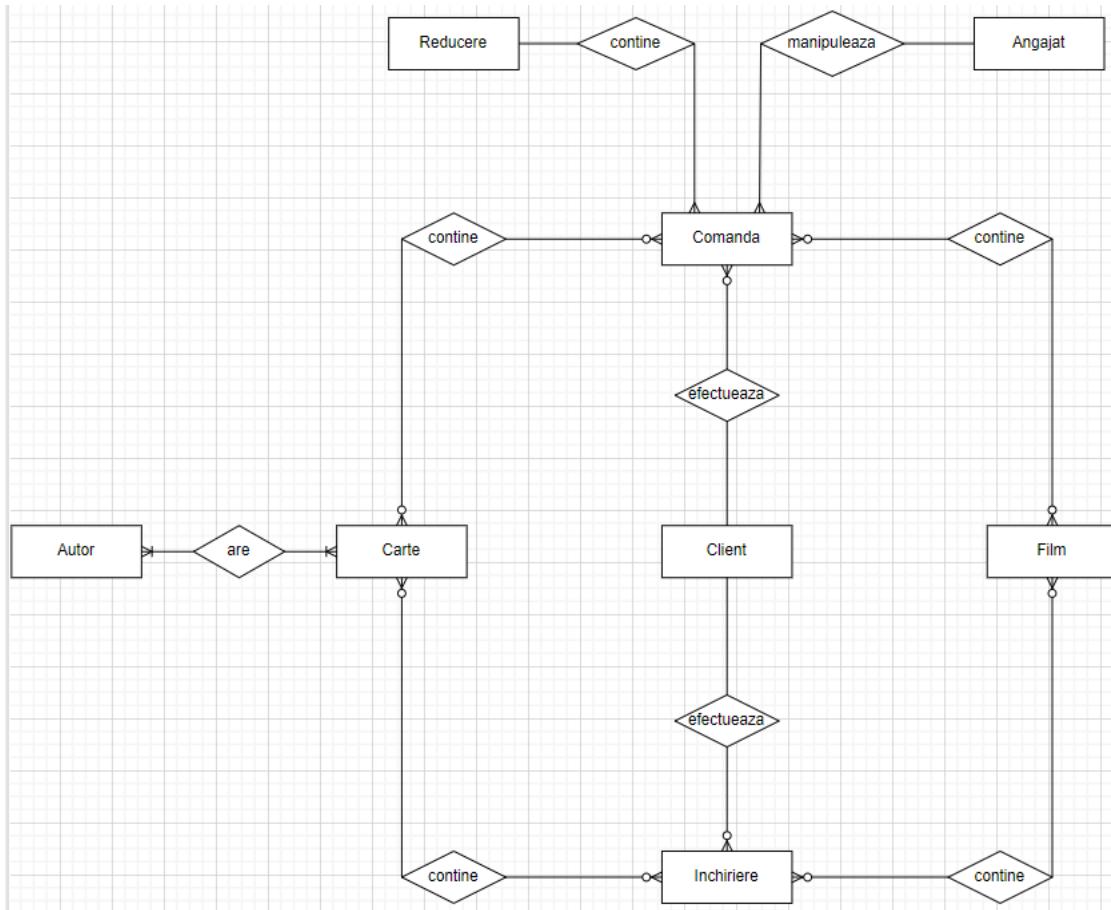


# Proiect Sisteme de Gestiune a Bazelor de Date

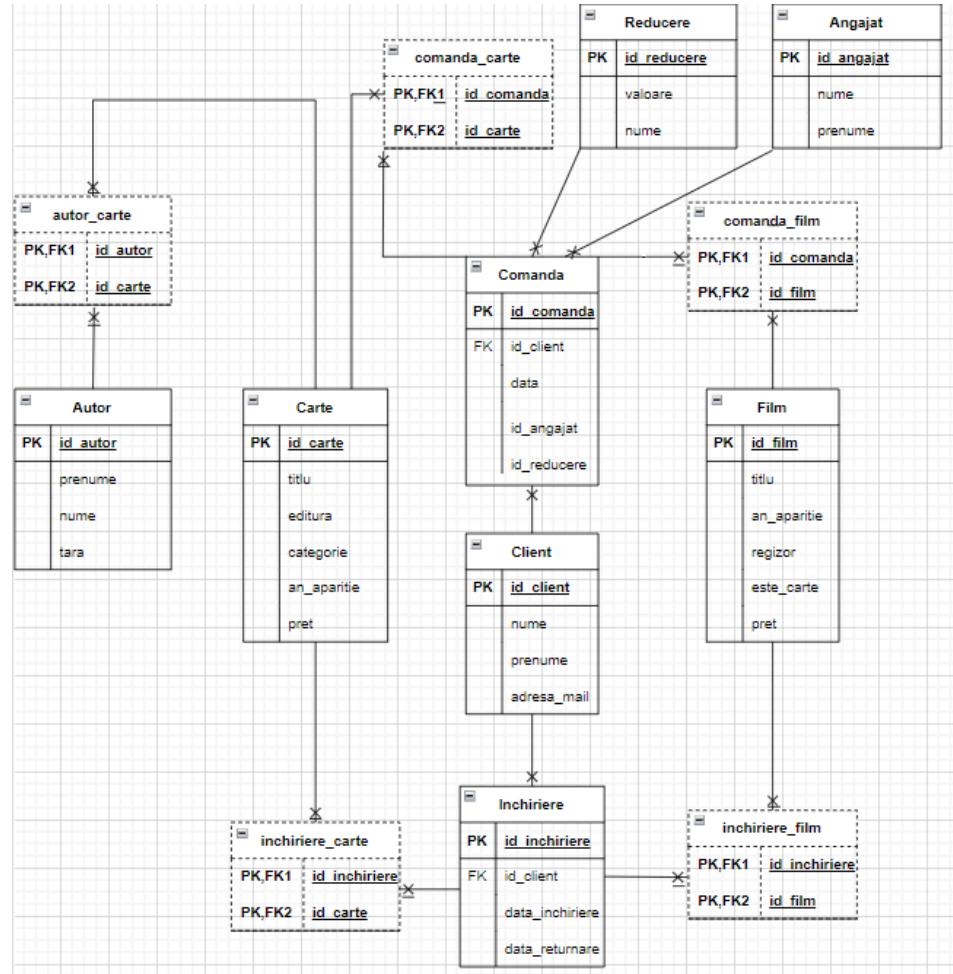
Voicu Bianca-Oana, grupa 252

1. Baza de date este pentru comenzi/ închirieri de cărți (E-books) și/sau filme. Acestea sunt comandate/ închiriate de către client folosindu-se de o adresă de mail. Clienții primesc la închiriere un link care devine inactiv după 30 de zile, dacă nu cumva cărțile sau filmele sunt returnate mai devreme. În cazul în care acestea sunt cumpărate, sunt primite ca atașament pe adresa de mail a clientului. Pentru închiriat se plătește un sfert din prețul întreg al cărții/ filmului. Un client poate efectua o comandă care conține cărți și/sau filme. La fel și în cazul închirierii. O comandă are reducere între 0 și 50%. Pentru fiecare comandă există un angajat care o manipulează. Baza de date conține separat o tabelă pentru autor, deoarece unele cărți au mai mulți autori. De asemenea, dacă pentru o carte există și un film, acest lucru este precizat în tabela film și numele acestora coincid.

## 2. Diagrama entitate-relație



### 3. Diagrama conceptuală



### 4.

```
CREATE TABLE AUTOR(
    id_autor NUMBER NOT NULL,
    prenume VARCHAR2(20),
    nume VARCHAR2(20),
    tara VARCHAR2(20),
    PRIMARY KEY(id_autor)
);
```

```
CREATE TABLE CARTE (
    id_carte NUMBER NOT NULL,
    titlu VARCHAR2(50),
    editura VARCHAR2(20),
    categorie VARCHAR2(20),
    an_aparitie NUMBER(4),
    pret NUMBER(4),
```

```

        PRIMARY KEY(id_carte) );

CREATE TABLE AUTOR_CARTE(
    id_autor NUMBER NOT NULL,
    id_carte NUMBER NOT NULL,
    PRIMARY KEY(id_autor, id_carte),
    FOREIGN KEY (id_autor)
    REFERENCES AUTOR(id_autor),
    FOREIGN KEY (id_carte)
    REFERENCES CARTE(id_carte)
);

CREATE TABLE CLIENT(
    id_client NUMBER NOT NULL,
    nume VARCHAR2(20),
    prenume VARCHAR2(20),
    adresa_mail VARCHAR2(50),
    PRIMARY KEY(id_client)
);

CREATE TABLE COMANDA(
    id_comanda NUMBER NOT NULL,
    id_client NUMBER,
    data DATE,
    id_angajat NUMBER,
    id_reducere NUMBER,
    PRIMARY KEY(id_comanda),
    FOREIGN KEY (id_client)
    REFERENCES CLIENT(id_client),
    FOREIGN KEY (id_angajat)
    REFERENCES ANGAJAT(id_angajat),
    FOREIGN KEY (id_reducere)
    REFERENCES REDUCERE(id_reducere)
);
CREATE TABLE COMANDA_CARTE(
    id_comanda NUMBER NOT NULL,
    id_carte NUMBER NOT NULL,
    PRIMARY KEY(id_comanda, id_carte),
    FOREIGN KEY (id_comanda)
    REFERENCES COMANDA(id_comanda),
    FOREIGN KEY (id_carte)
    REFERENCES CARTE(id_carte)
);

CREATE TABLE FILM(
    id_film NUMBER NOT NULL,
    titlu VARCHAR2(50),
    an_aparitie NUMBER(4),
    regizor VARCHAR2(50),
    este_carte VARCHAR2(20),
    pret NUMBER(4),
    PRIMARY KEY(id_film)
);

