, 43, 11, 'Pop', 24.99);
```

```
INSERT INTO AlbumMuzical (IDAlbum, IDProdus, Titlu, An, DurataAlbum, NumarPiese,  
GenAlbum, Pret)
```

```
VALUES (3, 3, 'Got to Be There', 1972, 35, 10, 'Pop', 19.99);
```

```
INSERT INTO AlbumMuzical (IDAlbum, IDProdus, Titlu, An, DurataAlbum, NumarPiese,  
GenAlbum, Pret)
```

```
VALUES (4, 4, 'Madonna', 1983, 41, 8, 'Pop', 21.99);
```

```
INSERT INTO AlbumMuzical (IDAlbum, IDProdus, Titlu, An, DurataAlbum, NumarPiese,  
GenAlbum, Pret)
```

```
VALUES (5, 5, 'Infinite', 1996, 50, 12, 'Rap', 29.99);
```

```
select * from AlbumMuzical order by IDAlbum;
```

The screenshot shows the Oracle SQL Developer interface. On the left, there is a script editor window containing the SQL code for inserting data into the 'AlbumMuzical' table and a select statement to retrieve all rows ordered by IDAlbum. On the right, there is a 'Query Result' window displaying the resulting table with five rows of data.

IDALBUM	IDPRODUS	TITLU	AN	DURATAALBUM	NUMARPIESE	GENALBUM	PRET
1	1	1 Mais	1991	42	14 MPB	19.99	
2	2	2 Kylie	1988	43	11 Pop	24.99	
3	3	3 Got to Be There	1972	35	10 Pop	19.99	
4	4	4 Madonna	1983	41	8 Pop	21.99	
5	5	5 Infinite	1996	50	12 Rap	29.99	

15. Tabel Carte

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
```

```
VALUES (1, 11, 'The Tin Drum', 1959, 'Harcourt', 600, 'Engleza', 35.99);
```

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
```

```
VALUES (2, 12, 'Red Sorghum', 1986, 'Methuen Publishing', 352, 'Engleza', 27.99);
```

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
```

```
VALUES (3, 13, 'True Stories', 2003, 'Random House', 224, 'Engleza', 19.99);
```

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
```

```
VALUES (4, 14, 'Foundation', 1951, 'Gnome Press', 255, 'Engleza', 22.99);
```

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
```

```
VALUES (5, 15, 'To Build a Castle', 1978, 'Viking Press', 416, 'Engleza', 29.99);
```

```
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
VALUES (1, 11, 'The Tin Drum', 1959, 'Harcourt', 600, 'Engleza', 35.99);
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
VALUES (2, 12, 'Red Sorghum', 1986, 'Methuen Publishing', 352, 'Engleza', 27.99);
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
VALUES (3, 13, 'True Stories', 2003, 'Random House', 224, 'Engleza', 19.99);
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
VALUES (4, 14, 'Foundation', 1951, 'Gnome Press', 255, 'Engleza', 22.99);
INSERT INTO Carte (IDCarte, IDProdus, Titlu, An, Editura, NumarPagini, Limba, Pret)
VALUES (5, 15, 'To Build a Castle', 1978, 'Viking Press', 416, 'Engleza', 29.99);
select * from Carte order by IDCarte;
```

Script Output						
Query Result						
SQL						
All Rows Fetched: 5 in 0.003 seconds						
IDCARTE	IDPRODUS	TITLU	AN	EDITURA	NUMARPAGINI	LIMBA
1	1	11The Tin Drum	1959Harcourt		600Engleza	35.99
2	2	12Red Sorghum	1986Methuen Publishing		352Engleza	27.99
3	3	13True Stories	2003Random House		224Engleza	19.99
4	4	14Foundation	1951Gnome Press		255Engleza	22.99
5	5	15To Build a Castle	1978Viking Press		416Engleza	29.99

16. Tabela Film

```
INSERT INTO Film (IDFilm, IDProdus, Titlu, ActorPrincipal, Gen, IDTara, Rating, Pret)  
VALUES (1, 6, 'Schindler''s List', 'Liam Neeson', 'Drama', 4, 'R', 29.99);  
  
INSERT INTO Film (IDFilm, IDProdus, Titlu, ActorPrincipal, Gen, IDTara, Rating, Pret)  
VALUES (2, 7, 'Goodfellas', 'Robert De Niro', 'Crime', 4, 'R', 24.99);  
  
INSERT INTO Film (IDFilm, IDProdus, Titlu, ActorPrincipal, Gen, IDTara, Rating, Pret)  
VALUES (3, 8, 'Pulp Fiction', 'John Travolta', 'Crime', 4, 'R', 27.99);  
  
INSERT INTO Film (IDFilm, IDProdus, Titlu, ActorPrincipal, Gen, IDTara, Rating, Pret)  
VALUES (4, 9, 'Spirited Away', 'Rumi Hiiragi', 'Animation', 6, 'PG', 19.99);  
  
INSERT INTO Film (IDFilm, IDProdus, Titlu, ActorPrincipal, Gen, IDTara, Rating, Pret)  
VALUES (5, 10, 'Moartea Domnului Lazarescu', 'Ion Fiscuteanu', 'Drama', 1, 'R', 22.99);
```

Script Output X Query Result X							
SQL All Rows Fetched: 5 in 0.002 seconds							
IDFILM	IDPRODUS	TITLU	ACTORPRINCIPAL	GEN	IDTARA	RATING	PRET
1	1	6 Schindler's List	Liam Neeson	Drama	4 R	29.99	
2	2	7 Goodfellas	Robert De Niro	Crime	4 R	24.99	
3	3	8 Pulp Fiction	John Travolta	Crime	4 R	27.99	
4	4	9 Spirited Away	Rumi Hiiragi	Animation	6 PG	19.99	
5	5	10 Moartea Domnului Lazarescu	Ion Fiscuteanu	Drama	1 R	22.99	

17. Tabela AlbumArtist

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
```

```
VALUES (1, 1, 3);
```

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
```

```
VALUES (2, 2, 4);
```

