

Proiect

Gestiunea unui magazin de închiriere filme

Rotaru Teodor Gabriel

Contents

Descrierea bazei de date	3
Schema bazei de date.....	4
Capitolul 1 - SQL	5
Capitolul 2 - PL/SQL	18

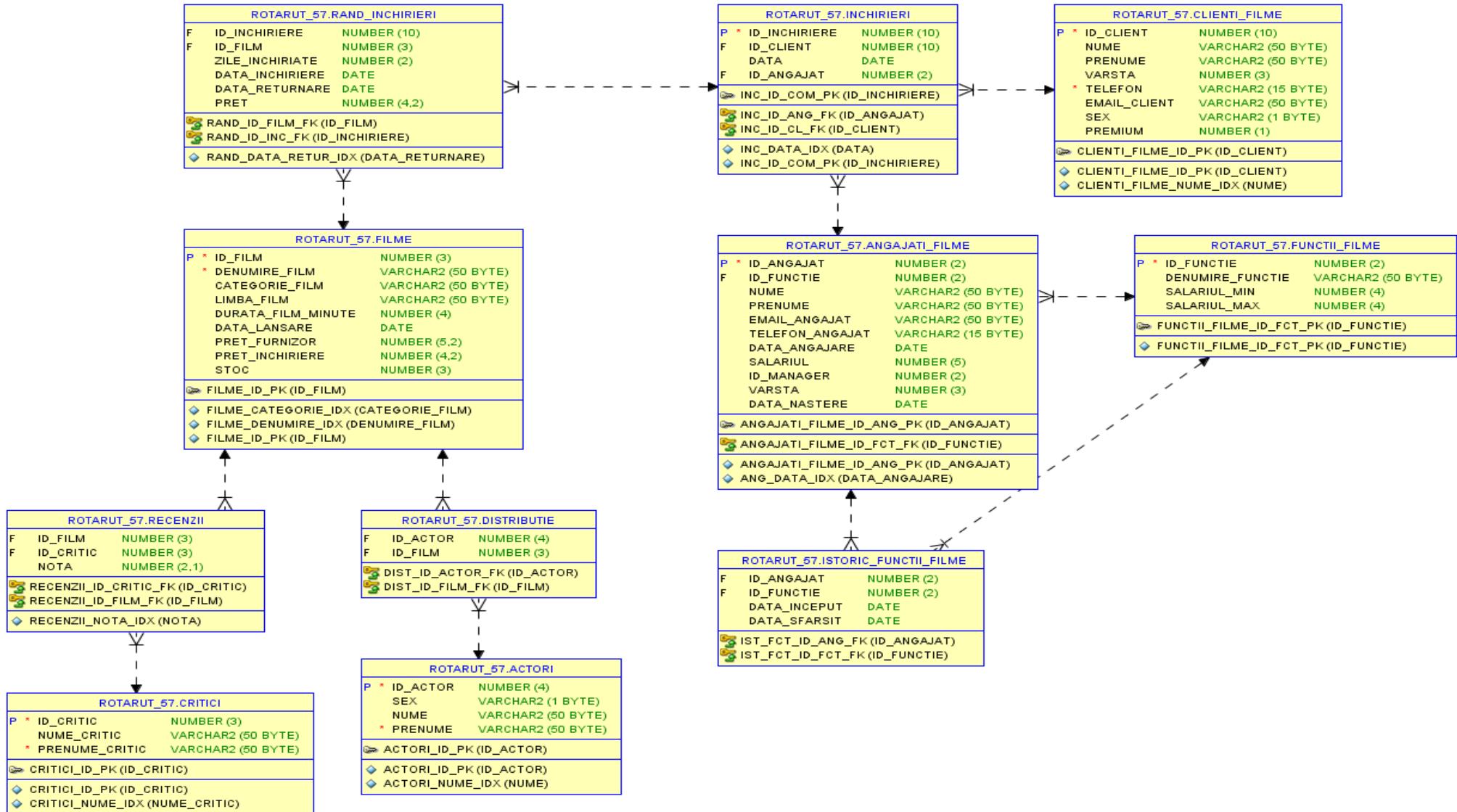
Descrierea bazei de date

Am ales să creez o bază de date pentru gestiunea unui magazin fizic de închiriere a filmelor. Aceasta conține informații generale despre angajații magazinului, despre filmele disponibile în magazin, despre actorii din distribuția acestora, și despre criticii de film ce le-au clasificat cu o notă de la 1 la 10. De asemenea, baza de date a proiectului păstrează evidența clientilor ce au închiriat filme și a comenzi acestora, aşadar corespunde nevoilor organizatorice și de gestiune ale unui magazin fizic.

O bună parte din exerciții au o temă statistică cu scopul dobândirii unor cunoștințe de bază în legătură cu dorințele clientilor, dar și cu funcționarea magazinului într-un mod cât mai profitabil. Astfel, se regăsesc instrucțiuni SQL pentru a descoperi cea mai populară categorie de film, valoarea medie cheltuită de clienti per comandă, media vârstelor clientilor ce au un anumit număr de comenzi, vârstele clientilor ce au închiriat filme dintr-o anumită categorie, număr de clienti pentru fiecare luna, etc. De asemenea, se regăsesc instrucțiuni SQL pentru modificarea bazei de date conform posibilelor nevoi ale unui astfel de magazin sau de afișare a diverselor informații relevante pentru angajați sau pentru întreprinzătorul afacerii.

- Tabela „FILME” – are ca primary key coloana ID_FILM. Prin aceasta coloana, tabela este în legătură cu tabelele „RAND_INCHIRIERI”, „RECENZII” și „DISTRIBUTIE”. Restul coloanelor contin informații generale despre filme, precum denumirea acestuia, categoria, dar și despre pretul de achiziție de la furnizor, pretul cu care se închiriază și stoc-ul disponibil.
- Tabela „INCHIRIERI” are ca primary key coloana „ID_INCHIRIERE”. Este în legătură cu tabelele „CLIENTI” și „ANGAJATI” prin coloanele „ID_CLIENT” și „ID_ANGAJAT”.
- Tabela RAND_INCHIRIERI conține liniile închirierii din tabela INCHIRIERI. Aceasta este în legătură cu tabela INCHIRIERI prin coloana ID_INCHIRIERE și conține id-ul filmului închiriat, numarul de zile pentru care a fost închiriat, data la care a fost închiriat, precum și cea la care trebuie returnat și pretul care reprezintă PRET_INCHIRIERE*ZILE_INCHIRIATE.
- Tabela CLIENTI_FILME conține informații generale despre oamenii care au închiriat filme în magazin. Aceasta se află în legătură doar cu tabela INCHIRIERI.
- Tabela ANGAJATI_FILME conține informații despre angajați și se află în legătură cu tabela FUNCTII_FILME prin coloana ID_FUNCTIE, cu tabela ISTORIC_FUNCTII_FILME prin coloana ID_ANGAJAT și cu tabela INCHIRIERI.
- Tabelele RECENZII și DISTRIBUTIE se află în legătură cu tabela FILME prin ID_FILM.
- Tabela RECENZII se află în legătură cu tabela CRITICI prin ID_CRITIC.
- Tabela DISTRIBUTIE se află în legătură cu tabela ACTORI prin ID_ACTOR.