```

```

CREATE TABLE COMANDA_FILM(
    id_comanda NUMBER NOT NULL,
    id_film NUMBER NOT NULL,
    PRIMARY KEY(id_comanda, id_film),
    FOREIGN KEY (id_comanda)
    REFERENCES COMANDA(id_comanda),
    FOREIGN KEY (id_film)
    REFERENCES FILM(id_film)
);

CREATE TABLE INCHIRIERE(
    id_inchiriere NUMBER NOT NULL,
    id_client NUMBER,
    data_inchiriere DATE,
    data_returnare DATE,
    PRIMARY KEY(id_inchiriere),
    FOREIGN KEY (id_client)
    REFERENCES CLIENT(id_client)
);

CREATE TABLE INCHIRIERE_FILM(
    id_inchiriere NUMBER NOT NULL,
    id_film NUMBER NOT NULL,
    PRIMARY KEY(id_inchiriere, id_film),
    FOREIGN KEY (id_inchiriere)
    REFERENCES INCHIRIERE(id_inchiriere),
    FOREIGN KEY (id_film)
    REFERENCES FILM(id_film)
);

CREATE TABLE INCHIRIERE_CARTE(
    id_inchiriere NUMBER NOT NULL,
    id_carte NUMBER NOT NULL,
    PRIMARY KEY(id_inchiriere, id_carte),
    FOREIGN KEY (id_inchiriere)
    REFERENCES INCHIRIERE(id_inchiriere),
    FOREIGN KEY (id_carte)
    REFERENCES CARTE(id_carte)
);

CREATE TABLE ANGAJAT(
    id_angajat NUMBER NOT NULL,
    nume VARCHAR2(20),
    prenume VARCHAR2(20),
    PRIMARY KEY(id_angajat)
);

CREATE TABLE REDUCERE(
    id_reducere NUMBER NOT NULL,
    valoare NUMBER,
    nume VARCHAR2(50),
    PRIMARY KEY(id_reducere)
);

```

## **5. Tabela autor:**

```
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (1, 'Cassandra', 'Clare', 'Iran');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (2, 'Sarah', 'Rees Brennan', 'Irlanda');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (3, 'Maureen', 'Johnson', 'America');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (4, 'Agatha', 'Christie', 'Anglia');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (5, 'Federico', 'Moccia', 'Italia');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (6, 'Amos', 'Oz', 'Israel');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (7, 'Jane', 'Austen', 'Anglia');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (8, 'Margaret', 'Mitchell', 'America');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (9, 'Lev', 'Tolstoy', 'Rusia');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (10, 'Miguel', 'de Cervantes', 'Spania');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (11, 'Andreea', 'Russo', 'Moldova');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (12, 'Lauren', 'Oliver', 'America');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (13, 'Kerri', 'Maniscalco', 'America');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (14, 'Joe', 'Dispenza', 'America');
INSERT INTO AUTOR (ID_AUTOR, PRENUME, NUME, TARA)
VALUES (15, 'Daniel', 'Goleman', 'America');
```

## **Tabela carte:**

```
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (1, 'Cronicile lui Magnus Bane', 'Leda', 'fictiune young adult', 2017, 50);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (2, 'Ingerul mecanic', 'Leda', 'fictiune young adult', 2011, 50);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (3, 'Sabrina, intre lumina si intuneric', 'Corint', 'fictiune young adult', 2020, 38);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (4, 'Crima din Orient Express', 'Litera', 'thriller', 2014, 20);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (5, 'Zece negri mititei', 'Litera', 'thriller', 2018, 50);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (6, 'Trei metri deasupra cerului', 'Bestseller', 'fictiune young adult', 2018, 70);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
```

```

VALUES (7, 'Deodata in adancul padurii', 'Humanitas', 'fictiune ', 2017,
25);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (8, 'Mandrie si prejudecata', 'RAO', 'fictiune', 2017, 36);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (9, 'Pe aripile vantului ', 'Litera', 'fictiune', 2019, 50);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (10, 'Anna Karenina', 'Polirom', 'fictiune', 2016, 33);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (11, 'Don Quijote', 'Prut', 'fictiune copii', 2021, 25);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (12, 'Fata cu vise alb-negru', 'Bestseller', 'fictiune young
adult', 2017, 60);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (13, 'Delirium', 'Nemira', 'fictiune young adult', 2014, 37);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (14, 'Jack Spintecatorul. Crimele din Whitechapel', 'Corint',
'thriller', 2018, 44);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (15, 'Distruge-ti obiceiurile nocive!', 'Curtea Veche', 'dezvoltare
personală', 2019, 55);
INSERT INTO CARTE (ID_CARTE, TITLU, EDITURA, CATEGORIE, AN_APARITIE, PRET)
VALUES (16, 'Inteligenta emotionala', 'Curtea Veche', 'dezvoltare
personală', 2018, 57);

```

***Tabela autor\_carte:***

```

INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (1, 1);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (2, 1);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (3, 1);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (4, 4);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (5, 6);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (6, 7);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (7, 8);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (8, 9);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (9, 10);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (10, 11);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (11, 12);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (12, 13);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)

```

```

VALUES (13, 14);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (14, 15);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (15, 16);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (1, 2);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (4, 5);
INSERT INTO AUTOR_CARTE (ID_AUTOR, ID_CARTE)
VALUES (2, 3);

```

**Tabela client:**

```

INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (1, 'Sebe', 'Lavinia', 'lavi.sebe13@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (2, 'Baceanu', 'Bianca', 'biancab@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (3, 'Coman', 'Camelia', 'cami.elena@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (4, 'Belu', 'Roxana', 'b.roxana23@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (5, 'Bugheciu', 'Eduard', 'edi_bugheciu@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (6, 'Sin', 'Costin', 'sin.costin@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (7, 'Popescu', 'Georgian', 'geo_popescu5@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (8, 'Savu', 'Paul', 's.paul98@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (9, 'Constantin', 'Monica', 'monica.constantin@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (10, 'Dobre', 'Daniela', 'dani_dobre9@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (11, 'Dumitru', 'Catalin', 'dumi.cata17@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (12, 'Radu', 'Diana', 'radudiana27@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (13, 'Ionescu', 'Teodora', 'ionescu_teo1@gmail.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (14, 'Stelea', 'Sebastian', 's.sebastian99@yahoo.com');
INSERT INTO CLIENT (ID_CLIENT, NUME, PRENUME, ADRESA_MAIL)
VALUES (15, 'Vasile', 'Andreea', 'andreeeavasile85@gmail.com');

```

**Tabela comanda:**

```

INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (1, 2, to_date('10/17/2021', 'MM/DD/YYYY'), 1, 1);

```

```

INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (2, 5, to_date('11/25/2021', 'MM/DD/YYYY'), 2, 2);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (3, 6, to_date('09/08/2021', 'MM/DD/YYYY'), 3, 3);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (4, 15, to_date('10/03/2021', 'MM/DD/YYYY'), 4, 4);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (5, 11, to_date('12/05/2021', 'MM/DD/YYYY'), 5, 5);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (6, 3, to_date('09/20/2021', 'MM/DD/YYYY'), 6, 5);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (7, 14, to_date('11/11/2021', 'MM/DD/YYYY'), 1, 5);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (8, 10, to_date('12/18/2021', 'MM/DD/YYYY'), 2, 4);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (9, 1, to_date('12/10/2021', 'MM/DD/YYYY'), 3, 3);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (10, 7, to_date('10/01/2021', 'MM/DD/YYYY'), 4, 5);
INSERT INTO COMANDA (ID_COMANDA, ID_CLIENT, DATA)
VALUES (11, 2, to_date('10/21/2021', 'MM/DD/YYYY'), 5, 4);

```

**Tabela comanda\_carte:**

```

INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (1, 4);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (1, 5);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (1, 16);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (2, 7);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (2, 15);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (3, 11);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (4, 3);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (4, 12);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (9, 1);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (9, 2);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (9, 13);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (6, 10);
INSERT INTO COMANDA_CARTE (ID_COMANDA, ID_CARTE)
VALUES (11, 4);

```

### **Tabela film:**

```
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (1, 'Cronicile Craciunului', 2018, 'Clay Kaytis', 'Nu', 30);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (2, 'Crima din Orient Express', 2017, 'Kenneth Branagh', 'Da', 50);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (3, 'Zece negri mititei', 1965, 'George Pollock', 'Da', 25);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (4, 'Maleficent', 2014, 'Robert Stromberg', 'Nu', 45);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (5, 'Trei metri deasupra cerului', 2010, 'Fernando González Molina', 'Da', 35);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (6, 'Bohemian Rhapsody', 2018, 'Bryan Singer', 'Nu', 50);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (7, 'Mandrie si prejudecata', 2005, 'Joe Wright', 'Da', 40);-
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (8, 'Pe aripile vantului', 1939, 'Victor Fleming', 'Da', 35);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (9, 'Secretul lui Adaline', 2015, 'Lee Toland Krieger', 'Nu', 30);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (10, 'Anna Karenina', 2012, 'Joe Wright', 'Da', 30);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (11, 'Un schimb regal', 2018, 'Michael Rohl', 'Nu', 35);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (12, 'Un castel de Craciun', 2021, 'Mary Lambert', 'Nu', 25);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (13, 'Don Quijote', 2018, 'Terry Gilliam', 'Da', 35);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (14, 'Anna', 2019, 'Luc Besson', 'Nu', 40);
INSERT INTO FILM (ID_FILM, TITLU, AN_APARITIE, REGIZOR, ESTE_CARTE, PRET)
VALUES (15, 'O simpla favoare', 2018, 'Paul Feig', 'Nu', 45);
```

### **Tabela comanda\_film:**

```
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (1, 2);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (1, 3);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (3, 6);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (5, 2);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (5, 13);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (5, 14);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (6, 1);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (6, 11);
```

```

INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (6, 12);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (7, 3);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (8, 5);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (8, 9);
INSERT INTO COMANDA_FILM (ID_COMANDA, ID_FILM)
VALUES (10, 15);