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
```

```
VALUES (3, 3, 7);
```

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
```

```
VALUES (4, 4, 8);
```

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
```

```
VALUES (5, 5, 9);
```

```
select * from AlbumArtist order by IDAlbumArtist;
```

The screenshot shows a database interface with two panes. The left pane contains the following SQL code:

```
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
VALUES (1, 1, 3);
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
VALUES (2, 2, 4);
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
VALUES (3, 3, 7);
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
VALUES (4, 4, 8);
INSERT INTO AlbumArtist(IDAlbumArtist, IDAlbum, IDArtist)
VALUES (5, 5, 9);
select * from AlbumArtist order by IDAlbumArtist;
```

The right pane displays the results of the query, showing a table with three columns: IDAlbumArtist, IDAlbum, and IDArtist. The data is as follows:

IDALBUMARTIST	IDALBUM	IDARTIST
1	1	3
2	2	4
3	3	7
4	4	8
5	5	9

18. Tabela FilmRegizor

```
INSERT INTO FilmRegizor(IDFilmRegizor, IDFilm, IDRegizor)
```

```
VALUES (1, 1, 1);
```

```
INSERT INTO FilmRegizor(IDFilmRegizor, IDFilm, IDRegizor)
```

```
VALUES (2, 2, 2);
```

```
INSERT INTO FilmRegizor(IDFilmRegizor, IDFilm, IDRegizor)
```

```
VALUES (3, 3, 3);
```

```
INSERT INTO FilmRegizor(IDFilmRegizor, IDFilm, IDRegizor)
```

```
VALUES (4, 4, 5);
```

```
INSERT INTO FilmRegizor(IDFilmRegizor, IDFilm, IDRegizor)
```

```
VALUES (5, 5, 8);
```

```
select * from FilmRegizor order by IDFilmRegizor;
```

The screenshot shows a database interface with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the SQL code for inserting data into the 'FilmRegizor' table and a 'select' statement. The 'Query Result' tab displays a table with five rows of data, where each row corresponds to one of the inserted entries.

IDFILMREGIZOR	IDFILM	IDREGIZOR
1	1	1
2	2	2
3	3	3
4	4	5
5	5	8

19. Tabela CarteAutor

```
INSERT INTO CarteAutor(IDCarteAutor, IDCarte, IDAutor)
```

```
VALUES (1, 1, 1);
```

```
INSERT INTO CarteAutor(IDCarteAutor, IDCarte, IDAutor)
```

```
VALUES (2, 2, 3);
```

```
INSERT INTO CarteAutor(IDCarteAutor, IDCarte, IDAutor)
```

```
VALUES (3, 3, 7);
```

```
INSERT INTO CarteAutor(IDCarteAutor, IDCarte, IDAutor)
```

```
VALUES (4, 4, 9);
```

```
INSERT INTO CarteAutor(IDCarteAutor, IDCarte, IDAutor)
```

```
VALUES (5, 5, 10);
```

```
select * from CarteAutor order by IDCarteAutor;
```

The screenshot shows a database interface with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the SQL code for inserting data into the 'CarteAutor' table and selecting all rows. The 'Query Result' tab displays a table with five rows of data, where each row corresponds to one of the inserted entries.

IDCARTEAUTOR	IDCARTE	IDAUTOR
1	1	1
2	2	3
3	3	7
4	4	9
5	5	10

6. Subprogram stocat care utilizeaza 3 colectii studiate.

Din tabelul Angajat luam prima jumata de angajati si din ei eliminam a doua jumata fara ultimul angajat, dar eliminam si primul numar sub a doua jumata. Angajatii ramasi isi vor schimba magazinul in care lucreaza, astfel incat, vor lucra in magazinul din Cluj-Napoca. Angajatii care sunt nascuti intre 1 ianuarie 1990 si 1 ianuarie 2000 isi vor schimba zilele de concediu de odihna si vor avea 33 de zile.

```
set serveroutput on;
create procedure AngajatCole
  (id in Angajat.IDConcediu%type,
   id1 in Angajat.IDMagazin%type
  )
is
  type angtab is table of Angajat%ROWTYPE index by pls_integer;
  type angtab1 is table of Angajat%ROWTYPE;
  type angtab2 is varray(10) of Angajat%ROWTYPE;

  angajati angtab;
  angajati1 angtab1 := angtab1();
  angajati2 angtab2 := angtab2();

  t number(4);
  t1 number;
  datanasterii1 date;

begin
  select floor(max(rownum)/2) into t from Angajat;

  for i in 1..t loop
    select * into angajati(i) from Angajat where IDangajat = i;
  end loop;

  t1 := floor(t/2);
```

```

angajati.delete(t1,t-1);

t1:=1;

for i in angajati.first..angajati.last loop
    if angajati.exists(i) then
        DBMS_OUTPUT.PUT_LINE('Nume: ' || angajati(i).nume);
        delete from angajat where IDAngajat=angajati(i).IDAngajat;
        angajati1.extend;
        angajati1(t1) := angajati(i);
        t1 := t1+1;
    end if;
end loop;

DBMS_OUTPUT.NEW_LINE();

for i in 1..angajati1.count loop
    angajati1(i).IDMagazin := id1;
    insert into Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail,
DataNasterii, IDConcediu, IDMagazin)
        values (angajati1(i).IDAngajat, angajati1(i).Nume, angajati1(i).Prenume, angajati1(i).NumarDeTelefon,
angajati1(i).Oras, angajati1(i).Strada, angajati1(i).IDJob,
        angajati1(i).AdresaDeEmail, angajati1(i).DataNasterii, angajati1(i).IDConcediu,
angajati1(i).IDMagazin);
end loop;

select count(*) into t from Angajat;

t1 := 1;
for i in 1..t loop
    select DataNasterii into datanasterii1 from Angajat where IDAngajat = i;
    if datanasterii1 between to_date('01.01.1990', 'DD.MM.YYYY') and to_date('01.01.2000', 'DD.MM.YYYY')
    then
        angajati2.extend;
        select * into angajati2(t1) from Angajat where IDAngajat=i;
        t1 := t1+1;
    end if;
end loop;

```