Schema bazei de date



Capitolul 1 - SQL

```
DROP TABLE FILME CASCADE CONSTRAINTS;
DROP TABLE RAND_INCHIRIERI CASCADE CONSTRAINTS;
DROP TABLE INCHIRIERI CASCADE CONSTRAINTS;
DROP TABLE CRITICI CASCADE CONSTRAINTS;
DROP TABLE RECENZII CASCADE CONSTRAINTS;
DROP TABLE ACTORI CASCADE CONSTRAINTS;
DROP TABLE DISTRIBUTIE CASCADE CONSTRAINTS;
DROP TABLE CLIENTI CASCADE CONSTRAINTS;
DROP TABLE ANGAJATI CASCADE CONSTRAINTS;
DROP TABLE FUNCTII CASCADE CONSTRAINTS;
DROP TABLE ISTORIC_FUNCTII CASCADE CONSTRAINTS;
DROP TABLE CLIENTI_ELIGIBILI_PENTRU_UN_FILM_GRATIS CASCADE CONSTRAINTS;
DROP SEQUENCE seq_idangajat;
DROP SYNONYM CEPUFG;
DROP SYNONYM RI;
DROP SYNONYM INC;

--SA SE CREEZE TABELA FILME

CREATE TABLE FILME (
    ID_FILM NUMBER(3),
    DENUMIRE_FILM VARCHAR2(50),
    CATEGORIE_FILM VARCHAR2(50),
    LIMBA_FILM VARCHAR2(50),
    DURATA_FILM_MINUTE NUMBER(4),
    DATA_LANSARE DATE,
    PRET_FURNIZOR NUMBER(5,2),
    PRET_INCHIRIERE NUMBER(4,2)
);
```

```
--SA SE ADAUGE COLOANA STOC NUMBER(3)
```

```
ALTER TABLE FILME ADD STOC NUMBER(3);
```

```
--SA SE CREEZE INDECSI PENTRU COLOANELE DENUMIRE_FILM SI CATEGORIE_FILM
```

```
CREATE INDEX FILME_DENUMIRE_IDX ON FILME (DENUMIRE_FILM);
```

```
CREATE INDEX FILME_CATEGORIE_IDX ON FILME (CATEGORIE_FILM);
```

```
--RESTRICTII
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_ID_PK PRIMARY KEY (ID_FILM);
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_DENUMIRE_NN CHECK ("DENUMIRE_FILM" IS NOT NULL);
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_DATA_LANS_NN CHECK ("DATA_LANSARE" IS NOT NULL);
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_PRET_FUR_CK CHECK (PRET_FURNIZOR > 0);
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_PRET_INC_CK CHECK (PRET_INCHIRIERE > 0);
```

```
ALTER TABLE FILME ADD CONSTRAINT FILME_STOC_CK CHECK (STOC > 0);
```

```
--SA SE STEARGA RESTRICTIA FILME_DATA_LANS_NN
```

```
ALTER TABLE FILME DROP CONSTRAINT FILME_DATA_LANS_NN;
```

```
CREATE TABLE CRITICI (
```

```
    ID_CRITIC NUMBER(3),
```

```
    NUME_CRITIC VARCHAR2(50),
```

```
    PRENUME_CRITIC VARCHAR2(50) NOT NULL
```

```
);
```

```
--SA SE CREEZE UN INDEX PENTRU COLOANA NUME_CRITIC
```

```
CREATE INDEX CRITICI_NUME_IDX ON CRITICI (NUME_CRITIC);
```

```

--SA SE ADAUGE RESTRICTIA DE PRIMARY KEY COLOANEI ID_CRITIC

ALTER TABLE CRITICI ADD CONSTRAINT CRITICI_ID_PK PRIMARY KEY (ID_CRITIC);

CREATE TABLE RECENZII (
    ID_FILM NUMBER(3),
    ID_CRITIC NUMBER(3),
    NOTA NUMBER(2,1)
);

--RESTRICTII

ALTER TABLE RECENZII ADD CONSTRAINT RECENZII_ID_FILM_FK FOREIGN KEY (ID_FILM) REFERENCES FILME (ID_FILM);
ALTER TABLE RECENZII ADD CONSTRAINT RECENZII_ID_CRITIC_FK FOREIGN KEY (ID_CRITIC) REFERENCES CRITICI (ID_CRITIC);

--SA SE CREEZE UN INDEX PENTRU COLOANA NOTA

CREATE INDEX RECENZII_NOTA_IDX ON RECENZII (NOTA);

CREATE TABLE ACTORI (
    ID_ACTOR NUMBER(4) ,
    SEX VARCHAR2(1),
    NUME VARCHAR2(50),
    PRENUME VARCHAR2(50) NOT NULL
);

ALTER TABLE ACTORI ADD CONSTRAINT ACTORI_ID_PK PRIMARY KEY (ID_ACTOR);
CREATE INDEX ACTORI_NUME_IDX ON ACTORI (NUME);
ALTER TABLE ACTORI ADD CONSTRAINT ACTORI_SEX_CK CHECK (SEX IN ('M', 'F'));

```

```

CREATE TABLE DISTRIBUTIE (
    ID_ACTOR NUMBER(4),
    ID_FILM NUMBER(3)
);

ALTER TABLE DISTRIBUTIE ADD CONSTRAINT DIST_ID_ACTOR_FK FOREIGN KEY (ID_ACTOR) REFERENCES ACTORI (ID_ACTOR);
ALTER TABLE DISTRIBUTIE ADD CONSTRAINT DIST_ID_FILM_FK FOREIGN KEY (ID_FILM) REFERENCES FILME (ID_FILM);

CREATE TABLE CLIENTI (
    ID_CLIENT NUMBER(10),
    NUME VARCHAR2(50),
    PRENUME VARCHAR2(50),
    VARSTA NUMBER(3),
    TELEFON VARCHAR2(15),
    EMAIL_CLIENT VARCHAR2(50),
    SEX VARCHAR2(1),
    PREMIUM NUMBER(1)
);

ALTER TABLE CLIENTI ADD CONSTRAINT CLIENTI_ID_PK PRIMARY KEY (ID_CLIENT);
ALTER TABLE CLIENTI ADD CONSTRAINT CLIENTI_SEX_CK CHECK (SEX IN ('M', 'F'));
ALTER TABLE CLIENTI ADD CONSTRAINT CLIENTI_TEL_NN CHECK ("TELEFON" IS NOT NULL);
ALTER TABLE CLIENTI ADD CONSTRAINT CLIENTI_PREMIUM_CK CHECK (PREMIUM IN (0, 1));
CREATE INDEX CLIENTI_NUME_IDX ON CLIENTI (NUME);

CREATE TABLE FUNCTII (
    ID_FUNCTIE NUMBER(2),
    DENUMIRE_FUNCTIE VARCHAR2(50),
    SALARIUL_MIN NUMBER(4),
    SALARIUL_MAX NUMBER(4)
);

```

```

ALTER TABLE FUNCTII ADD CONSTRAINT FUNCTII_ID_FCT_PK PRIMARY KEY (ID_FUNCTIE);
ALTER TABLE FUNCTII ADD CONSTRAINT FUNCTII_SAL_MIN_CK CHECK (SALARIUL_MIN > 0);
ALTER TABLE FUNCTII ADD CONSTRAINT FUNCTII_SAL_MAX_CK CHECK (SALARIUL_MAX >= SALARIUL_MIN);

CREATE TABLE ANGAJATI (
    ID_ANGAJAT NUMBER(2),
    ID_FUNCTIE NUMBER(2),
    NUME VARCHAR2(50),
    PRENUME VARCHAR2(50),
    EMAIL_ANGAJAT VARCHAR2(50),
    TELEFON_ANGAJAT VARCHAR2(15),
    DATA_ANGAJARE DATE,
    SALARIUL NUMBER(5),
    ID_MANAGER NUMBER(2),
    VARSTA NUMBER(3)
);

ALTER TABLE ANGAJATI ADD DATA_NASTERE DATE;
CREATE INDEX ANG_DATA_IDX ON ANGAJATI(DATA_ANGAJARE);
ALTER TABLE ANGAJATI ADD CONSTRAINT ANGAJATI_ID_ANG_PK PRIMARY KEY (ID_ANGAJAT);
ALTER TABLE ANGAJATI ADD CONSTRAINT ANGAJATI_ID_FCT_FK FOREIGN KEY (ID_FUNCTIE) REFERENCES FUNCTII (ID_FUNCTIE);
ALTER TABLE ANGAJATI ADD CONSTRAINT ANGAJATI_VARSTA_CK CHECK (VARSTA >= 16);

CREATE TABLE ISTORIC_FUNCTII (
    ID_ANGAJAT NUMBER(2),
    ID_FUNCTIE NUMBER(2),
    DATA_INCEPUT DATE,
    DATA_SFARSIT DATE
);

ALTER TABLE ISTORIC_FUNCTII ADD CONSTRAINT IST_FCT_ID_ANG_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI (ID_ANGAJAT);
ALTER TABLE ISTORIC_FUNCTII ADD CONSTRAINT IST_FCT_ID_FCT_FK FOREIGN KEY (ID_FUNCTIE) REFERENCES FUNCTII (ID_FUNCTIE);