```

**Tabela inchiriere:**

```

INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (1, 7, to_date('11/15/2021', 'MM/DD/YYYY'), to_date('12/15/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (2, 10, to_date('10/08/2021', 'MM/DD/YYYY'), to_date('11/07/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (3, 8, to_date('09/05/2021', 'MM/DD/YYYY'), to_date('10/05/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (4, 12, to_date('10/17/2021', 'MM/DD/YYYY'), to_date('10/31/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (5, 9, to_date('09/21/2021', 'MM/DD/YYYY'), to_date('10/11/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (6, 4, to_date('10/07/2021', 'MM/DD/YYYY'), to_date('11/06/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (7, 1, to_date('12/02/2021', 'MM/DD/YYYY'), to_date('12/18/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (8, 2, to_date('09/23/2021', 'MM/DD/YYYY'), to_date('10/09/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)
VALUES (9, 15, to_date('10/28/2021', 'MM/DD/YYYY'), to_date('11/27/2021',
'MM/DD/YYYY'));
INSERT INTO INCHIRIERE (ID_INCHIRIERE, ID_CLIENT, DATA_INCHIRIERE,
DATA_RETURNARE)

```

```
VALUES (10, 13, to_date('11/12/2021', 'MM/DD/YYYY'), to_date('11/29/2021', 'MM/DD/YYYY'));
```

**Tabela inchiriere\_film:**

```
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (3, 13);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (3, 14);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (5, 10);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (6, 4);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (6, 14);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (7, 9);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (8, 7);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (8, 10);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (9, 4);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (9, 5);
INSERT INTO INCHIRIERE_FILM (ID_INCHIRIERE, ID_FILM)
VALUES (10, 6);
```

**Tabela inchiriere\_carte:**

```
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (1, 11);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (1, 14);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (2, 8);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (2, 9);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (4, 3);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (4, 6);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (5, 10);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (7, 12);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (7, 13);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (8, 15);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
```

```

VALUES (9, 7);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (7, 16);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (8, 8);
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (8, 9);

```

**Tabela angajat:**

```

INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (1, 'Radu', 'Teodora');
INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (2, 'Georgescu', 'Vlad');
INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (3, 'Iancu', 'Daria');
INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (4, 'Dumitrache', 'Adriana');
INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (5, 'Nicu', 'Adrian');
INSERT INTO ANGAJAT (ID_ANGAJAT, NUME, PRENUME)
VALUES (6, 'Neagu', 'Mihaela');

```

**Tabela reducere:**

```

INSERT INTO REDUCERE (ID_REDUCERE, VALOARE, NUME)
VALUES (1, 0.5, 'Reducere Black Friday');
INSERT INTO REDUCERE (ID_REDUCERE, VALOARE, NUME)
VALUES (2, 0.4, 'Reducere surpriza');
INSERT INTO REDUCERE (ID_REDUCERE, VALOARE, NUME)
VALUES (3, 0.3, 'Reducere ocasionala');
INSERT INTO REDUCERE (ID_REDUCERE, VALOARE, NUME)
VALUES (4, 0.2, 'Reducere fulger');
INSERT INTO REDUCERE (ID_REDUCERE, VALOARE, NUME)
VALUES (5, 0, 'Fara reducere');

```

**6. Cu ajutorul unei proceduri să se afișeze:**

- a) autorii care provin din Anglia cu ajutorul unui tablou indexat, plus numărul total al acestora.
- b) titlurile filmelor cu ajutorul unui vector, plus numărul total al acestora.

```

CREATE OR REPLACE PROCEDURE proc_ex6
IS
  TYPE rec IS RECORD (a_prenume autor.prenume%TYPE, a_nume autor.nume%TYPE,
  a_tara autor.tara%TYPE);

```

```

TYPE tab_ind IS TABLE OF rec INDEX BY PLS_INTEGER;
t tab_ind;

TYPE tab_vec IS VARRAY(50) OF NUMBER;
tv tab_vec := tab_vec();
v_cnt NUMBER:= 0;
v_id NUMBER:= 0;
v_min NUMBER;
v_max NUMBER;
v_titlu film.titlu%TYPE;
BEGIN

SELECT prenume, nume, tara BULK COLLECT INTO t
FROM autor
WHERE LOWER(tara) = 'anglia';

DBMS_OUTPUT.PUT_LINE('Numarul de autori care provin din Anglia: ' ||
t.COUNT );
DBMS_OUTPUT.NEW_LINE;

FOR i IN t.FIRST..t.LAST LOOP
DBMS_OUTPUT.PUT_LINE('Autorul ' || t(i).a_prenume || ' ' || t(i).a_nume
|| ' ' || 'provine din tara: ' || t(i).a_tara);
END LOOP;

DBMS_OUTPUT.NEW_LINE;

SELECT MIN(id_film) INTO v_min
FROM film;

SELECT MAX(id_film) INTO v_max
FROM film;

FOR i IN v_min..v_max LOOP
v_id:=0;

SELECT id_film INTO v_id
FROM film
WHERE id_film = i;

IF v_id!=0 THEN
v_cnt:= v_cnt+1;
tv.EXTEND;
tv(v_cnt):=i;
END IF;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Numarul filmelor: ' || tv.COUNT);
DBMS_OUTPUT.NEW_LINE;

FOR i IN tv.FIRST..tv.LAST LOOP
SELECT titlu INTO v_titlu
FROM film
WHERE id_film = i;
DBMS_OUTPUT.PUT_LINE(tv(i) || ' ' || v_titlu);

```

```

END LOOP;
END;
/
EXEC proc_ex6;

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists several database connections, with 'project\_sgbd' selected. The 'Worksheet' tab in the main area contains the following PL/SQL code:

```

END;
/
EXEC proc_ex6;

```

The 'Script Output' tab shows the results of the execution:

```

Procedure PROC_EX6 compiled

Numarul de autori care provin din Anglia: 2

Autorul Agatha Christie provine din tara: Anglia
Autorul Jane Austen provine din tara: Anglia

Numarul filmelor: 15

1 Cronicile Craciunului
2 Crimi din Orient Express
3 Zece negri mititei
4 Maleficent
5 Trei metri deasupra cerului
6 Bohemian Rhapsody
7 Mandrie si prejudecata
8 Pe aripile vantului
9 Secretul lui Adaline
10 Anna Karenina
11 Un schimb regal
12 Un castel de Craciun
13 Don Quijote
14 Anna
15 O simpla favoare

```

The 'Query Result' tab shows the message: 'Task completed in 0.031 seconds'.

This screenshot is nearly identical to the one above, showing the same Oracle SQL Developer interface and the execution of the 'proc\_ex6' procedure. The 'Script Output' tab displays the same list of 15 books, and the 'Query Result' tab shows the task completion message.

7. Să se returneze cu ajutorul unei funcții cel mai mic an al apariției al unei cărți care este și film.

```

CREATE OR REPLACE FUNCTION func_ex7
RETURN carte.an_aparitie%TYPE IS
v_min carte.an_aparitie%TYPE:= 3000;
v_an carte.an_aparitie%TYPE;
CURSOR crs IS
    SELECT *
    FROM film
    WHERE LOWER(este_carte)= 'da';
BEGIN

FOR i in crs LOOP
SELECT an_aparitie INTO v_an
FROM carte
WHERE LOWER(titlu) = LOWER(i.titlu);

IF v_an < v_min THEN
v_min := v_an;
END IF;
END LOOP;
RETURN v_min;
END;
/
SELECT func_ex7
FROM DUAL;
/

```

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

- Connections:** Oracle Connections, project\_sgbdb (selected).
- Worksheet:** Query Builder window containing the PL/SQL code for the function func\_ex7.
- Script Output:** Shows the execution results of the function call SELECT func\_ex7 FROM DUAL; with the output being 1 2014.
- Compiler - Log:** A status bar at the bottom left.
- Bottom Status Bar:** Shows the current line (Line 32), column (Column 11), and other status information like Insert, Modified, Windows: C.

8. Să se returneze cu ajutorul unei funcții suma totală a comenzilor efectuate de un client cu id-ul dat ca parametru.

```
CREATE OR REPLACE FUNCTION func_ex8(p_id IN comanda.id_client%TYPE)
  RETURN NUMBER
IS
v_suma NUMBER:= 0;
v_id_client NUMBER;
v_select NUMBER;
v_id_com NUMBER;
v_reducere NUMBER;

CURSOR carti IS (SELECT *
FROM carte c JOIN comanda_carte cc ON c.id_carte=cc.id_carte
JOIN comanda com ON cc.id_comanda=com.id_comanda
WHERE com.id_client = p_id);

CURSOR filme IS (SELECT *
FROM film f JOIN comanda_film cf ON f.id_film=cf.id_film
JOIN comanda com ON cf.id_comanda=com.id_comanda
WHERE com.id_client = p_id);

BEGIN

v_select:= 1;
SELECT id_client INTO v_id_client
FROM client
WHERE id_client = p_id;

v_select:= 2;
SELECT id_comanda INTO v_id_com
FROM comanda
WHERE id_client = p_id AND ROWNUM = 1;

v_select:= 3;
SELECT r.valoare INTO v_reducere
FROM comanda c JOIN reducere r ON c.id_reducere=r.id_reducere
WHERE c.id_comanda = v_id_com;

FOR i IN carti LOOP
v_suma:= i.pret + v_suma;
END LOOP;

FOR j IN filme LOOP
v_suma:= j.pret + v_suma;
END LOOP;

v_suma:= v_suma - v_reducere*v_suma;

RETURN v_suma;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
```

```

    IF v_select=1 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista id-ul clientului!');
    ELSE
        RAISE_APPLICATION_ERROR(-20000, 'Clientul nu a efectuat nicio
comanda!');
    END IF;
END;
/