```

for i in 1..angajati2.count loop
    DBMS_OUTPUT.PUT_LINE('Nume: ' || angajati2(i).nume);
    delete from Angajat where IDAngajat=angajati2(i).IDAngajat;
end loop;

for i in 1..angajati2.count loop
    angajati2(i).IDConcediu := id;
    insert into Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail,
DataNasterii, IDConcediu, IDMagazin)
        values (angajati2(i).IDAngajat, angajati2(i).Nume, angajati2(i).Prenume, angajati2(i).NumarDeTelefon,
angajati2(i).Oras, angajati2(i).Strada, angajati2(i).IDJob,
angajati2(i).AdresaDeEmail, angajati2(i).DataNasterii, angajati2(i).IDConcediu,
angajati2(i).IDMagazin);
end loop;
end;
/
begin
    AngajatCole(id=>3,id1=>2);
end;

set serveroutput on;
create procedure AngajatCole
    (id in Angajat.IDConcediu%type,
     id1 in Angajat.IDMagazin%type
    )
is
    type angtab is table of Angajat%ROWTYPE index by pls_integer;
    type angtab1 is table of Angajat%ROWTYPE;
    type angtab2 is varray(10) of Angajat%ROWTYPE;
    angjati angtab;
    angjatil angtab1 := angtab1();
    angjati2 angtab2 := angtab2();

    t number(4);
    t1 number;
    datanasterii date;
begin
    select floor(max(rownum)/2) into t from Angajat;
    for i in 1..t loop
        select * into angjati(i) from Angajat where IDangajat = i;
    end loop;
    t1 := floor(t/2);
    angjati.delete(t1,t-1);
    t1:=1;
    for i in angjati.first..angjati.last loop
        if angjati.exists(i) then
            DBMS_OUTPUT.PUT_LINE('Nume: ' || angjati(i).nume);
            delete from angajat where IDAngajat=angjati(i).IDAngajat;
        end if;
    end loop;
end;

```

```

        angajatil.extend;
        angajatil(tl) := angajatil(i);
        tl := tl+1;
    end if;
end loop;

DBMS_OUTPUT.NEW_LINE();

for i in 1..angajatil.count loop
    angajatil(i).IDMagazin := idl;
    insert into Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataMasterii, IDConcediu, IDMagazin)
    values (angajatil(i).IDAngajat, angajatil(i).Nume, angajatil(i).Prenume, angajatil(i).NumarDeTelefon, angajatil(i).Oras, angajatil(i).Strada, angajatil(i).IDJob,
            angajatil(i).AdresaDeEmail, angajatil(i).DataMasterii, angajatil(i).IDConcediu, angajatil(i).IDMagazin);
end loop;

select count(*) into t from Angajat;

tl := 1;
for i in 1..t loop
    select DataMasterii into datanasteriil from Angajat where IDAngajat = i;
    if datanasteriil between to_date('01.01.1990', 'DD.MM.YYYY') and to_date('01.01.2000', 'DD.MM.YYYY') then
        angajati2.extend;
        select * into angajati2(tl) from Angajat where IDAngajat=i;
        tl := tl+1;
    end if;
end loop;

for i in 1..angajati2.count loop
    DBMS_OUTPUT.PUT_LINE('Nume: ' || angajati2(i).nume);
    delete from Angajat where IDAngajat=angajati2(i).IDAngajat;
end loop;

for i in 1..angajati2.count loop
    angajati2(i).IDConcediu := id;
    insert into Angajat (IDAngajat, Nume, Prenume, NumarDeTelefon, Oras, Strada, IDJob, AdresaDeEmail, DataMasterii, IDConcediu, IDMagazin)
    values (angajati2(i).IDAngajat, angajati2(i).Nume, angajati2(i).Prenume, angajati2(i).NumarDeTelefon, angajati2(i).Oras, angajati2(i).Strada, angajati2(i).IDJob,
            angajati2(i).AdresaDeEmail, angajati2(i).DataMasterii, angajati2(i).IDConcediu, angajati2(i).IDMagazin);
end loop;
end;
/
begin
    AngajatCole(id>3,idl>2);
end;

```

```

set serveroutput on;
create procedure AngajatCole
    (id in Angajat.IDConcediu%type,
     idl in Angajat.IDMagazin%type
    )
is
    type angtab is table of Angajat%ROWTYPE index by pls_integer;
    type angtab1 is table of Angajat%ROWTYPE;
    type angtab2 is varray(10) of Angajat%ROWTYPE;

    angajati angtab;
    angajatil angtab1 := angtab1();
    angajati2 angtab2 := angtab2();

    t number(4);
    tl number;
    datanasteriil date;

```

Script Output X | Query Result X | Task completed in 0.039 seconds

Nume: Popescu
Nume: Cotoarba

Nume: Popescu
Nume: Nedelcu
Nume: Cotoarba
Nume: Georgescu
Nume: Fotcovariu
Nume: Stoica
Nume: Florea

PL/SQL procedure successfully completed.

7. Subprogram stocat care utilizeaza 2 cursoare diferite.

Din tabelul Tara observam despre fiecare tara daca are mai mult de 40000000 de milioane de locuitori. Apoi pentru fiecare tara verificam artistii din tabelul Artist si scriem despre ei in ce oras din tara respectiva s-au nascut.