```

```

CREATE TABLE INCHIRIERI (
    ID_INCHIRIERE NUMBER(10),
    ID_CLIENT NUMBER(10),
    DATA DATE,
    ID_ANGAJAT NUMBER(2)
);

ALTER TABLE INCHIRIERI ADD CONSTRAINT INC_ID_COM_PK PRIMARY KEY (ID_INCHIRIERE);
ALTER TABLE INCHIRIERI ADD CONSTRAINT INC_ID_CL_FK FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI (ID_CLIENT);
ALTER TABLE INCHIRIERI ADD CONSTRAINT INC_ID_ANG_FK FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI (ID_ANGAJAT);
CREATE INDEX INC_DATA_IDX ON INCHIRIERI (DATA);

CREATE TABLE RAND_INCHIRIERI (
    ID_INCHIRIERE NUMBER(10),
    ID_FILM NUMBER(3),
    ZILE_INCHIRIATE NUMBER(2),
    DATA_INCHIRIERE DATE,
    DATA_RETURNARE DATE,
    PRET NUMBER(4,2)
);

ALTER TABLE RAND_INCHIRIERI ADD CONSTRAINT RAND_ID_INC_FK FOREIGN KEY (ID_INCHIRIERE) REFERENCES INCHIRIERI (ID_INCHIRIERE);
ALTER TABLE RAND_INCHIRIERI ADD CONSTRAINT RAND_ID_FILM_FK FOREIGN KEY (ID_FILM) REFERENCES FILME (ID_FILM);
ALTER TABLE RAND_INCHIRIERI ADD CONSTRAINT RAND_SAPT_CK CHECK (ZILE_INCHIRIATE > 0);
ALTER TABLE RAND_INCHIRIERI ADD CONSTRAINT RAND_PRET_CK CHECK (PRET > 0);
CREATE INDEX RAND_DATA_RETUR_IDX ON RAND_INCHIRIERI ( DATA_RETURNARE );

--SA SE CREEZE O SEVENTA PENTRU ASIGURAREA UNICITATII CHEII PRIMARE DIN TABELA ANGAJATI

CREATE SEQUENCE seq_idangajat START WITH 1 INCREMENT BY 1 MAXVALUE 14 NOCYCLE;

```

```

--SA SE SCADA CU 5% PRETUL DE INCHIRIERE AL FILMELOR CARE AU PRETUL DE LA FURNIZOR MAI MARE DE 50 LEI

UPDATE FILME SET PRET_INCHIRIERE = 0.95*PRET_INCHIRIERE WHERE PRET_FURNIZOR > 50;

--SA SE INLOCUIASCA CU "Indisponibil" MAIL-UL CLIENTILOR CE FOLOSESC YAHOO

UPDATE CLIENTI SET EMAIL_CLIENT='Indisponibil' WHERE LOWER(EMAIL_CLIENT) LIKE '%@yahoo.com';

--SA SE STEARGA TOTI ANGAJATII CARE NU AU VANDUT (INCHIRIAT) NICIU FILM, EXCEPTAND MANAGERUL

DELETE FROM ANGAJATI WHERE ID_FUNCTIE NOT IN ( SELECT ID_FUNCTIE FROM FUNCTII WHERE UPPER(DENUMIRE_FUNCTIE) = 'MANAGER')
AND ID_ANGAJAT NOT IN (SELECT ID_ANGAJAT FROM INCHIRIERI);

--SA SE AFISEZE TOTI SUBORDONATII DIRECTI AI ANGAJATULUI CU FUNCTIA DE "Supervizor magazin"

SELECT NUME, PRENUME FROM ANGAJATI WHERE ID_MANAGER IN ( SELECT ID_FUNCTIE FROM FUNCTII
WHERE LOWER(DENUMIRE_FUNCTIE)='supervizor magazin');

--SA SE CALCULEZE VARSTA MEDIE A CLIENTILOR CARE AU INCHIRIAT FILME DIN CATEGORIA "Horror"

SELECT AVG(VARSTA) AS MEDIE FROM CLIENTI WHERE ID_CLIENT IN ( SELECT ID_CLIENT FROM INCHIRIERI
JOIN RAND_INCHIRIERI ON INCHIRIERI.ID_INCHIRIERE=RAND_INCHIRIERI.ID_INCHIRIERE WHERE ID_FILM IN ( SELECT ID_FILM FROM FILME
WHERE LOWER(CATEGORIE_FILM)='horror'));
```

```
SELECT AVG(VARSTA) AS MEDIE FROM CLIENTI WHERE ID_CLIENT IN ( SELECT ID_CLIENT FROM INCHIRIERI JOIN RAND_INCHIRIERI ON INCHIRIERI.ID_INCHIRIERE=RAND_INCHIRIERI.ID_INCHIRIERE WHERE ID_FILM IN ( SELECT ID_FILM FROM FILME WHERE
LOWER(CATEGORIE_FILM)='horror'))
```

MEDIE
26.8

Statement processed. 0.07 seconds

--SA SE CALCULEZE RAPORTUL DINTRE NUMARUL DE ANGAJATI CU VARSTA SUB 18 ANI SI NUMARUL CELOR MAJORI

```
SELECT COUNT(CASE WHEN VARSTA < 18 THEN 1 END) / NULLIF( COUNT(CASE WHEN VARSTA >= 18 THEN 1 END), 0) AS RAPORT FROM ANGAJATI;
```

```
SELECT COUNT(CASE WHEN VARSTA < 18 THEN 1 END) / NULLIF( COUNT(CASE WHEN VARSTA >= 18 THEN 1 END), 0) AS RAPORT FROM ANGAJATI
```

RAPORT

2

Statement processed. 0.00 seconds

--SA SE MAREASCA SALARIUL CU 10% ANGAJATILOR CARE IMPLINESC ASTAZI 18 ANI SI SA SE MODIFICE VARSTA LOR

```
UPDATE ANGAJATI SET SALARIUL=1.10*SALARIUL, VARSTA=VARSTA+1 WHERE EXTRACT(MONTH FROM DATA_NASTERE)=EXTRACT(MONTH FROM SYSDATE) AND EXTRACT(DAY FROM DATA_NASTERE)=EXTRACT(DAY FROM SYSDATE) AND VARSTA = 17;
```

--SA SE MICSOREZE CU 15% PRETUL DE INCHIRIERE AL FILMELOR CARE NU AU FOST INCHIRIATE NICIODATA, FARĂ A TRECE DE PRAGUL DE 0.99

```
UPDATE FILME SET PRET_INCHIRIERE=CASE
WHEN 0.85*PRET_INCHIRIERE>=0.99 THEN 0.85*PRET_INCHIRIERE
ELSE 0.99 END
WHERE ID_FILM NOT IN (SELECT ID_FILM FROM RAND_INCHIRIERI);
```

--SA SE AFISEZE NUMARUL DE FILME INCHIRIATE DE LA INCEPUTUL ANGAJARII AL FIECARUI ANGAJAT IN ORDINE DESCRESCATOARE

```
SELECT ID_ANGAJAT, COUNT(*) AS FILME_INCHIRIATE FROM RAND_INCHIRIERI JOIN INCHIRIERI ON
RAND_INCHIRIERI.ID_INCHIRIERE=INCHIRIERI.ID_INCHIRIERE GROUP BY ID_ANGAJAT ORDER BY 2 DESC;
```

--SA SE MODIFICE IN CLIENTI PREMIUM CLIENTII CARE AU CEL PUTIN 3 COMENZI (INCHIRIERI)

```
UPDATE CLIENTI SET PREMIUM=1 WHERE ID_CLIENT IN (SELECT ID_CLIENT FROM INCHIRIERI GROUP BY ID_CLIENT HAVING COUNT(*)>=3);
```