-- Pentru cazul in care nu exista id-ul clientului in baza de date
SELECT func_ex8(32)
FROM DUAL;
/
-- Pentru cazul in care clientul nu a efectuat nicio comanda
SELECT func_ex8(12)
FROM DUAL;
/
-- Pentru cazul in care se calculeaza suma
SELECT func_ex8(2)
FROM DUAL;
/

```

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree displays several database connections, including 'project\_sgbd'. The central area contains a 'Worksheet' tab with the following PL/SQL code:

```

    IF v_select=1 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Nu exista id-ul clientului!');
    ELSE
        RAISE_APPLICATION_ERROR(-20000, 'Clientul nu a efectuat nicio
comanda!');
    END IF;
END;
/

-- Pentru cazul in care nu exista id-ul clientului in baza de date
SELECT func_ex8(32)
FROM DUAL;
/
-- Pentru cazul in care clientul nu a efectuat nicio comanda
SELECT func_ex8(12)
FROM DUAL;
/
-- Pentru cazul in care se calculeaza suma
SELECT func_ex8(2)
FROM DUAL;
/

```

Below the worksheet, the 'Script Output' tab shows the execution results:

```

ORA-20000: Nu exista id-ul clientului
ORA-06512: at "TBANEA.FUNC_EX8", line 43
20000. 000 "%"*
*Cause:  The stored procedure 'raise_application_error'
        was called which causes this error to be generated.
*Action:  Correct the problem as described in the error message or contact
        the application administrator or DBA for more information.

```

Oracle SQL Developer : project\_sgbd

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - grupa251\_test
  - grupa252
  - lab
  - mateinfo
  - project
  - project\_sgbd
    - Tables (Filtered)
      - AUTOR
      - AUTOR\_CARTE
      - CARTE
      - CLIENT
      - COMANDA
      - COMANDA\_CARTE
      - COMANDA\_FILM
      - FILM
      - INCHIRIERE
      - INCHIRIERE\_CARTE
      - INCHIRIERE\_COPY
      - INCHIRIERE\_FILM
      - TABELLE\_STERSE
    - Views
    - Indexes
    - Packages
      - PACK\_EX13
      - PACK\_EX14
    - Procedures
    - Functions
    - Operators
    - Queues

Worksheet Query Builder

```

RAISE_APPLICATION_ERROR(-20000, 'Clientul nu a efectuat nicio comanda');

END IF;
/
-- Pentru cazul in care nu exista id-ul clientului in baza de date
SELECT func_ex8(32)
FROM DUAL;
/
-- Pentru cazul in care clientul nu a efectuat nicio comanda
SELECT func_ex8(12)
FROM DUAL;
/
-- Pentru cazul in care se calculeaza suma
SELECT func_ex8(2)
FROM DUAL;
/

```

Script Output x Query Result x

Executing:SELECT func\_ex8(12)FROM DUAL in 0 seconds

ORA-20000: Clientul nu a efectuat nicio comanda!  
ORA-06512: at "BIANCA.FUNC\_EX8", line 45  
20000. 00000 - "%s"  
\*Cause: The stored procedure 'raise\_application\_error'  
was called which causes this error to be generated.  
>Action: Correct the problem as described in the error message or contact  
the application administrator or DBA for more information.

Compiler - Log Click on an identifier with the Control key down to perform "Go to Declaration"

Line 60 Column 11 | Insert | Modified | Windows: C

Oracle SQL Developer : project\_sgbd

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - grupa251\_test
  - grupa252
  - lab
  - mateinfo
  - project
  - project\_sgbd
    - Tables (Filtered)
      - ANGAJAT
      - AUTOR
      - AUTOR\_CARTE
      - CARTE
      - CLIENT
      - COMANDA
      - COMANDA\_CARTE
      - COMANDA\_FILM
      - FILM
      - INCHIRIERE
      - INCHIRIERE\_CARTE
      - INCHIRIERE\_COPY
      - INCHIRIERE\_FILM
      - REDUCERE
      - TABELLE\_STERSE
    - Views
    - Indexes
    - Packages
    - Procedures
    - Functions
    - Operators
    - Queues
    - Queues Tables
    - Triggers
    - Types

Worksheet Query Builder

```

-- Pentru cazul in care nu exista id-ul clientului in baza de date
SELECT func_ex8(32)
FROM DUAL;
/
-- Pentru cazul in care clientul nu a efectuat nicio comanda
SELECT func_ex8(12)
FROM DUAL;
/
-- Pentru cazul in care se calculeaza suma
SELECT func_ex8(2)
FROM DUAL;
/

```

Script Output x Query Result x

All Rows Fetched: 1 in 0.002 seconds

	FUNC_EX8(2)
1	111

Upper Keywords

Line 8 Column 19 | Insert | Modified | Windows: C

- 9.** Să se afișeze numele și prenumele clientului care a închiriat cele mai multe cărți scrise de autori dintr-o țară dată ca parametru (pot să închirieze cărți scrise și de autori din alte țări).

```

CREATE OR REPLACE PROCEDURE proc_ex9(p_tara IN autor.tara%TYPE)
IS
v_nume autor.nume%TYPE;
v_prenume autor.prenume%TYPE;
v_max NUMBER;
v_cnt NUMBER:= 0;
v_id_client client.id_client%TYPE;
v_count NUMBER:= 0;
v_id_final client.id_client%TYPE;
v_tara autor.tara%TYPE;
v_ok NUMBER:=1;

CURSOR crs IS (SELECT client.id_client, COUNT(ca.id_carte)
FROM client
JOIN inchiriere i ON client.id_client=i.id_client JOIN
inchiriere_carte ic ON ic.id_inchiriere=i.id_inchiriere JOIN
carte ca ON ic.id_carte=ca.id_carte JOIN autor_carte ac ON
ac.id_carte=ca.id_carte JOIN autor a ON a.id_autor=ac.id_autor
WHERE LOWER(a.tara)=LOWER(p_tara)
GROUP BY client.id_client);

CURSOR inchirieri IS (SELECT id_inchiriere FROM inchiriere);

CURSOR tari IS (SELECT tara
                  FROM autor
                  WHERE LOWER(tara) = LOWER(p_tara) AND ROWNUM = 1);

exceptie EXCEPTION;

BEGIN

OPEN tari;
FETCH tari INTO v_tara;

IF tari%ROWCOUNT = 0 THEN
    RAISE NO_DATA_FOUND;
END IF;
CLOSE tari;

OPEN inchirieri;
IF inchirieri%NOTFOUND THEN
    RAISE exceptie;
END IF;

CLOSE inchirieri;

SELECT MAX(COUNT(ca.id_carte)) INTO v_max

```

```

FROM inchiriere i JOIN inchiriere_carte ic ON
ic.id_inchiriere=i.id_inchiriere
JOIN carte ca ON ic.id_carte=ca.id_carte JOIN autor_carte ac ON
ac.id_carte=ca.id_carte JOIN autor a ON a.id_autor=ac.id_autor
WHERE LOWER(a.tara)=lower(p_tara)
GROUP BY ca.id_carte;

OPEN crs;

LOOP
FETCH crs INTO v_id_client, v_count;
EXIT WHEN crs%NOTFOUND;

IF v_count=v_max THEN
v_id_final:=v_id_client;
v_cnt:= v_cnt +1;
END IF;

END LOOP;

IF v_cnt>1 THEN
RAISE TOO_MANY_ROWS;
END IF;