```
set serveroutput on;
create procedure PopuOras (pop in Tara.Populatie%type)
is
    cursor c is
        select Nume,Populatie,>IDTara
        from Tara;
    cursor c2(v_c Tara.IDTara%type) is
        select Nume,OrasNastere
        from Artist
        where Artist.IDTara= v_c;
        v_nume Tara.Nume%type;
        v_idtara Tara.IDTara%type;
        v_populatie Tara.Populatie%type;
        v_numeartist Artist.Nume%type;
        v_orasnastere Artist.OrasNastere%type;
begin
    open c;
    loop
        fetch c into v_nume,v_populatie,v_idtara;
        exit when c%notfound;

        if v_populatie>pop then
            DBMS_OUTPUT.PUT_LINE(v_nume || ' are mai mult de ' || pop || ' locuitori avand ' || v_populatie);
        elsif v_idtara<pop then
            DBMS_OUTPUT.PUT_LINE(v_nume || ' are mai putin de ' || pop || ' locuitori avand doar ' || v_populatie);
        end if;
    end loop;
end;
```

```

open c2(v_idtara);

loop

    fetch c2 into v_numearist,v_orasnastere;

    exit when c2%notfound;

    DBMS_OUTPUT.PUT_LINE(v_numearist || ' s-a nascut in tara asta in orasul ' || v_orasnastere);

    end loop;

    DBMS_OUTPUT.NEW_LINE();

    close c2;

end loop;

close c;

end;

/

begin

PopuOras(pop=>40000000);

end;

```

:set serveroutput on;

create procedure PopuOras (pop in Tara.Populatie%type)

is

cursor c is

select Nume,Populatie,IDAra

from Tara;

cursor c2(v_c Tara.IDTara%type) is

select Nume,OrasNastere

from Artist

where Artist.IDTara= v_c;

v_nume Tara.Nume%type;

v_idtara Tara.IDTara%type;

v_populatie Tara.Populatie%type;

v_numearist Artist.Nume%type;

v_orasnastere Artist.OrasNastere%type;

begin

open c;

loop

fetch c into v_nume,v_populatie,v_idtara;

exit when c%notfound;

if v_populatie>pop then

DBMS_OUTPUT.PUT_LINE(v_nume || ' are mai mult de ' || pop || ' locuitori avand ' || v_populatie);

elsif v_idtara>pop then

DBMS_OUTPUT.PUT_LINE(v_nume || ' are mai putin de ' || pop || ' locuitori avand doar ' || v_populatie);

end if;

open c2(v_idtara);

loop

fetch c2 into v_numearist,v_orasnastere;

exit when c2%notfound;

exit when c2%notfound;

DBMS_OUTPUT.PUT_LINE(v_numearist || ' s-a nascut in tara asta in orasul ' || v_orasnastere);

end loop;

DBMS_OUTPUT.NEW_LINE();

close c2;

end loop;

close c;

end;

begin

PopuOras(pop=>40000000);

end;

```

set serveroutput on;
create procedure PopuOras (pop in Tara.Populatie@type)
is
    cursor c is
        select Nume,Populatie,>IDTara
        from Tara;

    cursor c2(v_c Tara.IDTara@type) is
        select Nume,OrasMastere
        from Artist
        where Artist.IDTara= v_c;
    v_nume Tara.Nume@type;
    v_idtara Tara.IDTara@type;
    v_populatie Tara.Populatie@type;

```

Script Output | Task completed in 0.026 seconds

Romania are mai putin de 40000000 locuitori avand doar 19500000

Germania are mai mult de 40000000 locuitori avand 83019200

Franta are mai mult de 40000000 locuitori avand 66991000

SUA are mai mult de 40000000 locuitori avand 331445281

Michael Jackson s-a nascut in tara asta in orasul Gary

Madonna s-a nascut in tara asta in orasul Bay City

Eminem s-a nascut in tara asta in orasul Saint Joseph

Whitney Houston s-a nascut in tara asta in orasul Newark

```

set serveroutput on;
create procedure PopuOras (pop in Tara.Populatie@type)
is
    cursor c is
        select Nume,Populatie,getIDTara
        from Tara;

    cursor c2(v_c Tara.IDTara@type) is
        select Nume,OrasMastere
        from Artist
        where Artist.IDTara= v_c;
    v_nume Tara.Nume@type;
    v_idtara Tara.IDTara@type;
    v_populatie Tara.Populatie@type;

```

Script Output | Task completed in 0.026 seconds

Whitney Houston s-a nascut in tara asta in orasul Newark

China are mai mult de 40000000 locuitori avand 1444216107

Japonia are mai mult de 40000000 locuitori avand 126476461

India are mai mult de 40000000 locuitori avand 1380004385

Brazilia are mai mult de 40000000 locuitori avand 213993437

Antonio Carlos Jobim s-a nascut in tara asta in orasul Rio de Janeiro

Gilberto Gil s-a nascut in tara asta in orasul Salvador

Marisa Monte s-a nascut in tara asta in orasul Rio de Janeiro

Rusia are mai mult de 40000000 locuitori avand 146748590

Australia are mai putin de 40000000 locuitori avand doar 25687041

Kylie Minogue s-a nascut in tara asta in orasul Melbourne

Sia s-a nascut in tara asta in orasul Adelaide

Keith Urban s-a nascut in tara asta in orasul Whangarei

8. Subprogram stocat de tip functie care utilizeaza 3 tabele.

Alegem un numar, iar cu ajutorul acestui numar o sa selectam Produsul cu ID-ul respectiv si o sa afisam orasul magazinului in care se afla produsul. Daca acesta se afla in mai multe magazine sau nu se afla in niciun magazin o sa primim eroare.