--SA SE AFISEZE DENUMIREA SI CATEGORIA CELUI MAI POPULAR FILM

```
SELECT DENUMIRE_FILM, CATEGORIE_FILM FROM FILME WHERE ID_FILM IN ( SELECT STATS_MODE(ID_FILM) FROM RAND_INCHIRIERI );
```

--SA SE AFISEZE DATA LANSARII CELUI MAI POPULAR FILM IN FORMAT LUNA/AN

```
SELECT TO_CHAR(DATA_LANSARE, 'MM/YYYY') FROM FILME WHERE ID_FILM IN ( SELECT STATS_MODE(ID_FILM) FROM RAND_INCHIRIERI );
```

```
--SA SE AFISEZE TOATE FILMELE(DENUMIREA) CE AU PRIMIT CEL PUTIN O NOTA INTRE 8.99 SI 10
SELECT DENUMIRE_FILM FROM FILME WHERE ID_FILM IN ( SELECT ID_FILM FROM RECENZII WHERE NOTA BETWEEN 8.99 AND 10);

--SA SE AFISEZE DENUMIREA SI LIMBA TUTUROR FILMELOR CARE AU ACTORI DE SEX FEMININ, IAR DACA LIMBA NU ESTE IN BAZA DE DATE, SA SE AFISEZE IN LOCUL ACESTEIA "Indisponibil"
SELECT DENUMIRE_FILM, NVL(LIMBA_FILM, 'Indisponibil') AS LIMBA FROM FILME WHERE ID_FILM IN (SELECT ID_FILM FROM DISTRIBUTIE JOIN ACTORI ON DISTRIBUTIE.ID_ACTOR=ACTORI.ID_ACTOR WHERE SEX='F');

--SA SE MODIFICE CAMPUL LIMBA_FILM IN "N/A" UNDE VALOAREA ESTE NULL
UPDATE FILME SET LIMBA_FILM='N/A' WHERE LIMBA_FILM IS NULL;

--SA SE AFISEZE PREFIXUL NUMERELOR DE TELEFON ALE ANGAJATILOR CE FOLOSESC SERVICII DE TELEFONIE MOBILA DIN ROMANIA
SELECT SUBSTR(TELEFON_ANGAJAT, 0, 4) AS PREFIX FROM ANGAJATI WHERE TELEFON_ANGAJAT LIKE '07%';

--AFISATI NUMELE ANGAJATILOR SI SALARIUL ACESTORA CARE AU SALARIUL MAI MARE DECAT SALARIUL MEDIU DIN FIRMA PENTRU FIECARE FUNCTIE
SELECT NUME, PRENUME, SALARIUL, DENUMIRE_FUNCTIE FROM ANGAJATI ANG JOIN FUNCTII FUN ON ANG.ID_FUNCTIE=FUN.ID_FUNCTIE
WHERE SALARIUL >= (SELECT AVG(SALARIUL) FROM ANGAJATI WHERE ID_FUNCTIE=ANG.ID_FUNCTIE);
```

```
SELECT NUME, PRENUME, SALARIUL, DENUMIRE_FUNCTIE FROM ANGAJATI ANG JOIN FUNCTII FUN ON ANG.ID_FUNCTIE=FUN.ID_FUNCTIE WHERE SALARIUL >= (SELECT AVG(SALARIUL) FROM ANGAJATI WHERE ID_FUNCTIE=ANG.ID_FUNCTIE)
```

NUME	PRENUME	SALARIUL	DENUMIRE_FUNCTIE
Constantin	Rares	2400	Lucrator comercial
Marcus	Stefan	3500	Supervizor magazin
Gheorghita	Vlad	2300	Lucrator comercial
Avram	Diana	2300	Lucrator comercial
Jean	Mohammed	4000	Manager