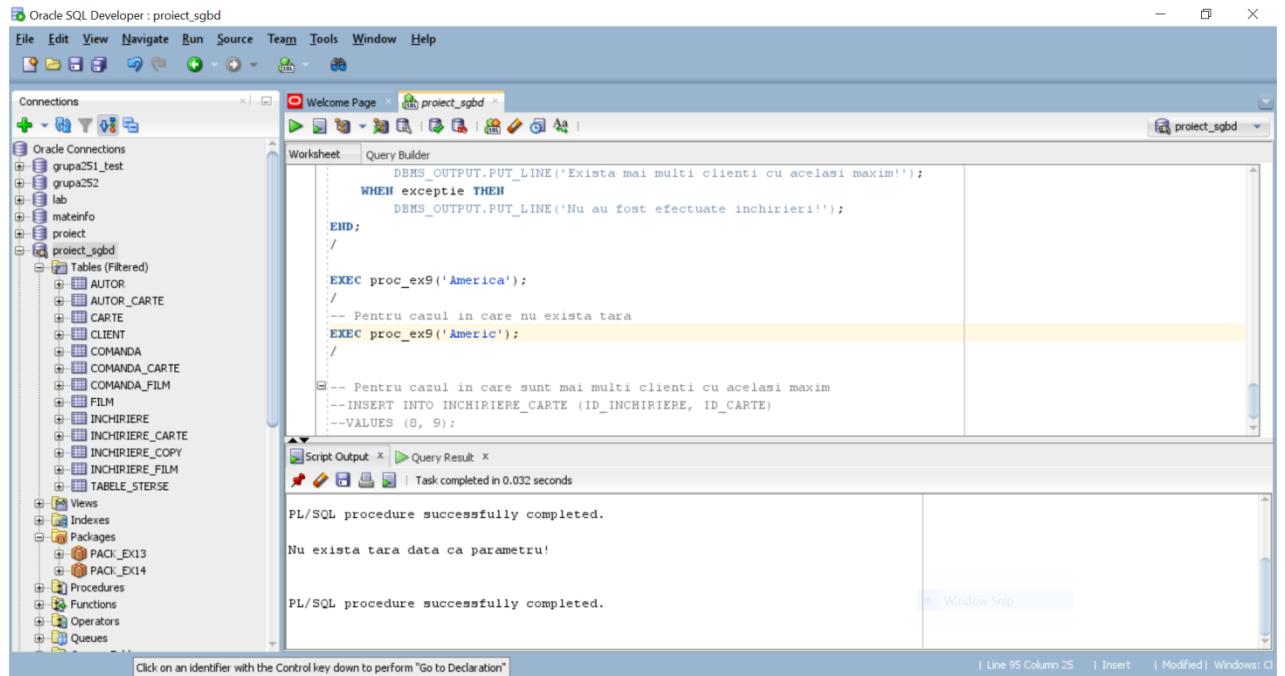
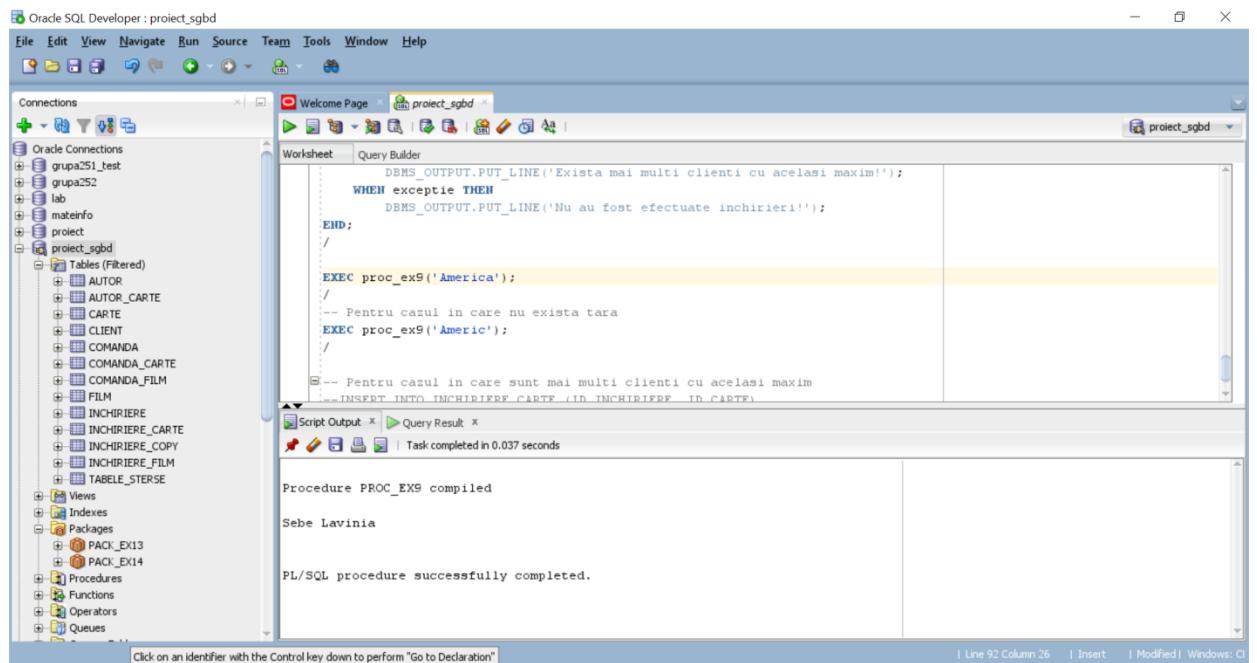
v_ok:=0;

SELECT nume, prenume INTO v_nume, v_prenume
FROM client
WHERE id_client = v_id_final;

DBMS_OUTPUT.PUT_LINE(v_nume || ' ' || v_prenume);
CLOSE crs;
EXCEPTION
WHEN exceptie THEN
    DBMS_OUTPUT.PUT_LINE('Nu au fost efectuate inchirieri!');
WHEN NO_DATA_FOUND THEN
    IF v_ok=1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista tara data ca parametru!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Niciun client nu a inchiriat carti din tara
data ca parametru!');
    END IF;
WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Există mai multi clienti cu același
maxim!');
END;
EXEC proc_ex9('America');
/
-- Pentru cazul in care nu exista tara
EXEC proc_ex9('Americ');
/
-- Pentru cazul in care sunt mai multi clienti cu același maxim
--INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)

```

```
--VALUES (8, 9);
EXEC proc_ex9('America');
/
-- Pentru cazul in care niciun client nu a inchiriat carti din tara data
ca parametru
EXEC proc_ex9('Anglia');
```



Oracle SQL Developer : project\_sgbdb

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - grupa251\_test
  - grupa252
  - lab
  - mateinfo
  - project
  - project\_sgbdb
    - Tables (Filtered)
      - AUTOR
      - AUTOR\_CARTE
      - CARTE
      - CLIENT
      - COMANDA
      - COMANDA\_CARTE
      - COMANDA\_FILM
      - FILM
      - INCHIRIERE
      - INCHIRIERE\_CARTE
      - INCHIRIERE\_COPY
      - INCHIRIERE\_FILM
      - TABELE\_STERSE
    - Views
    - Indexes
    - Packages
      - PACK\_EX13
      - PACK\_EX14
    - Procedures
    - Functions
    - Operators
    - Queues

Worksheet Query Builder

```

DBMS_OUTPUT.PUT_LINE('Nu au fost efectuate inchirieri!');

END;
/

EXEC proc_ex9('America');

-- Pentru cazul in care nu exista tara
EXEC proc_ex9('Americ');

-- Pentru cazul in care sunt mai multi clienti cu acelasi maxim
INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
VALUES (8, 9);
EXEC proc_ex9('America');
/

```

Script Output X Query Result X

Task completed in 0.034 seconds

1 row inserted.

Exista mai multi clienti cu acelasi maxim!

PL/SQL procedure successfully completed.

Windows Task

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

Line 101 Column 26 | Insert | Modified | Windows: C

Oracle SQL Developer : project\_sgbdb

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - grupa251\_test
  - grupa252
  - lab
  - mateinfo
  - project
  - project\_sgbdb
    - Tables (Filtered)
      - AUTOR
      - AUTOR\_CARTE
      - CARTE
      - CLIENT
      - COMANDA
      - COMANDA\_CARTE
      - COMANDA\_FILM
      - FILM
      - INCHIRIERE
      - INCHIRIERE\_CARTE
      - INCHIRIERE\_COPY
      - INCHIRIERE\_FILM
      - TABELE\_STERSE
    - Views
    - Indexes
    - Packages
      - PACK\_EX13
      - PACK\_EX14
    - Procedures
    - Functions
    - Operators
    - Queues

Worksheet Query Builder

```

EXEC proc_ex9('America');

-- Pentru cazul in care nu exista tara
EXEC proc_ex9('Americ');

-- Pentru cazul in care sunt mai multi clienti cu acelasi maxim
-- INSERT INTO INCHIRIERE_CARTE (ID_INCHIRIERE, ID_CARTE)
-- VALUES (8, 9);
EXEC proc_ex9('America');

-- Pentru cazul in care niciun client nu a inchiriat carti din tara data ca parametru
EXEC proc_ex9('Anglia');
/

```

Script Output X Query Result X

Task completed in 0.119 seconds

Procedure PROC\_EX9 compiled

Niciun client nu a inchiriat carti din tara data ca parametru!

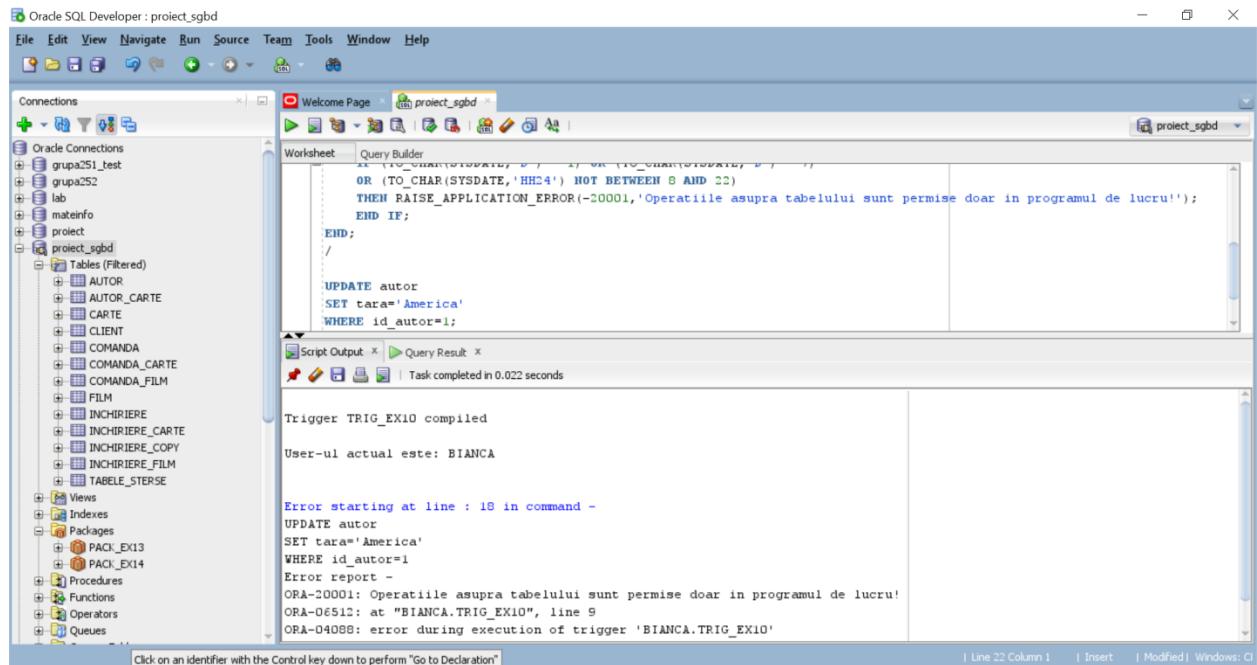
PL/SQL procedure successfully completed.

Windows Task

Line 107 Column 25 | Insert | Modified | Windows: C

**10.** Afisez user-ul și creez un program de lucru de luni până vineri între orele 8 și 22.