```
create function OrasProdus4(p_nr in Number) return Varchar2 is
    v_oras Magazin.Oras%type;
begin
    if p_nr < 0 then
        raise_application_error(-20001, 'ID-ul produsului nu poate fi negativ');
    end if;

    select m.Oras into v_oras
    from Magazin m, Produsmagazin pm, Produs p
    where m.IDMagazin=pm.IDMagazin
    and pm.IDProdus=p.IDProdus
    and p.IDProdus=p_nr;

    return v_oras;
exception
    when no_data_found then
        return 'Produsul nu este in niciun magazin';
    when too_many_rows then
        return 'Produsul se afla in mai multe magazine';
    when others then
        return 'Codul erorii: ' || SQLCODE || ' Mesajul erorii: ' || SQLERRM;
end;
/
set serveroutput on;
declare
    v_rezultat varchar2(100);
    v_nr number := &p_nr;
```

```

begin
  v_rezultat := OrasProdus4(v_nr);
  DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || '' || v_rezultat);
end;
/

```

create function OrasProdus4(p_nr in Number) return Varchar2 is
 v_oras Magazin.OrasType;
begin
 if p_nr < 0 then
 raise_application_error(-20001, 'ID-ul produsului nu poate fi negativ');
 end if;

 select m.Oras into v_oras
 from Magazin m, ProdusMagazin pm, Produs p
 where m.IDMagazin=pm.IDMagazin
 and pm.IDProdus=p.IDProdus
 and p.IDProdus=p_nr;

 return v_oras;
exception
 when no_data_found then
 return 'Produsul nu este in niciun magazin';
 when too_many_rows then
 return 'Produsul se afla in mai multe magazine';
 when others then
 return 'Codul erorii: ' || SQLCODE || ' Mesajul erorii: ' || SQLERRM;
end;
/

set serveroutput on;
declare
 v_rezultat varchar2(100);
 v_nr number := 4p_nr;
begin
 v_rezultat := OrasProdus4(v_nr);
 DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || '' || v_rezultat);
end;
.

Apel Subprogram: v_nr=1

```

set serveroutput on;
declare
  v_rezultat varchar2(100);
  v_nr number := 4p_nr;
begin
  v_rezultat := OrasProdus4(v_nr);
  DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || '' || v_rezultat);
end;
/

```

Script Output | Query Result | Task completed in 1.207 seconds

IDProdus: 1 Bucuresti

PL/SQL procedure successfully completed.

Apel Subprogram: v_nr=2

```

set serveroutput on;
declare
  v_rezultat varchar2(100);
  v_nr number := 4p_nr;
begin
  v_rezultat := OrasProdus4(v_nr);
  DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || '' || v_rezultat);
end;
/

```

Script Output | Query Result | Task completed in 0.593 seconds

IDProdus: 2 Produsul nu este in niciun magazin

Apel Subprogram: v_nr=3

```

set serveroutput on;
declare
  v_rezultat varchar2(100);
  v_nr number := 4p_nr;
begin
  v_rezultat := OrasProdus4(v_nr);
  DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || '' || v_rezultat);
end;
A

```

Script Output | Query Result | Task completed in 0.875 seconds

IDProdus: 3 Produsul se afla in mai multe magazine

PL/SQL procedure successfully completed.

Apel Subprogram: v_nr=-1

```
set serveroutput on;
declare
    v_rezultat varchar2(100);
    v_nr number := 4p_nr;
begin
    v_rezultat := OrasProdus4(v_nr);
    DBMS_OUTPUT.PUT_LINE('IDProdus: ' || v_nr || ' '|| v_rezultat);
end;
/

```

Script Output | Query Result | Task completed in 1.367 seconds

IDProdus: -1 Codul erorii: -20001 Mesajul erorii: ORA-20001: ID-ul produsului nu poate fi negativ

PL/SQL procedure successfully completed.

9. Subprogram stocat de tip procedura care utilizeaza 5 tabele.

Afisam clientii care au comandat orice carte, dar care au ultimele 2 cifre din numarul de telefon egale cu cele alese de noi si ca numarul de comenzi pe care clientul le-a dat sa fie egal cu numarul ales de noi.

```
create procedure ClCa1
(v_nr in number,
 v_li in varchar2
)
is
    v_nume Client.Nume%type;
    v_prenume Client.Prenume%type;

    ex_nrinvalid exception;
    ex_telefoninvalid exception;

begin
    if v_nr <= 0 then
        raise ex_nrinvalid;
    end if;

    if length(v_li) < 2 then
        raise ex_telefoninvalid;
    end if;

    select Cl.Nume,Cl.Prenume into v_nume, v_prenume
    from Client Cl, Comanda Cm, DetaliiComanda Dc, Produs P, Carte C
    where C.IDProdus = P.IDProdus
    and P.IDProdus = Dc.IDProdus
    and Dc.IDComanda = Cm.IDComanda
    and Cm.IDClient = Cl.IDClient
    and Cl.NumarDeTelefon like '%' || v_li
    and (select count(*) from Comanda Cm1 where Cm1.IDClient = Cl.IDClient group by IDClient)=v_nr;
```

```

DBMS_OUTPUT.PUT_LINE('Clientul este ' || v_nume || ' ' || v_prenume);

exception
when ex_nrinvalid then
DBMS_OUTPUT.PUT_LINE('Eroare: Numarul de comenzi trebuie sa fie mai mare ca 0');

when ex_telefoninvalid then
DBMS_OUTPUT.PUT_LINE('Eroare: Lungimea parametrului trebuie sa fie de cel putin 2 caractere');

when no_data_found then
DBMS_OUTPUT.PUT_LINE('Nu exista clienti cu specificatiile acestea');

when too_many_rows then
DBMS_OUTPUT.PUT_LINE('Există mai mult de 1 client cu specificatiile acestea');

when others then
DBMS_OUTPUT.PUT_LINE('Codul erorii: ' || SQLCODE || ' Mesajul erorii: ' || SQLERRM);

end;
/

```

declare

```

p_nr number := &p_nr;
p_li varchar2(2) := '&p_li';

begin
CICa1(v_nr=>p_nr, v_li=>p_li);
end;

```

```

create procedure CICa1
(v_nr in number,
v_li in varchar2
)
is
v_nume Client.Nume%type;
v_prenume Client.Prenume%type;
|
ex_nrinvalid exception;
ex_telefoninvalid exception;

begin
if v_nr <= 0 then
raise ex_nrinvalid;
end if;

if length(v_li) < 2 then
raise ex_telefoninvalid;
end if;

select Cl.Nume,Cl.Prenume into v_nume, v_prenume
from Client Cl, Comanda Cn, DetaliuComanda Dc, Produs P, Carte C
where C.IDProdus = P.IDProdus
and P.IDProdus = Dc.IDProdus
and Dc.IDComanda = Cn.IDComanda
and Cn.IDClient = Cl.IDClient
and Cl.NumerDeTelefon like '%' || v_li
and (select count(*) from Comanda Cnl where Cnl.IDClient = Cl.IDClient group by IDClient)=v_nr;

```