5 rows selected. 0.02 seconds

```
--SA SE MAREASCA CU 150 RON SALARIILE CARE SUNT SUB MEDIE
UPDATE ANGAJATI SET SALARIUL=SALARIUL+150 WHERE SALARIUL < ( SELECT AVG(SALARIUL) FROM ANGAJATI );

--SA SE AFISEZE FILMELE SI TOATE RECENZIILE ACESTORA, INCLUSIV CELE CARE NU AU RECENZII, IN ORDINE ALFABETICA
```

```
SELECT DENUMIRE_FILM, NOTA FROM FILME FULL OUTER JOIN RECENZII ON FILME.ID_FILM=RECENZII.ID_FILM ORDER BY DENUMIRE_FILM;
```

```
--SA SE AFISEZE MEDIA VARSTELOR CLIENTILOR CE AU CEL PUTIN 3 COMENZI

SELECT AVG(VARSTA) FROM CLIENTI WHERE ID_CLIENT IN (SELECT ID_CLIENT FROM INCHIRIERI GROUP BY ID_CLIENT HAVING COUNT(*)>=3);

--SA SE AFISEZE NUMARUL DE CLIENTI PENTRU FIECARE LUNA A ANULUI 2020

SELECT EXTRACT(MONTH FROM DATA) AS LUNA, COUNT(*) AS NUMAR_CLIENTI FROM INCHIRIERI WHERE EXTRACT(YEAR FROM DATA)=2020
GROUP BY EXTRACT(MONTH FROM DATA);

--SA SE AFISEZE LUNA CU CEI MAI MULTI CLIENTI DIN ANUL 2019

SELECT STATS_MODE(EXTRACT(MONTH FROM DATA)) AS LUNA FROM INCHIRIERI WHERE EXTRACT(YEAR FROM DATA)=2019;

--SA SE AFISEZE TOTI NUMELE ANGAJATILOR SI FUNCTIILE SUPERIORILOR SAI FOLOSIND DECODE

SELECT NUME, PRENUME, DECODE (ID_MANAGER, NULL, '-',
                                2, 'Supervizor magazin',
                                'Manager') AS FUNCTIE_SUPERIOR FROM ANGAJATI;

--SA SE CREEZE TABLEA "CLIENTI_ELIGIBILI_PENTRU_UN_FILM_GRATIS" CARE VA CONTINE ID-URILE CLIENTILOR CE AU CHELTUIT MINIM 20 LEI IN MAGAZIN

CREATE TABLE CLIENTI_ELIGIBILI_PENTRU_UN_FILM_GRATIS
AS
SELECT ID_CLIENT FROM INCHIRIERI INC JOIN RAND_INCHIRIERI RAN ON INC.ID_INCHIRIERE=RAN.ID_INCHIRIERE GROUP BY ID_CLIENT
HAVING SUM(PRET*ZILE_INCHIRIATE)>=20;

--SA SE MICSOREZE CU 5% PRETUL DE INCHIRIERE AL FILMELOR CE NU AU NICIO RECENZIC PESTE 7.99

UPDATE FILME SET PRET_INCHIRIERE=0.95*PRET_INCHIRIERE WHERE ID_FILM NOT IN (SELECT ID_FILM FROM RECENZII WHERE NOTA > 7.99);

--SA SE CREEZE UN SINONIM PENTRU TABELA RAND_INCHIRIERI SI CLIENTI_ELIGIBILI_PENTRU_UN_FILM_GRATIS

CREATE SYNONYM RI FOR RAND_INCHIRIERI;
CREATE SYNONYM CEPUFG FOR CLIENTI_ELIGIBILI_PENTRU_UN_FILM_GRATIS;

--SA SE AFISEZE SINONIMELE BAZEI DE DATE

SELECT SYNONYM_NAME, TABLE_NAME FROM USER_SYNONYMS;
```

```
--SA SE CREEZE O TABELA VIRTUALA CU DENUMIRE_FILM SI PRET_INCHIRIERE PENTRU FILMELE CE CONTIN CATEGORIA "Drama"

CREATE OR REPLACE VIEW V_FILME_DRAMA AS SELECT ID_FILM, DENUMIRE_FILM, PRET_INCHIRIERE FROM FILME WHERE LOWER(CATEGORIE_FILM) LIKE '%drama%';

--SA SE CREASCA CU 10% PRETUL DE INCHIRIERE AL FILMELOR DIN TABELA VIRTUALA

UPDATE V_FILME_DRAMA SET PRET_INCHIRIERE=1.10*PRET_INCHIRIERE;

--SA SE AFISEZE ID-UL SI MEDIA RECENZIILOR FILMULUI CU CEA MAI MARE MEDIE

SELECT ID_FILM, AVG(NOTA) AS MEDIE_NOTA FROM RECENZII GROUP BY ID_FILM ORDER BY MEDIE_NOTA DESC FETCH FIRST 1 ROWS ONLY;

--SA SE AFISEZE ID-UL INCHIRIERII SI VALOAREA TUTUROR INCHIERIILOR PESTE 20 DE LEI, MAI PUTIN CELE CE AU FOST DATE DUPA ANUL 2019

CREATE SYNONYM INC FOR INCHIRIERI;

SELECT RI.ID_INCHIRIERE, SUM(RI.PRET*RI.ZILE_INCHIRIATE) AS SUMA FROM INC, RI WHERE INC.ID_INCHIRIERE=RI.ID_INCHIRIERE
GROUP BY RI.ID_INCHIRIERE HAVING SUM(RI.PRET*RI.ZILE_INCHIRIATE) > 20
MINUS
SELECT RI.ID_INCHIRIERE, SUM(RI.PRET*RI.ZILE_INCHIRIATE) AS SUMA FROM INC, RI WHERE INC.ID_INCHIRIERE=RI.ID_INCHIRIERE AND EXTRACT(YEAR FROM INC.DA
TA) > 2019 GROUP BY RI.ID_INCHIRIERE;

--SA SE CALCULEZE DIFERENTA DINTRE BANII OBTINUTI DE PE URMA INCHIRIELILOR SI TOTALUL PLATIT FURNIZORILOR PENTRU FILME PENTRU A
VEDEA DACA MAGAZINUL SI-A ACOPERIT CHELTUIELILE CU FILMELE

SELECT X - Y AS "PROFIT/PIERDERE" FROM ( SELECT SUM(RI.PRET*RI.ZILE_INCHIRIATE) X FROM RI ), (SELECT SUM(FILME.PRET_FURNIZOR*FILME.STOC) Y
FROM FILME);

--SA SE AFISEZE CAT CHELTUIESTE UN CLIENT, IN MEDIE, PE INCHIRIERE

SELECT TRUNC( AVG(VALOARE_INCHIRIERE), 2 ) AS MEDIE_CLIENT FROM (SELECT ID_INCHIRIERE, SUM(PRET*ZILE_INCHIRIATE)
AS VALOARE_INCHIRIERE FROM RI GROUP BY ID_INCHIRIERE);

SELECT TRUNC( AVG(VALOARE_INCHIRIERE), 2 ) AS MEDIE_CLIENT FROM (SELECT ID_INCHIRIERE, SUM(PRET*ZILE_INCHIRIATE) AS VALOARE_INCHIRIERE FROM RI GROUP BY ID_INCHIRIERE)



| MEDIE_CLIENT |
|--------------|
| 10.61        |



Statement processed. 0.01 seconds
```

```

--SA SE AFISEZE DENUMIREA SI CATEGORIA FILMELOR INCHIRIATE PE O PERIOADA MAI MARE DE 3 ZILE

SELECT DENUMIRE_FILM, CATEGORIE_FILM FROM FILME WHERE ID_FILM IN ( SELECT ID_FILM FROM RAND_INCHIRIERI WHERE ZILE_INCHIRIATE > 3);

--RAPORTUL DINTRE CLIENTII CARE S-AU INTORS IN MAGAZIN(AU 2 SAU MAI MULTE INCHIRIERI) SI CEI CARE NU S-AU MAI INTORS

SELECT X/Y AS "RAPORT" FROM (SELECT COUNT(*) X FROM
(SELECT ID_CLIENT, COUNT(*) FROM INCHIRIERI GROUP BY ID_CLIENT HAVING COUNT(*) >= 2),
(SELECT COUNT(*) Y FROM (SELECT ID_CLIENT, COUNT(*) FROM INCHIRIERI GROUP BY ID_CLIENT HAVING COUNT(*) < 2));

--SA SE STEARGA DIN BAZA DE DATE TOATE INCHIRIERILE MAI VECHI DE 1 AN

DELETE FROM RAND_INCHIRIERI WHERE SYSDATE - DATA_INCHIRIERE > 365;
DELETE FROM INCHIRIERI WHERE SYSDATE - DATA > 365;

--SA SE STEARGA CLIENTII CARE NU AU MAI INCHIRIAT UN FILM DE CEL PUTIN 1 AN

DELETE FROM CLIENTI WHERE ID_CLIENT NOT IN ( SELECT ID_CLIENT FROM INCHIRIERI );

SA SE AFISEZE ANGAJATII SI NIVELUL IERARHIC AL ACESTORA, INCEPAND CU MANAGERUL, SI SA SE ORDONEZE DESCRESATOR IN FUNCTIE DE NIVEL

SELECT ID_ANGAJAT, NUME, PRENUME, LEVEL FROM ANGAJATI CONNECT BY PRIOR ID_ANGAJAT=ID_MANAGER START WITH ID_MANAGER IS NULL
ORDER BY LEVEL DESC;

--SA SE AFISEZE NUMARUL DE SUBORDONATI AI MANAGERULUI, GRUPATI PE FUNCTII

SELECT ID_FUNCTIE, COUNT(*) AS "SUBORDONATI" FROM ANGAJATI CONNECT BY PRIOR ID_ANGAJAT=ID_MANAGER START WITH ID_MANAGER IS NULL
GROUP BY ID_FUNCTIE;

--SA SE AFISEZE ID-URILE TUTUROR FILMELOR CARE AU FOST INCHIRIATE DE 3 SAU 4 ORI

SELECT FILME.ID_FILM, COUNT(*) FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM=RAND_INCHIRIERI.ID_FILM GROUP BY FILME.ID_FILM
HAVING COUNT(*)=3
UNION
SELECT FILME.ID_FILM, COUNT(*) FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM=RAND_INCHIRIERI.ID_FILM GROUP BY FILME.ID_FILM
HAVING COUNT(*)=4;

```

--SA SE AFISEZE ID-UL FILMELOR CARE AU FOST COMANDATE DE CEL PUTIN 2 ORI SI AU CEL PUTIN O RECENZIE

```
SELECT FILME.ID_FILM FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM=RAND_INCHIRIERI.ID_FILM GROUP BY FILME.ID_FILM  
HAVING COUNT(*)>=2  
INTERSECT  
SELECT ID_FILM FROM FILME WHERE ID_FILM IN ( SELECT ID_FILM FROM RECENZII );
```

Capitolul 2 - PL/SQL

Exercitiul 1:

The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and App Gallery. The SQL Workshop tab is selected. The schema dropdown shows 'WKSP_ROTARUTEODORGA'. The main area displays a PL/SQL block:

```
1 -- Sa se afiseze numele managerului
2 DECLARE
3     V_NUME VARCHAR2(30);
4     V_PRENUME VARCHAR2(30);
5 BEGIN
6     SELECT NUME,PRENUME INTO V_NUME, V_PRENUME FROM ANGAJATI WHERE ID_MANAGER IS NULL;
7     DBMS_OUTPUT.PUT_LINE('Numele managerului este'|| V_NUME || ' ' || V_PRENUME);
8 END;
9
10
```

The code is run successfully, and the results panel shows the output: "Numele managerului este Jean Mohammed". Below the results, it says "Statement processed." and "0.01 seconds".