```
CREATE OR REPLACE TRIGGER trig_ex10
    BEFORE INSERT OR DELETE OR UPDATE ON autor
DECLARE
    v_user VARCHAR2(50);
BEGIN
    SELECT user INTO v_user FROM dual;
    DBMS_OUTPUT.PUT_LINE('User-ul actual este: ' || v_user);
    IF (TO_CHAR(SYSDATE,'D') = 1) OR (TO_CHAR(SYSDATE,'D') = 7)
        OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 22)
    THEN RAISE_APPLICATION_ERROR(-20001,'Operatiile asupra tabelului sunt
permise doar in programul de lucru!');
    END IF;
END;
/
UPDATE autor
SET tara='America'
WHERE id_autor=1;
```



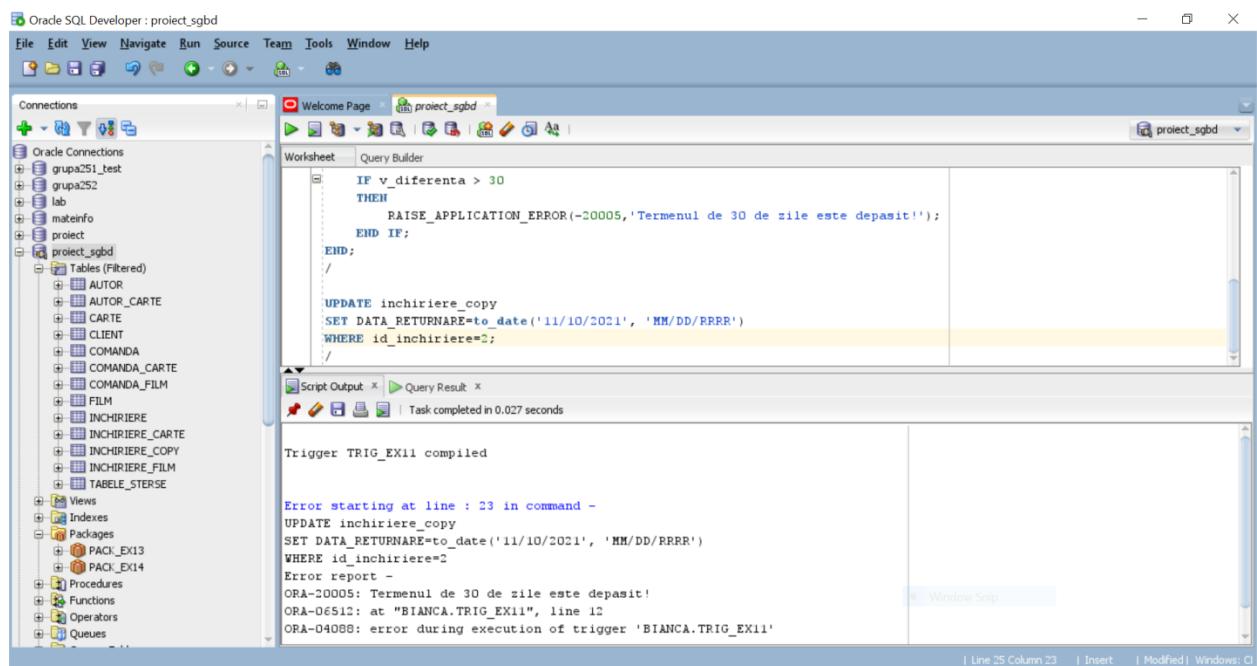
**11.** În cazul în care se face update pe data returnării, verific ca aceasta să nu depășească termenul de 30 de zile.

```
CREATE OR REPLACE TRIGGER trig_ex11
    BEFORE UPDATE OF data_returnare ON inchiriere_copy
    FOR EACH ROW
DECLARE
    v_diferenta NUMBER;
BEGIN
```

```

SELECT :NEW.data_returnare-:OLD.data_inchiriere
INTO v_diferenta
FROM inchiriere
WHERE id_inchiriere =:OLD.id_inchiriere;
IF v_diferenta > 30
THEN
    RAISE_APPLICATION_ERROR(-20005,'Termenul de 30 de zile este
depasit!');
END IF;
END;
/
UPDATE inchiriere_copy
SET DATA_RETURNARE=to_date('11/10/2021', 'MM/DD/YYYY')
WHERE id_inchiriere=2;
/

```



**12.** Acest trigger atenționează faptul că o tabelă a fost ștearsă și salvează în tabela `tabele_sterse` date despre aceasta.

```

CREATE TABLE tabele_sterse
(nume_bd VARCHAR2(50),
 user_logat VARCHAR2(30),
 nume_object_referit VARCHAR2(30),
 data TIMESTAMP(3));

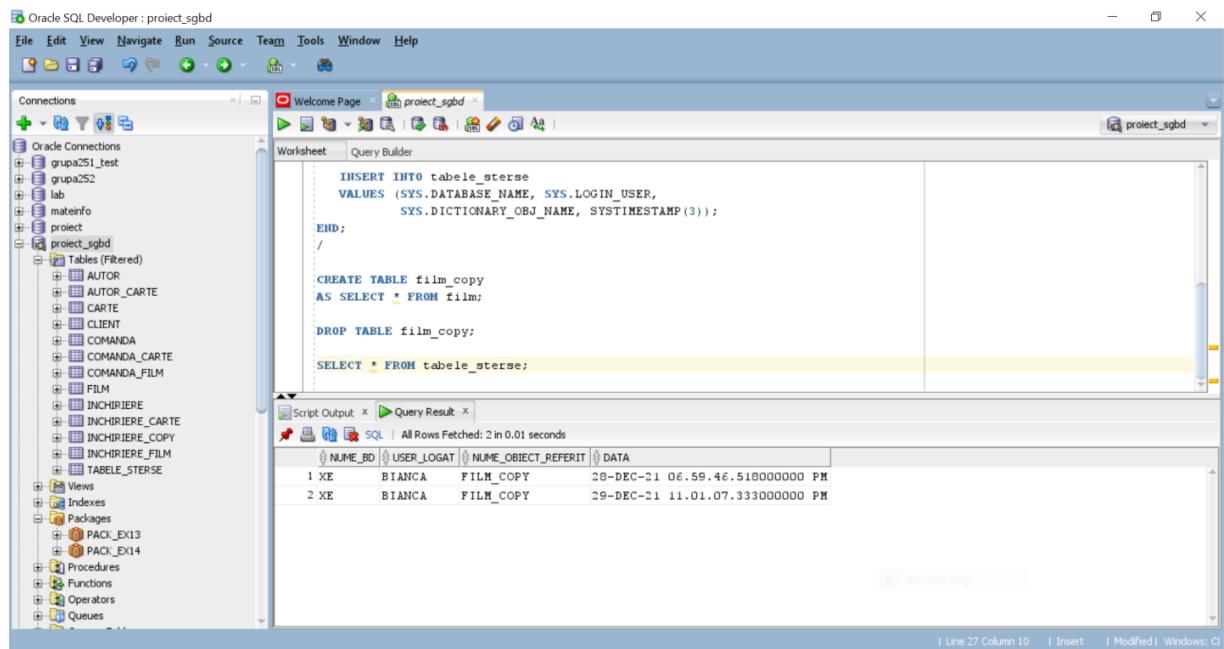
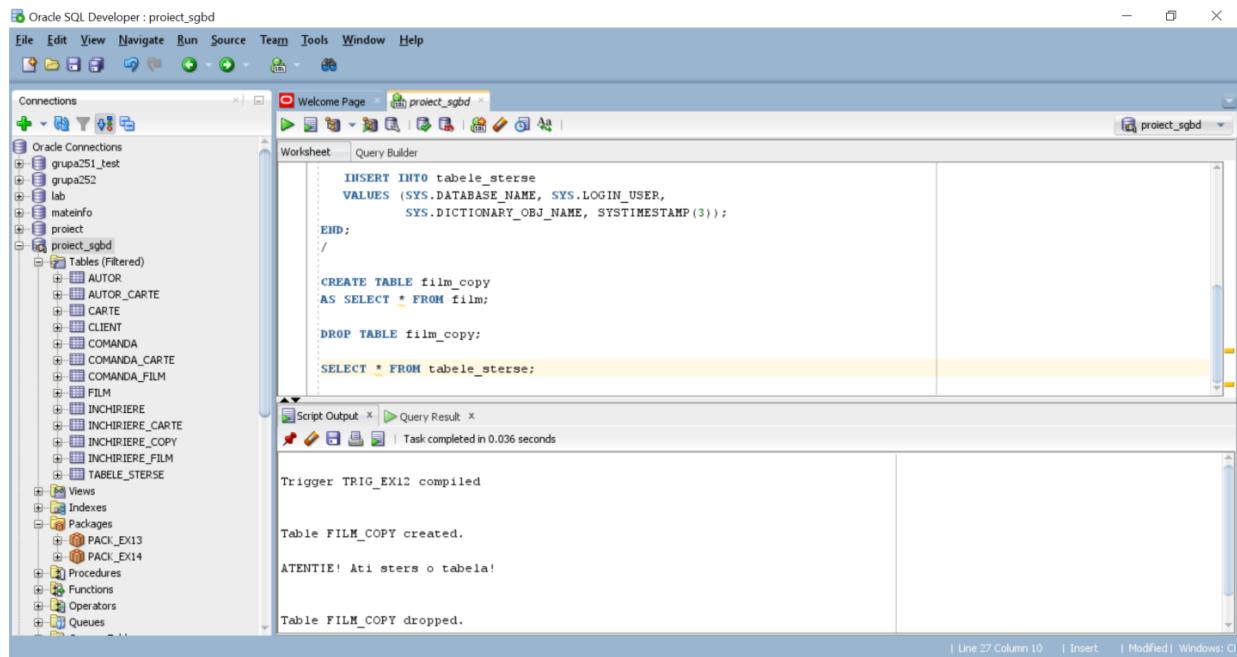
CREATE OR REPLACE TRIGGER trig_ex12
BEFORE DROP ON SCHEMA
BEGIN
    DBMS_OUTPUT.PUT_LINE('ATENTIE! Ati sters o tabela!');

```

```

INSERT INTO tabela_sterse
VALUES (SYS DATABASE_NAME, SYS LOGIN_USER,
        SYS DICTIONARY_OBJ_NAME, SYSTIMESTAMP(3));
END;
CREATE TABLE film_copy
AS SELECT * FROM film;
DROP TABLE film_copy;
SELECT * FROM tabela_sterse;

```



### 13.