```

DBMS_OUTPUT.PUT_LINE('Clientul este ' || v_name || ' ' || v_prenume);

exception
when ex_clientinvalid then
DBMS_OUTPUT.PUT_LINE('Eroare: Numarul de comenzi trebuie sa fie mai mare ca 0');
when ex_telenonvalid then
DBMS_OUTPUT.PUT_LINE('Eroare: Lungimea parametrului trebuie sa fie de cel putin 2 caractere');
when no_data_found then
DBMS_OUTPUT.PUT_LINE('Nu exista clienti cu specificatiile acestea');
when too_many_rows then
DBMS_OUTPUT.PUT_LINE('Există mai mult de 1 client cu specificatiile acestea');
when others then
DBMS_OUTPUT.PUT_LINE('Codul erorii: ' || SQLCODE || ' Mesajul erorii: ' || SQLERRM);
end;
/

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Apel subprogram: p_nr=2 si p_li=32

```

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Script Output | Query Result | Task completed in 2.613 seconds

Clientul este Mutuleasa Andrei

PL/SQL procedure successfully completed.

Apel subprogram: p_nr=1 si p_li=12

```

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Script Output | Query Result | Task completed in 2.622 seconds

Există mai mult de 1 client cu specificatiile acestea

Apel subprogram: p_nr=3 si p_li=43

```

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Script Output | Query Result | Task completed in 2.083 seconds

Nu există clienti cu specificatiile acestea

Apel subprogram: p_nr=-1 si p_li=44

```

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Script Output | Query Result | Task completed in 3.924 seconds

Eroare: Numarul de comenzi trebuie sa fie mai mare ca 0

PL/SQL procedure successfully completed.

Apel subprogram: p_nr=3 si p_li=2

```

declare
p_nr number := ap_nr;
p_li varchar2(2) := 'ap_li';
begin
CICal(v_nr=>p_nr, v_li=>p_li);
end;

```

Script Output | Query Result | Task completed in 19.762 seconds

Eroare: Lungimea parametrului trebuie sa fie de cel putin 2 caractere

PL/SQL procedure successfully completed.

10. Trigger LMD la nivel de comanda.

De fiecare data cand se adauga, modifica sau sterge un client o sa primim un mesaj cu actiunea care a fost realizata.

```
create trigger ClientAnaliza  
    after insert or delete or update on client  
begin  
if inserting then  
    DBMS_OUTPUT.PUT_LINE('A fost adaugat un client in baza de date');  
elsif updating then  
    DBMS_OUTPUT.PUT_LINE('Informatiile despre un client au fost actualizate');  
elsif deleting then  
    DBMS_OUTPUT.PUT_LINE('Un client a fost sters din baza de date');  
end if;  
end;
```

```
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada,  
AdresaDeEmail, DataNasterii)
```

```
VALUES (9, 'Damian', 'Ivanovici', '0743613236', 'Arad', 'Strada Aurel Vlaicu',  
'damiva@gmail.com', TO_DATE('2002-03-22', 'YYYY-MM-DD'));
```

```
delete from Client where IDClient=9;
```

```
update client set NumarDeTelefon='0777777777' where IDClient=1;
```

```
create trigger ClientAnaliza  
    after insert or delete or update on client  
begin  
if inserting then  
    DBMS_OUTPUT.PUT_LINE('A fost adaugat un client in baza de date');  
elsif updating then  
    DBMS_OUTPUT.PUT_LINE('Informatiile despre un client au fost actualizate');  
elsif deleting then  
    DBMS_OUTPUT.PUT_LINE('Un client a fost sters din baza de date');  
end if;  
end;  
  
INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)  
VALUES (9, 'Damian', 'Ivanovici', '0743613236', 'Arad', 'Strada Aurel Vlaicu', 'damiva@gmail.com', TO_DATE('2002-03-22', 'YYYY-MM-DD'));  
delete from Client where IDClient=9;  
update client set NumarDeTelefon='0777777777' where IDClient=1;
```

```

INSERT INTO Client (IDClient, Nume, Prenume, NumarDeTelefon, Oras, Strada, AdresaDeEmail, DataNasterii)
VALUES (6, 'Damian', 'Ivanovici', '0743613256', 'Arad', 'Strada Aurel Vlaicu', 'damiva@gmail.com', TO_DATE('2002-03-22', 'YYYY-MM-DD'));

Script Output | Query Result
| Task completed in 0.039 seconds

Rollback complete.

A fost adaugat un client in baza de date

1 row inserted.


```

IDCLIENT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	ADRESADEEMAIL	DATANASTERII
1	Avrindte	Ieodor	0743623136	Valcea	Strada I.L.Caragiale	avrindteic@gmail.com	30-SEP-04
2	4Mutuleasa	Andrei	0734241632	Craiova	Strada Mihai Eminescu	sorinbeta@gmail.com	08-JUL-90
3	7Ivanancu	David	0721353212	Braila	Strada Ion Creanga	omeganajoc@yahoo.com	20-MAY-98
4	2Bortez	Marcus	0725612351	Galati	Strada Lucian Blaga	ioscarnd3@yahoo.com	03-APR-96
5	6Ciocan	Vlad	0798312312	Timis	Strada Nicita Stanescu	mamani1251@gmail.com	17-JAN-01
6	3Nistor	Georgian	0712351241	Cluj	Strada Nicolae Iorga	jupiter3214@gmail.com	12-DEC-02
7	8Anghel	Dan	0736224432	Iasi	Strada Nicolae Grigorescu	assasic1@yahoo.com	03-JUL-99
8	5Deaconu	Mihnea	0723152512	Constanta	Strada George Enescu	dadarace5@yahoo.com	21-SEP-06
9	9Damian	Ivanovici	0743613256	Arad	Strada Aurel Vlaicu	damiva@gmail.com	22-MAR-02

```

delete from Client where IDclient=9;

Script Output | Query Result
| Task completed in 0.025 seconds

Un client a fost sters din baza de date

1 row deleted.