Exercitiul 2:

The screenshot shows the Oracle Application Express SQL Workshop interface. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and App Gallery. The SQL Workshop tab is selected. The schema dropdown is set to WKSP_ROTARUTEODORGA. The code editor contains the following PL/SQL block:

```
1 -- Sa se afiseze numele managerului folosind %TYPE
2 DECLARE
3     V_NUME ANGAJATI.NUME%TYPE;
4     V_PRENUME ANGAJATI.PRENUME%TYPE;
5 BEGIN
6     SELECT NUME,PRENUME INTO V_NUME, V_PRENUME FROM ANGAJATI WHERE ID_MANAGER IS NULL;
7     DBMS_OUTPUT.PUT_LINE('Numele managerului este '|| V_NUME || ' ' || V_PRENUME);
8 END;
```

The results panel shows the output of the query: "Numele managerului este Jean Mohammed". Below the results, it says "Statement processed." and "0.05 seconds".

Application Express 20.2.0.0.0.20

Exercitiul 3:

The screenshot shows the Oracle Application Express (APEX) interface, specifically the SQL Workshop module. The top navigation bar includes links for APEX, App Builder, SQL Workshop, Team Development, and App Gallery. The current schema is set to WKSP_ROTARUTEODORG. The SQL Commands editor contains the following PL/SQL code:

```
1 --Mariti cu 10 bucati stocul filmului cu id-ul 7
2 --Afisati filmul
3
4 DECLARE
5     V_DENUMIRE_FILM FILME.DENUMIRE_FILM%TYPE;
6     V_ID_FILM FILME.ID_FILM%TYPE;
7 BEGIN
8     UPDATE FILME SET STOC = STOC + 10 WHERE ID_FILM = 7;
9     SELECT DENUMIRE_FILM, ID_FILM INTO V_DENUMIRE_FILM, V_ID_FILM FROM FILME WHERE ID_FILM=7;
10    DBMS_OUTPUT.PUT_LINE(V_DENUMIRE_FILM || ' ' || V_ID_FILM);
11 END;
12
13
```

The results section shows the output of the executed query:

```
Cardinalul 1
1 row(s) updated.

0.00 seconds
```

At the bottom left, there are user profile icons and a language selection button (en). The bottom right corner displays the copyright notice: Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved. Application Express 20.2.0.0.0.20.

Exercitiul 4:

Oracle SQL Developer: C:\Users\FOLIO\RotaruTeodorSGBD~11.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD~11.sql FILME

SQL Worksheet History 3,4890008 seconds

Worksheet Query Builder

```
--Sa se actualizeze pretul de inchiriere al filmului Insidious
--Sa se actualizeze pretul de inchiriere al filmelor care au fost inchiriate de mai putin de 3 ori cu un pret introdus de la tastatura

BEGIN
    UPDATE FILME SET PRET_INCHIRIERE = &PRET_INPUT WHERE 3 > (SELECT COUNT(*) FROM RAND_INCHIRIERI WHERE RAND_INCHIRIERI.ID_FILM = FILME.ID_FILM);
END;
ROLLBACK;
```

Script Output Task completed in 3,489 seconds

```
old:BEGIN
    UPDATE FILME SET PRET_INCHIRIERE = &PRET_INPUT WHERE 3 > (SELECT COUNT(*) FROM RAND_INCHIRIERI WHERE RAND_INCHIRIERI.ID_FILM = FILME.ID_FILM);
END;
new:BEGIN
    UPDATE FILME SET PRET_INCHIRIERE = 5 WHERE 3 > (SELECT COUNT(*) FROM RAND_INCHIRIERI WHERE RAND_INCHIRIERI.ID_FILM = FILME.ID_FILM);
END;

PL/SQL procedure successfully completed.
```

SQL History

Line 74 Column 1 | Insert | Modified | Windows: C

Exercitiul 5:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar contains a Connections tree showing a connection to 'RotaruTeodorSGBD' with tables like FILME, FUNCTII, and REPORTS. The Reports section lists Analytic View Reports, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, TimesTen Reports, and User Defined Reports. The main workspace has two tabs: 'RotaruTeodorSGBD.sql' and 'RotaruTeodorSGBD~11.sql'. The 'RotaruTeodorSGBD~11.sql' tab is active, showing a PL/SQL script. The script starts with comments about deleting a movie review and selecting its ID. It then begins a PL/SQL block with a BEGIN keyword, followed by an INSERT INTO statement for the FILME table. The script ends with an END; statement. Below the script, the 'Script Output' window displays the results of the execution. It shows an error message about mismatched input data and date formats, followed by the actual SQL statements executed: an old row was deleted and a new row was inserted with ID 49, title 'A haunted house', category 'Horror', language 'Engleza', duration 121, release date '05/09/2009', price 123, and stock level 99. The bottom status bar indicates the script completed in 48,214 seconds.

```
--sa se stearga recenzia filmului cu id-ul 7 din tabela RECENZII
--BEGIN
--  DELETE FROM RECENZII WHERE ID_FILM = 7;
--END;

--SELECT * FROM RECENZII WHERE ID_FILM = 7;

BEGIN
  INSERT INTO FILME VALUES(&id_input, '&denumire_input', '&categorie_input', '&limba_input', &durata_input, TO_DATE('&data_input', 'dd/mm/yyyy'), &pret_film_input, &stoc_input);
END;

*Action: Fix the input data or the date format model to make sure the
elements match in number and type. Then retry the operation.
old:BEGIN
  INSERT INTO FILME VALUES(&id_input, '&denumire_input', '&categorie_input', '&limba_input', &durata_input, TO_DATE('&data_input', 'dd/mm/yyyy'), &pret_film_input, &stoc_input);
END;
new:BEGIN
  INSERT INTO FILME VALUES(49, 'A haunted house', 'Horror', 'Engleza', 121, TO_DATE('05/09/2009', 'dd/mm/yyyy'), 123, 12, 99);
END;

PL/SQL procedure successfully completed.
```

Exercitiul 6:

Oracle SQL Developer : C:\Users\FOLIO\RotaruTeodorSGBD~1.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD~1.sql CLIENTI_FILME

SQL Worksheet History 0,257 seconds

Worksheet Query Builder

```
--tema partea a doua

--SA SE AFISEZE DATELE CLIENTULUI DE SEX MASCULIN SI CU ID-UL 10
SET SERVEROUTPUT ON
DECLARE
    TYPE V_COMPUS IS RECORD(
        V_ID_CLIENT CLIENTI_FILME.ID_CLIENT%TYPE,
        V_NUME_CLIENT CLIENTI_FILME.NUME%TYPE,
        V_EMAIL_CLIENT CLIENTI_FILME.EMAIL_CLIENT%TYPE
    );
    V_DETALII_CLIENT V_COMPUS;
BEGIN
    SELECT ID_CLIENT, NUME, EMAIL_CLIENT INTO V_DETALII_CLIENT FROM CLIENTI_FILME WHERE SEX='M' AND ID_CLIENT=10;
    DBMS_OUTPUT.PUT_LINE('DETALII CLIENT:' || V_DETALII_CLIENT.V_ID_CLIENT || ', ' || V_DETALII_CLIENT.V_NUME_CLIENT || ', ' || V_DETALII_CLIENT.V_EMAIL_CLIENT);
END;
```

Script Output Task completed in 0,257 seconds

```
DETALII CLIENT:10, Calin, calinB@gmail.com

PL/SQL procedure successfully completed.
```

SQL History

Exercitiul 7:

Oracle SQL Developer : C:\Users\FOLIO\RotaruTeodorSGBD~1.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD~1.sql INCHIRIERI

SQL Worksheet History ScriptRunner Task

Worksheet Query Builder

```
-- V_DETALII_CLIENT V_COMPUS;
--BEGIN
--  SELECT ID_CLIENT, NUME, EMAIL_CLIENT INTO V_DETALII_CLIENT FROM CLIENTI_FILME WHERE SEX='M' AND ID_CLIENT=10;
--  DBMS_OUTPUT.PUT_LINE('DETALII CLIENT:' || V_DETALII_CLIENT.V_ID_CLIENT || ', ' || V_DETALII_CLIENT.V_NUME_CLIENT || ', ' || V_DETALII_CLIENT.V_EMAIL_CLIENT);
--END;

--sa se afiseze de cate ori a fost închiriat filmul cu id 4
DECLARE
    NR_INCHIRIERI NUMBER(3);
BEGIN
    SELECT COUNT(*) INTO NR_INCHIRIERI
    DBMS_OUTPUT.PUT_LINE('Acest film a fost inchiriat ' || NR_INCHIRIERI || ' ori');
END;
```