```
CREATE OR REPLACE PACKAGE pack_ex13 AS
  PROCEDURE proc_ex6;
  FUNCTION func_ex7
    RETURN carte.an_aparitie%TYPE;
  FUNCTION func_ex8(p_id IN comanda.id_client%TYPE)
    RETURN NUMBER;
  PROCEDURE proc_ex9(p_tara IN autor.tara%TYPE);

END pack_ex13;
/

CREATE OR REPLACE PACKAGE BODY pack_ex13 AS

  PROCEDURE proc_ex6
  IS
    TYPE rec IS RECORD (a_prenume autor.prenume%TYPE, a_nume autor.nume%TYPE,
    a_tara autor.tara%TYPE);
    TYPE tab_ind IS TABLE OF rec INDEX BY PLS_INTEGER;
    t tab_ind;

    TYPE tab_vec IS VARRAY(50) OF NUMBER;
    tv tab_vec := tab_vec();
    v_cnt NUMBER:= 0;
    v_id NUMBER:= 0;
    v_min NUMBER;
    v_max NUMBER;
    v_titlu film.titlu%TYPE;
  BEGIN

    SELECT prenume, nume, tara BULK COLLECT INTO t
    FROM autor
    WHERE LOWER(tara) = 'anglia';

    DBMS_OUTPUT.PUT_LINE('Numarul de autori care provin din Anglia: ' ||
    t.COUNT );
    DBMS_OUTPUT.NEW_LINE;

    FOR i IN t.FIRST..t.LAST LOOP
      DBMS_OUTPUT.PUT_LINE('Autorul ' || t(i).a_prenume || ' ' || t(i).a_nume
      || ' ' || 'provine din tara: ' || t(i).a_tara);
    END LOOP;

    DBMS_OUTPUT.NEW_LINE;

    SELECT MIN(id_film) INTO v_min
    FROM film;

    SELECT MAX(id_film) INTO v_max
    FROM film;
```

```

FOR i IN v_min..v_max LOOP
  v_id:=0;

  SELECT id_film INTO v_id
  FROM film
  WHERE id_film = i;

  IF v_id!=0 THEN
    v_cnt:= v_cnt+1;
    tv.EXTEND;
    tv(v_cnt):=i;
  END IF;

  END LOOP;

  DBMS_OUTPUT.PUT_LINE('Numarul filmelor: ' || tv.COUNT);
  DBMS_OUTPUT.NEW_LINE;

  FOR i IN tv.FIRST..tv.LAST LOOP
    SELECT titlu INTO v_titlu
    FROM film
    WHERE id_film = i;
    DBMS_OUTPUT.PUT_LINE(tv(i) || ' ' || v_titlu);
  END LOOP;

END;

FUNCTION func_ex7
  RETURN carte.an_aparitie%TYPE IS
  v_min carte.an_aparitie%TYPE:= 3000;
  v_an carte.an_aparitie%TYPE;
  CURSOR crs IS
    SELECT *
    FROM film
    WHERE LOWER(este_carte)= 'da';
BEGIN

  FOR i in crs LOOP
    SELECT an_aparitie INTO v_an
    FROM carte
    WHERE LOWER(titlu) = LOWER(i.titlu);

    IF v_an < v_min THEN
      v_min := v_an;
    END IF;

  END LOOP;

  RETURN v_min;
END;

```

```

FUNCTION func_ex8(p_id IN comanda.id_client%TYPE)
    RETURN NUMBER
IS
v_suma NUMBER:= 0;
v_id_client NUMBER;
v_select NUMBER;
v_id_com NUMBER;
v_reducere NUMBER;

CURSOR carti IS (SELECT *
FROM carte c JOIN comanda_carte cc ON c.id_carte=cc.id_carte
JOIN comanda com ON cc.id_comanda=com.id_comanda
WHERE com.id_client = p_id);

CURSOR filme IS (SELECT *
FROM film f JOIN comanda_film cf ON f.id_film=cf.id_film
JOIN comanda com ON cf.id_comanda=com.id_comanda
WHERE com.id_client = p_id);

BEGIN

v_select:= 1;
SELECT id_client INTO v_id_client
FROM client
WHERE id_client = p_id;

v_select:= 2;
SELECT id_comanda INTO v_id_com
FROM comanda
WHERE id_client = p_id AND ROWNUM = 1;

v_select:= 3;
SELECT r.valoare INTO v_reducere
FROM comanda c JOIN reducere r ON c.id_reducere=r.id_reducere
WHERE c.id_comanda = v_id_com;

FOR i IN carti LOOP
v_suma:= i.pret + v_suma;
END LOOP;

FOR j IN filme LOOP
v_suma:= j.pret + v_suma;
END LOOP;

v_suma:= v_suma - v_reducere*v_suma;

RETURN v_suma;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        IF v_select=1 THEN
            RAISE_APPLICATION_ERROR(-20000, 'Nu exista id-ul clientului!');
        ELSE

```

```

        RAISE_APPLICATION_ERROR(-20000, 'Clientul nu a efectuat nicio
comanda!');
      END IF;
END;

PROCEDURE proc_ex9(p_tara IN autor.tara%TYPE)
IS
v_nume autor.nume%TYPE;
v_prenume autor.prenume%TYPE;
v_max NUMBER;
v_cnt NUMBER:= 0;
v_id_client client.id_client%TYPE;
v_count NUMBER:= 0;
v_id_final client.id_client%TYPE;
v_tara autor.tara%TYPE;
v_ok NUMBER:=1;

CURSOR crs IS (SELECT client.id_client, COUNT(ca.id_carte)
FROM client
JOIN inchiriere i ON client.id_client=i.id_client JOIN
inchiriere_carte ic ON ic.id_inchiriere=i.id_inchiriere JOIN
carte ca ON ic.id_carte=ca.id_carte JOIN autor_carte ac ON
ac.id_carte=ca.id_carte JOIN autor a ON a.id_autor=ac.id_autor
WHERE LOWER(a.tara)=LOWER(p_tara)
GROUP BY client.id_client);

CURSOR inchirieri IS (SELECT id_inchiriere FROM inchiriere);

CURSOR tari IS (SELECT tara
                  FROM autor
                  WHERE LOWER(tara) = LOWER(p_tara) AND ROWNUM = 1);

exceptie EXCEPTION;

BEGIN

OPEN tari;
FETCH tari INTO v_tara;

IF tari%ROWCOUNT = 0 THEN
  RAISE NO_DATA_FOUND;
END IF;
CLOSE tari;

OPEN inchirieri;
IF inchirieri%NOTFOUND THEN
  RAISE exceptie;
END IF;

CLOSE inchirieri;

SELECT MAX(COUNT(ca.id_carte)) INTO v_max

```

```

FROM inchiriere i JOIN inchiriere_carte ic ON
ic.id_inchiriere=i.id_inchiriere
JOIN carte ca ON ic.id_carte=ca.id_carte JOIN autor_carte ac ON
ac.id_carte=ca.id_carte JOIN autor a ON a.id_autor=ac.id_autor
WHERE LOWER(a.tara)=lower(p_tara)
GROUP BY ca.id_carte;

OPEN crs;
LOOP
FETCH crs INTO v_id_client, v_count;
EXIT WHEN crs%NOTFOUND;

IF v_count=v_max THEN
v_id_final:=v_id_client;
v_cnt:= v_cnt +1;
END IF;
END LOOP;
IF v_cnt>1 THEN
RAISE TOO_MANY_ROWS;
END IF;

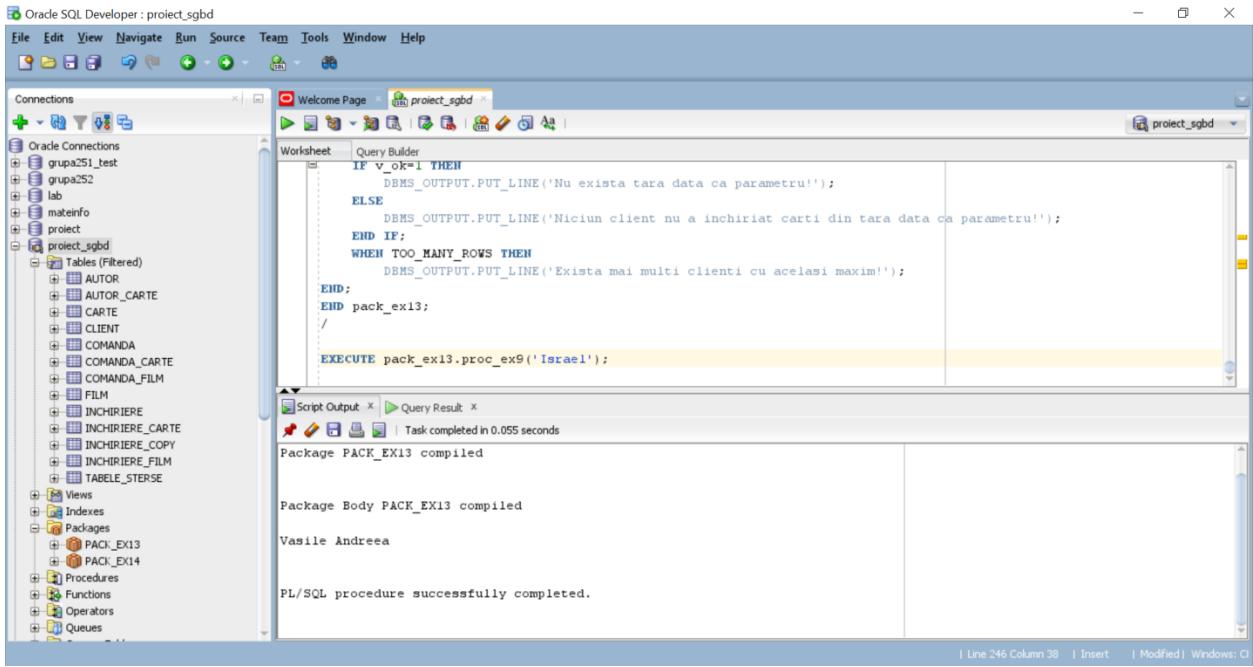
v_ok:=0;