```

IDCLIENT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	ADRESADEEMAIL	DATANASTERII
1	Avrindte	Ieodor	0743623136	Valcea	Strada I.L.Caragiale	avrindteic@gmail.com	30-SEP-04
2	4Mutuleasa	Andrei	0734241632	Craiova	Strada Mihai Eminescu	sorinbeta@gmail.com	08-JUL-90
3	7Ivanancu	David	0721353212	Braila	Strada Ion Creanga	omeganajoc@yahoo.com	20-MAY-98
4	2Bortez	Marcus	0725612351	Galati	Strada Lucian Blaga	ioscarnd3@yahoo.com	03-APR-96
5	6Ciocan	Vlad	0798312312	Timis	Strada Nicita Stanescu	mamani1251@gmail.com	17-JAN-01
6	3Nistor	Georgian	0712351241	Cluj	Strada Nicolae Iorga	jupiter3214@gmail.com	12-DEC-02
7	8Anghel	Dan	0736224432	Iasi	Strada Nicolae Grigorescu	assasic1@yahoo.com	03-JUL-99
8	5Deaconu	Mihnea	0723152512	Constanta	Strada George Enescu	dadarace5@yahoo.com	21-SEP-06

```

update client set NumarDeTelefon='0777777777' where IDclient=1;

Script Output | Query Result
| Task completed in 0.035 seconds

Informatiile despre un client au fost actualizate

1 row updated.


```

IDCLIENT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	ADRESADEEMAIL	DATANASTERII
1	Avrindte	Ieodor	0777777777	Valcea	Strada I.L.Caragiale	avrindteic@gmail.com	30-SEP-04
2	4Mutuleasa	Andrei	0734241632	Craiova	Strada Mihai Eminescu	sorinbeta@gmail.com	08-JUL-90
3	7Ivanancu	David	0721353212	Braila	Strada Ion Creanga	omeganajoc@yahoo.com	20-MAY-98
4	2Bortez	Marcus	0725612351	Galati	Strada Lucian Blaga	ioscarnd3@yahoo.com	03-APR-96
5	6Ciocan	Vlad	0798312312	Timis	Strada Nicita Stanescu	mamani1251@gmail.com	17-JAN-01
6	3Nistor	Georgian	0712351241	Cluj	Strada Nicolae Iorga	jupiter3214@gmail.com	12-DEC-02
7	8Anghel	Dan	0736224432	Iasi	Strada Nicolae Grigorescu	assasic1@yahoo.com	03-JUL-99
8	5Deaconu	Mihnea	0723152512	Constanta	Strada George Enescu	dadarace5@yahoo.com	21-SEP-06

11. Trigger LMD la nivel de linie.

De fiecare data cand stergem un Job toti angajatii care lucrau job-ul respectiv vor fi stersi la randul lor din baza de date. Daca numarul de ore pe saptamana la un anumit job creste cu 5 ore vom da afara un angajat.

```
create procedure Anga(v_id Angajat.IDJob%type)
```

```
is
```

```
begin
```

```
    delete from Angajat where v_id=IDJob;
```

```
end;
```

```
create trigger JobAng
```

```
after delete or update of OrePeSaptamana on Job
```

```
for each row
```

```
begin
```

```
if deleting then
```

```
    Anga(:old.IDJob);
```

```
elsif updating then
```

```
    if :old.OrePeSaptamana+5<=:new.OrePeSaptamana then
```

```
        delete from Angajat where IDAngajat =
```

```
            (Select IDAngajat from Angajat where IDJob = :old.IDJob and
rownum=1);
```

```
    end if;
```

```
end if;
```

```
end;
```

```
delete from Job where IDJob=1;
```

update Job set OrePeSaptamana=40 where IDJob=3;

```

create procedure Anga(v_id Angajat.IDJob&type)
is
begin
    delete from Angajat where v_id=IDJob;
end;

create trigger JobAng
after delete or update of OrePeSaptamana on Job
for each row
begin
    if deleting then
        Anga(:old.IDJob);
    elsif updating then
        if :old.OrePeSaptamana+5<=:new.OrePeSaptamana then
            delete from Angajat where IDAngajat =
                (Select IDAngajat from Angajat where IDJob = :old.IDJob and rownum=1);
        end if;
    end if;
end;

delete from Job where IDJob=1;
update Job set OrePeSaptamana=40 where IDJob=3;
```

IDANGAJAT	NUME	PRENUME	NUMARDETELEFON	ORAS	STRADA	IDJOB	ADRESADEEMAIL	DATANASTERII	IDCONCEDIU	IDMAGAZIN
1	I Popescu	Ion	0712345678	Bucuresti	Str Libertatii	1.popescu.ion@gmail.com	15-MAY-90	3	1	
2	2 Ionescu	Maria	0723456789	Cluj-Napoca	Str Unirii	2.ionescu.maria@gmail.com	25-AUG-85	6	5	
3	3 Medelciu	Andrei	0734567890	Timisoara	Str Victoriei	3.medelciu.andrei@yahoo.com	10-NOV-93	4	2	
4	4 Pana	Elena	0745678901	Iasi	Str Stefan cel Mare	4.pana.elena@gmail.com	20-APR-88	1	4	
5	5 Cotoarba	Cristian	0756789012	Constanta	Str Mihai Viteazu	5.cotoarba.cristian@gmail.com	05-SEP-95	2	3	
6	6 Georgeescu	Ana	0767890123	Brasov	Str Brasovului	6.georgeescuана@gmail.com	15-FEB-91	3	2	
7	7 Stan	Mihai	0778901234	Ploiesti	Str Mihai Eminescu	7.stan.mihai@yahoo.com	30-JUL-87	1	3	
8	8 Fotcovariu	Mihela	0789012345	Sibiu	Str Aviatorilor	8.fotcovariu.mihela@gmail.com	10-DEC-94	4	1	
9	9 Stoica	Catalin	0790123456	Galati	Str Eroilor	9.stoica.catalin@gmail.com	25-MAR-90	6	5	
10	10 Flores	Ana-Maria	0752303443	Bacau	Str Republicii	10.flores.anamaria@gmail.com	20-JUN-92	2		