Enter Substitution Variable

Enter value for ID_FILM: 4

OK Renunțare

Script Output

ScriptRunner Task

SQL History

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

| Line 56 Column 1 | Insert | Modified | Windows: C

Oracle SQL Developer : C:\Users\FOLIO\RotaruTeodorSGBD~1.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

SQL Worksheet History

Worksheet Query Builder

```
-- V_DETALII_CLIENT V_COMPUS;
--BEGIN
--  SELECT ID_CLIENT, NUME, EMAIL_CLIENT INTO V_DETALII_CLIENT FROM CLIENTI_FILME WHERE SEX='M' AND ID_CLIENT=10;
--  DBMS_OUTPUT.PUT_LINE('DETALII CLIENT:' || V_DETALII_CLIENT.V_ID_CLIENT || ', ' || V_DETALII_CLIENT.V_NUME_CLIENT || ', ' || V_DETALII_CLIENT.V_EMAIL_CLIENT);
--END;

--sa se afiseze de cate ori a fost inchiriat filmul cu id-ul introdus de utilizator

DECLARE
    NR_INCHIRIERI NUMBER(3);
BEGIN
    SELECT COUNT(*) INTO NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=&ID_FILM;
    DBMS_OUTPUT.PUT_LINE('Acest film a fost inchiriat de ' || NR_INCHIRIERI || ' ori');
END;
```

Script Output

Task completed in 31,697 seconds

```
END;
Acest film a fost inchiriat de 3 ori

PL/SQL procedure successfully completed.
```

SQL History

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

Line 54 Column 5 | Insert | Modified | Windows: C | Filter

Exercitiul 8:

Oracle SQL Developer : RotaruTeodorSGBD~1

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD~1

FILME FUNCTII FUNCTII_FILME INCHIRIERI ISTORIC_FUNCTII ISTORIC_FUNCTII_FILME_FILM LOCATII PRODUSE RAND_COMENZI RAND_INCHIRIERI RECENZII ID_FILM ID_CRITIC NOTA REGIUNI

Worksheet Query Builder

```
--BEGIN  
--    SELECT COUNT(*) INTO NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=&ID_FILM;  
--    DBMS_OUTPUT.PUT_LINE('Acest film a fost inchiriat de ' || NR_INCHIRIERI || ' ori');  
--END;  
  
--sa se stearga recenzia filmului cu id-ul 7 din tabela RECENZII  
BEGIN  
    DELETE FROM RECENZII WHERE ID_FILM = 7;  
END;  
  
SELECT * FROM RECENZII WHERE ID_FILM = 7;
```

Script Output Query Result

All Reports Analytic View Reports Data Dictionary Reports Data Modeler Reports OLAP Reports TimesTen Reports User Defined Reports

ID_FILM ID_CRITIC NOTA

SQL History

| Line 63 Column 42 | Insert | Modified | Windows: C|

Exercitiul 9:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema with the 'Tables (Filtered)' section expanded, showing tables like ACTORI, ANGAJATI, ANGAJATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUTIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME_FI, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII, REGIUNI, TARI, and Views.

The central workspace contains a PL/SQL script:

```
--SA SE AFISEZE ANGAJATII CARE AU SALARIUL MAI MIC DECAT CEL MEDIU

SET SERVEROUTPUT ON
SET AUTOPRINT ON
VARIABLE SALARIU_MEDIU NUMBER

BEGIN
    SELECT AVG(SALARIUL) INTO :SALARIU_MEDIU FROM ANGAJATI_FILME;
END;

SELECT NUME, PRENUME FROM ANGAJATI_FILME WHERE SALARIUL < :SALARIU_MEDIU;
```

The 'Query Result' tab shows the output of the query:

NUME	PRENUME
Constantin	Rares
Gheorghita	Vlad
Avram	Diana
Topor	Ana
Clopot	Marius

Below the table, the value of the variable SALARIU_MEDIU is displayed as 2700.

The bottom status bar indicates: Line 22 Column 74 | Insert | Modified | Windows: Cf

Exercitiul 10:

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the Worksheet tab. The script retrieves information about a film with ID 4 and prints its title, category, and language to the console. The output window shows the results and a success message. The bottom right corner of the interface has a status bar with various icons and text.

```
--SA SE AFISEZE DENUMIREA, CATEGORIA SI LIMBA FILMULUI CU ID-UL 4
SET SERVEROUTPUT ON
DECLARE
    TYPE TIP_COMPUS IS RECORD
    (
        V_DENUMIRE FILME.DENUMIRE_FILM%TYPE,
        V_CATEGORIE FILME.CATEGORIE_FILM%TYPE,
        V_LIMBA FILME.LIMBA_FILM%TYPE
    );
    INFO_FILM TIP_COMPUS;
BEGIN
    SELECT DENUMIRE_FILM, CATEGORIE_FILM, LIMBA_FILM INTO INFO_FILM FROM FILME WHERE ID_FILM = 4;
    DBMS_OUTPUT.PUT_LINE('Denumire: ' || INFO_FILM.V_DENUMIRE);
    DBMS_OUTPUT.PUT_LINE('Categorie: ' || INFO_FILM.V_CATEGORIE);
    DBMS_OUTPUT.PUT_LINE('Limba: ' || INFO_FILM.V_LIMBA);
END;
```

Script Output | Task completed in 0,462 seconds

```
Denumire: A walk to remember
Categorie: Dragoste / Drama
Limba: Engleza

PL/SQL procedure successfully completed.
```

SQL History | Line 1 Column 1 | Insert | Modified | Windows: C:\

Exercitiul 11:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations, navigation, and tools. The Connections panel on the left shows a single connection named "RotaruTeodorSGBD". The main workspace is a "Worksheet" tab, which displays the following PL/SQL code:

```
--SA SE MODIFICE NUMARUL DE TELEFON AL CLIENTULUI CU EMAILUL 'raduC@gmail.com' IN 0723996483 SI EMAILUL IN 'raduCernea@yahoo.com'

SET SERVEROUTPUT ON
DECLARE
    INFO_CLIENT CLIENTI_FILME%ROWTYPE;
BEGIN
    SELECT * INTO INFO_CLIENT FROM CLIENTI_FILME WHERE EMAIL_CLIENT = 'raduC@gmail.com';
    INFO_CLIENT.TELEFON := '0723996483';
    INFO_CLIENT.EMAIL_CLIENT := 'raduCernea@yahoo.com';
    DBMS_OUTPUT.PUT_LINE('Informatii actualizate => ' || INFO_CLIENT.NUME || ' ' || INFO_CLIENT.PRENUME || ' ' || INFO_CLIENT.TELEFON || ' ' || INFO_CLIENT.EMAIL_CLIENT);
END;
```

The "Script Output" panel at the bottom shows the results of the execution:

```
Informatii actualizate => Radu Cernea 0723996483 raduCernea@yahoo.com

PL/SQL procedure successfully completed.
```

The bottom right corner of the interface shows the page number: 29.

Exercitiul 12:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations, database connections, and code editing. A connection named "RotaruTeodorSGBD" is selected. The main workspace is a "Worksheet" tab where a PL/SQL script is being run. The script retrieves information about a client based on their email. The output window shows the results and a message indicating the procedure was successfully completed. The bottom right corner displays the line and column numbers (Line 9 Column 5) and other status information.

```
--SA SE AFISEZE INFORMATIILE DESPRE CLIENTUL CU EMAILUL 'raduC@gmail.com'

SET SERVEROUTPUT ON
DECLARE
    INFO_CLIENT CLIENTI_FILME%ROWTYPE;
BEGIN
    SELECT * INTO INFO_CLIENT FROM CLIENTI_FILME WHERE EMAIL_CLIENT = 'raduC@gmail.com';
    DBMS_OUTPUT.PUT_LINE('Informatii => ' || INFO_CLIENT.NUME || ' ' || INFO_CLIENT.PRENUME || ' ' || INFO_CLIENT.TELEFON);
END;
```

Script Output x

Informatii => Radu Cernea 0721275513

PL/SQL procedure successfully completed.