SELECT nume, prenume INTO v_nume, v_prenume
FROM client
WHERE id_client = v_id_final;

DBMS_OUTPUT.PUT_LINE(v_nume || ' ' || v_prenume);
CLOSE crs;
EXCEPTION
WHEN exceptie THEN
    DBMS_OUTPUT.PUT_LINE('Nu au fost efectuate inchirieri!');
WHEN NO_DATA_FOUND THEN
    IF v_ok=1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista tara data ca parametru!');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Niciun client nu a inchiriat carti din tara
data ca parametru!');
    END IF;
WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Există mai mulți clienti cu același
maxim!');

END;
END pack_ex13;
/
EXECUTE pack_ex13.proc_ex9('Israel');
/

```



#### 14. Pachetul conține:

- o funcție care să calculeze suma prețurilor filmelor închiriate de un client dat de la tastatură
- o funcție care să calculeze suma prețurilor cărților închiriate de un client dat de la tastatură
- o procedură prin care afișez suma totală a celor două funcții
- o procedură care afișează titlul și anul apariției filmelor cele mai comandate, cu ajutorul unui vector
- o procedură care afișează titlurile cărților care au fost împrumutate măcar o data, cu ajutorul unui tablou indexat

```

CREATE OR REPLACE PACKAGE pack_ex14 AS
FUNCTION func_filme(p_id IN inchiriere.id_client%TYPE)
    RETURN NUMBER;
FUNCTION func_carte(p_id IN inchiriere.id_client%TYPE)
    RETURN NUMBER;
PROCEDURE suma_inchiriere(p_id IN inchiriere.id_client%TYPE);
PROCEDURE filme_comandate;
PROCEDURE carti_inchiriate;
TYPE tab_vec IS VARRAY(50) OF NUMBER;
TYPE tab_ind IS TABLE OF NUMBER INDEX BY PLS_INTEGER;

END pack_ex14;
/

CREATE OR REPLACE PACKAGE BODY pack_ex14 AS
FUNCTION func_filme(p_id IN inchiriere.id_client%TYPE)
    RETURN NUMBER

```

```

IS
v_suma NUMBER:= 0;
v_id_client NUMBER;
v_id_in NUMBER;

CURSOR filme IS (SELECT *
FROM film f JOIN inchiriere_film if ON f.id_film=if.id_film
JOIN inchiriere i ON if.id_inchiriere=i.id_inchiriere
WHERE i.id_client = p_id);

BEGIN

SELECT id_client INTO v_id_client
FROM client
WHERE id_client = p_id;

FOR i IN filme LOOP
v_suma:= i.pret/4 + v_suma;
END LOOP;

RETURN v_suma;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20000, 'Nu exista id-ul clientului!');
END;

FUNCTION func_carti(p_id IN inchiriere.id_client%TYPE)
RETURN NUMBER
IS
v_suma NUMBER:= 0;
v_id_client NUMBER;
v_id_in NUMBER;

CURSOR carti IS (SELECT *
FROM carte c JOIN inchiriere_carte ic ON c.id_carte=ic.id_carte
JOIN inchiriere i ON ic.id_inchiriere=i.id_inchiriere
WHERE i.id_client = p_id);

BEGIN

SELECT id_client INTO v_id_client
FROM client
WHERE id_client = p_id;

FOR i IN carti LOOP
v_suma:= i.pret/4 + v_suma;
END LOOP;

RETURN v_suma;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RAISE_APPLICATION_ERROR(-20000, 'Nu exista id-ul clientului!');
END;

```

```

PROCEDURE suma_inchiriere(p_id IN inchiriere.id_client%TYPE)
IS
v_suma_filme NUMBER:=0;
v_suma_carti NUMBER:=0;
v_suma NUMBER:=0;

v_nume client.nume%TYPE;
v_prenume client.prenume%TYPE;

BEGIN
v_suma_filme:=func_filme(p_id);
v_suma_carti:=func_carti(p_id);

SELECT nume INTO v_nume
FROM client
WHERE id_client=p_id;

SELECT prenume INTO v_prenume
FROM client
WHERE id_client=p_id;

v_suma:=v_suma_filme + v_suma_carti;

DBMS_OUTPUT.PUT_LINE('Clientul ' || v_nume || ' ' || v_prenume || ' a
platit in total pentru inchirieri: '
|| v_suma);
END;

PROCEDURE filme_comandate
IS

CURSOR filme_com IS
(
    SELECT f.id_film, COUNT(f.id_film)
    FROM film f JOIN comanda_film cf ON f.id_film=cf.id_film
    GROUP BY f.id_film
);

vec tab_vec := tab_vec();
v_max NUMBER;
v_id_film film.id_film%TYPE;
v_cnt NUMBER;
v_i NUMBER:=1;
v_titlu film.titlu%TYPE;
v_an film.an_aparitie%TYPE;
BEGIN
    SELECT MAX(COUNT(f.id_film)) INTO v_max
    FROM film f JOIN comanda_film cf ON f.id_film=cf.id_film
    GROUP BY f.id_film;

OPEN filme_com;

LOOP
    FETCH filme_com INTO v_id_film, v_cnt;

```

```

        EXIT WHEN filme_com%NOTFOUND;

        IF v_cnt=v_max THEN
        vec.EXTEND;
        vec(v_i):=v_id_film;
        v_i:= v_i+1;
        END IF;

    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Cele mai comandate filme: ');
    FOR i IN vec.FIRST..vec.LAST LOOP
        SELECT titlu INTO v_titlu
        FROM film
        WHERE id_film=vec(i);

        SELECT an_aparitie INTO v_an
        FROM film
        WHERE id_film=vec(i);

        DBMS_OUTPUT.PUT_LINE(v_titlu || ' ' || v_an);

    END LOOP;
    CLOSE filme_com;
    END;

PROCEDURE carti_inchiriate
IS
CURSOR carti_inc IS
(
    SELECT DISTINCT c.id_carte
    FROM carte c JOIN inchiriere_carte ic ON c.id_carte=ic.id_carte
);
t tab_ind;
v_id_carte carte.id_carte%TYPE;
v_cnt NUMBER:=0;
v_i NUMBER:=1;
v_titlu carte.titlu%TYPE;

BEGIN
OPEN carti_inc;

    LOOP
        FETCH carti_inc INTO v_id_carte;
        EXIT WHEN carti_inc%NOTFOUND;

        t(v_i):=v_id_carte;
        v_i:= v_i+1;

    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Carti inchiriate: ');
    FOR i IN t.FIRST..t.LAST LOOP
        SELECT titlu INTO v_titlu

```

```

    FROM carte
    WHERE id_carte=t(i);

    DBMS_OUTPUT.PUT_LINE(v_titlu);
END LOOP;
CLOSE carti_inc;
END;

END pack_ex14;
/

EXECUTE pack_ex14.carti_inchiriate;
EXECUTE pack_ex14.filme_comandate;
EXECUTE pack_ex14.suma_inchiriere(1);
/

```

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree displays several database connections, with 'project\_sgbdb' selected. The central workspace contains a 'Worksheet' tab with the following PL/SQL code:

```

BEGIN pack_ex14;
/

EXECUTE pack_ex14.carti_inchiriate;
EXECUTE pack_ex14.filme_comandate;
EXECUTE pack_ex14.suma_inchiriere(1);

```

Below the worksheet is a 'Script Output' tab showing the results of the execution:

```

Package Body PACK_EX14 compiled

Carti inchiriate:
Don Quijote
Trei metri deasupra cerului
Delirium
Jack Spintecatorul. Crimele din Whitechapel
Mandrie si prejudecata
Sabrina, intre lumina si intuneric
Deodata in adancul padurii
Pe ariile vantului
Anna Karenina
Fata cu vise alb-negru
Inteligenta emotionala
Distrugeti obiceiurile nocive!

```

The status bar at the bottom indicates 'Line 217 Column 38 | Insert | Modified | Windows: C'.

Oracle SQL Developer : project\_sgbd

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
  - grupa2S1\_test
  - grupa2S2
  - lab
  - mateinfo
  - project
  - project\_sgbd
    - Tables (Filtered)
      - AUTOR
      - AUTOR\_CARTE
      - CARTE
      - CLIENT
      - COMANDA
      - COMANDA\_CARTE
      - COMANDA\_FILM
      - FILM
      - INCHIRIERE
      - INCHIRIERE\_CARTE
      - INCHIRIERE\_COPY
      - INCHIRIERE\_FILM
      - TABLELE\_STERSE
    - Views
    - Indexes
    - Packages
      - PACI\_EX13
      - PACI\_EX14
    - Procedures
    - Functions
    - Operators
    - Queues

Welcome Page : project\_sgbd

Worksheet Query Builder

```
EHID pack_ex14;
/
EXECUTE pack_ex14.carti_inchiriate;
EXECUTE pack_ex14.filme_comandate;
EXECUTE pack_ex14.suma_inchiriere(1);
/
```

Script Output X | Query Result X

Cele mai comandate filme:  
Crima din Orient Express 2017  
Zecă negri mititei 1965

PL/SQL procedure successfully completed.

Clientul Sebe Lavinia a platit in total pentru inchirieri: 46

PL/SQL procedure successfully completed.

| Line 217 Column 38 | Insert | Modified | Windows: Cl