Delete from Job where IDJob=1;										
Script Output X Query Result X										
All Rows Fetched: 7 in 0.001 seconds										
1.popescu.ion@gmail.com										
2.ionescu.maria@gmail.com										
3.medelciu.andrei@yahoo.com										
4.pana.elena@gmail.com										
5.cotoarba.cristian@gmail.com										
6.georgeescuана@gmail.com										
7.stan.mihai@yahoo.com										
8.fotcovariu.mihela@gmail.com										
9.stoica.catalin@gmail.com										
10.flores.anamaria@gmail.com										

update Job set OrePeSaptamana=40 where IDJob=3;										
Script Output X Query Result X										
All Rows Fetched: 9 in 0.001 seconds										
1.popescu.ion@gmail.com										
2.ionescu.maria@gmail.com										
3.medelciu.andrei@yahoo.com										
4.pana.elena@gmail.com										
5.cotoarba.cristian@gmail.com										
6.georgeescuана@gmail.com										
7.stan.mihai@yahoo.com										
8.fotcovariu.mihela@gmail.com										
9.stoica.catalin@gmail.com										
10.flores.anamaria@gmail.com										

12. Trigger LDD.

Nu avem voie sa modificam nimic la structura tabelelor Regizor, Artist si Autor si vom primi o eroare cand incercam sa facem asta.

```
create or replace trigger SchimbarePersonalitatii  
before alter  
on schema  
begin  
if upper(ora_dict_obj_name) = 'REGIZOR' then  
    raise_application_error(-20002, 'Nu ai voie sa modifici tabelul Regizor');  
elsif upper(ora_dict_obj_name) = 'AUTOR' then  
    raise_application_error(-20003, 'Nu ai voie sa modifici tabelul Autor');  
elsif upper(ora_dict_obj_name) = 'ARTIST' then  
    raise_application_error(-20004, 'Nu ai voie sa modifici tabelul Artist');  
end if;  
end;  
/  
/
```

```
alter table Regizor add ultimulfilm varchar2(30);  
alter table Autor add ultimacarte varchar2(30);  
alter table Artist add ultimulalbum varchar2(30);
```

The screenshot shows the Oracle SQL Developer interface with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the PL/SQL code for creating the trigger and altering the three tables. The 'Query Result' tab shows the execution of the alter statements, followed by an error message indicating that the trigger was triggered during the execution of the alter statements, which caused a recursive SQL level 1 error. The error details mention ORA-04009 and ORA-06404 errors, along with ORA-20002, ORA-06512, and ORA-04088 errors, all pointing to the trigger 'SYSTEM.SCHIMBAREPERSONALITATI'.

```
create or replace trigger SchimbarePersonalitatii  
before alter  
on schema  
begin  
if upper(ora_dict_obj_name) = 'REGIZOR' then  
    raise_application_error(-20002, 'Nu ai voie sa modifici tabelul Regizor');  
elsif upper(ora_dict_obj_name) = 'AUTOR' then  
    raise_application_error(-20003, 'Nu ai voie sa modifici tabelul Autor');  
elsif upper(ora_dict_obj_name) = 'ARTIST' then  
    raise_application_error(-20004, 'Nu ai voie sa modifici tabelul Artist');  
end if;  
end;  
/  
/
```

```
alter table Regizor add ultimulfilm varchar2(30);  
alter table Autor add ultimacarte varchar2(30);  
alter table Artist add ultimulalbum varchar2(30);
```

```
alter table Regizor add ultimulfilm varchar2(30);  
alter table Autor add ultimacarte varchar2(30);  
alter table Artist add ultimulalbum varchar2(30);
```

```
alter table Regizor add ultimulfilm varchar2(30);  
alter table Autor add ultimacarte varchar2(30);  
alter table Artist add ultimulalbum varchar2(30);
```

```
ORA-04009: error during execution of trigger 'SYSTEM.SCHIMBAREPERSONALITATI'  
ORA-06404: error occurred at recursive SQL level 1  
ORA-20002: Nu ai voie sa modifici tabelul Regizor  
ORA-06512: at line 3  
04088. 00000 - "error during execution of trigger '%s.%s'"  
*Cause: A runtime error occurred during execution of a trigger.  
*Action: Check the triggers which were involved in the operation.
```

```
| alter table Autor add ultimacarte varchar2(30); |
```

Script Output X | Query Result X
| Task completed in 0.036 seconds

```
alter table Autor add ultimacarte varchar2(30)
Error report -
ORA-04081: error during execution of trigger 'SYSTEM.SCHIMBAREPERSONALITATI'
ORA-00604: error occurred at recursive SQL level 1
ORA-20004: Nu ai voie sa modifici tabelul Autor
ORA-06512: at line 5
ORA-0888: 00000 - "error during execution of trigger '#s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.
```

```
| alter table Artist add ultimyalbum varchar2(30); |
```

Script Output X | Query Result X
| Task completed in 0.035 seconds

```
alter table Artist add ultimyalbum varchar2(30)
Error report -
ORA-04081: error during execution of trigger 'SYSTEM.SCHIMBAREPERSONALITATI'
ORA-00604: error occurred at recursive SQL level 1
ORA-20004: Nu ai voie sa modifici tabelul Artist
ORA-06512: at line 7
ORA-0888: 00000 - "error during execution of trigger '#s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.
```