Messages - Log

Messages | Logging Page ^ | Statements ^

Line 9 Column 5 | Insert | Modified | Windows: C

Exercitiul 13:

The screenshot shows the Oracle SQL Developer interface. On the left, the Connections tree displays a single connection named "RotaruTeodorSGBD" under "Oracle Connections". The "Tables (Filtered)" node is expanded, showing various tables such as ACTORI, ANGAJATI, ANGAJATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUTIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII, REGIUNI, and TARI. Below this are sections for Views, Indexes, Packages, Procedures, Functions, Operators, Queues, and Queue Tables.

The central workspace contains a "Worksheet" tab with the following PL/SQL code:

```
--SA SE AFISEZE DE CATE ORI A FOST INCHIRIAT FILMUL CU ID-UL INTRODUS DE UTILIZATOR

SET SERVEROUTPUT ON
VARIABLE NR_INCHIRIERI NUMBER

BEGIN
    SELECT COUNT(*) INTO :NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=&ID_FILM;
END;

PRINT NR_INCHIRIERI
```

Below the worksheet is a "Script Output" window showing the results of the execution:

```
old:BEGIN
    SELECT COUNT(*) INTO :NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=&ID_FILM;
END;
new:BEGIN
    SELECT COUNT(*) INTO :NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=4;
END;

PL/SQL procedure successfully completed.
```

The bottom right corner of the interface shows status information: Line 8 Column 5, Insert, Modified, Windows: Cf.

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

RotaruTeodorSGBD

Tables (Filtered)

- ACTORI
- ANGAJATI
- ANGAJATI_FILME
- CLIENTI
- CLIENTI_FILME
- COMENZI
- CRITICI
- DEPARTAMENTE
- DISTRIBUETE
- FILME
- FUNCTII
- FUNCTII_FILME
- INCHIRIERI
- ISTORIC_FUNCTII
- ISTORIC_FUNCTII_FILME_FI
- LOCATII
- PRODUSE
- RAND_COMENZI
- RAND_INCHIRIERI
- RECENTII
- REGIUNI
- TARI

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

RotaruTeodorSGBD.sql RotaruTeodorSGBD 0,141 seconds

Worksheet Query Builder

```
--SA SE AFISEZE DE CATE ORI A FOST INCHIRIAT FILMUL CU ID-UL INTRODUS DE UTILIZATOR

SET SERVEROUTPUT ON
VARIABLE NR_INCHIRIERI NUMBER

BEGIN
    SELECT COUNT(*) INTO :NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=&ID_FILM;
END;

PRINT NR_INCHIRIERI
```

Script Output Task completed in 0,141 seconds

```
END;
new:BEGIN
    SELECT COUNT(*) INTO :NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM=4;
END;

PL/SQL procedure successfully completed.
```

NR_INCHIRIERI

```
-----
3
```

SQL History

Messages - Log

Messages Statements Logging Page

Line 10 Column 20 Insert Modified Windows: C

Exercitiul 14:

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections

+

- CRITICI
- DEPARTAMENTE
- DISTRIBUTIE
- FILME
 - ID_FILM
 - DENUMIRE_FILM
 - CATEGORIE_FILM
 - LIMBA_FILM
 - DURATA_FILM_MINUTE
 - DATA_LANSARE
 - PRET_FURNIZOR
 - PRET_INCHIRIERE
 - STOC
- FUNCTII
- FUNCTII_FILME
- INCHIRIERI
- ISTORIC_FUNCTII
- ISTORIC_FUNCTII_FILME_FI
- LOCATII
- PRODUSE
- RAND_COMENZI
- RAND_INCHIRIERI
 - ID_INCHIRIERE
 - ID_FILM
 - ZILE_INCHIRIATE
 - DATA_INCHIRIERE
 - DATA_RETURNARE
 - PRET
- RECENZII
- REGIUNI
- TARI
- Views
- Indexes

RotaruTeodorSGBD.sql x RotaruTeodorSGBD x

Worksheet Query Builder

```
--SA SE AFISEZE PRETUL DE INCHIRIERE AL FILMULUI CU ID-UL INTRODUS DE UTILIZATOR SI SA SE MODIFICE ASTFEL:  
--DACA A FOST INCHIRIAT DE MAI MULT DE 3 ORI, SA SE MAREASCA CU 10%  
--ALTFEL, SA SE SCADA MICSOREZE CU 20%  
  
SET SERVEROUTPUT ON  
DECLARE  
    V_PRET_INCHIRIERE FILME.PRET_INCHIRIERE%TYPE;  
    V_NR_INCHIRIERI NUMBER(2);  
    V_ID_FILM FILME.ID_FILM%TYPE;  
BEGIN  
    V_ID_FILM := &ID_FILM;  
    SELECT COUNT(*) INTO V_NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM = V_ID_FILM;  
    DBMS_OUTPUT.PUT_LINE('Filmul ales a fost inchiriat de ' || V_NR_INCHIRIERI || ' ori :');  
    IF V_NR_INCHIRIERI > 3 THEN  
        DBMS_OUTPUT.PUT_LINE('Se marestea pretul cu 10%...');  
        UPDATE FILME SET PRET_INCHIRIERE=1.1*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
        DBMS_OUTPUT.PUT_LINE('DONE');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Se micșorează pretul cu 20%...');  
        UPDATE FILME SET PRET_INCHIRIERE=0.8*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
        DBMS_OUTPUT.PUT_LINE('DONE');  
    END IF;  
END;
```

Script Output x

Task completed in 5,705 seconds

DBMS_OUTPUT.PUT_LINE('DONE');

SQL History

Messages - Log

Filter Messages Logging Page Statements

Line 23 Column 5 | Insert | Modified | Windows: Cf

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD

Worksheet Query Builder

```
--SA SE AFISEZE PRETUL DE INCHIRIERE AL FILMULUI CU ID-UL INTRODUS DE UTILIZATOR SI SA SE MODIFICE ASTFEL:  
--DACA A FOST INCHIRIAT DE MAI MULT DE 3 ORI, SA SE MAREASCA CU 10%  
--ALTfel, SA SE SCADA MICSOREZE CU 20%  
  
SET SERVEROUTPUT ON  
DECLARE  
    V_PRET_INCHIRIERE FILME.PRET_INCHIRIERE%TYPE;  
    V_NR_INCHIRIERI NUMBER(2);  
    V_ID_FILM FILME.ID_FILM%TYPE;  
BEGIN  
    V_ID_FILM := &ID_FILM;  
    SELECT COUNT(*) INTO V_NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM = V_ID_FILM;  
    DBMS_OUTPUT.PUT_LINE('Filmul ales a fost inchiriat de ' || V_NR_INCHIRIERI || ' ori :');  
    IF V_NR_INCHIRIERI > 3 THEN  
        DBMS_OUTPUT.PUT_LINE('Se marestea pretul cu 10%...');  
        UPDATE FILME SET PRET_INCHIRIERE=1.1*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
        DBMS_OUTPUT.PUT_LINE('DONE');  
    ELSE  
        DBMS_OUTPUT.PUT_LINE('Se micsoreaza pretul cu 20%...');  
        UPDATE FILME SET PRET_INCHIRIERE=0.8*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
        DBMS_OUTPUT.PUT_LINE('DONE');  
    END IF;  
    DBMS_OUTPUT.PUT_LINE('DONE');  
END;  
/
```

Script Output Task completed in 5,705 seconds

```
DBMS_OUTPUT.PUT_LINE('DONE');  
END IF;  
END;  
Filmul ales a fost inchiriat de 3 ori :  
Se marestea pretul cu 10%...  
DONE  
  
PL/SQL procedure successfully completed.
```

SQL History Messages - Log
Messages Logging Page Statements

Line 23 Column 5 | Insert | Modified | Windows: Cf

Exercitiul 15:

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

- Connections:** RotaruTeodorSGBD
- Worksheet:** The code is written in PL/SQL. It includes comments about displaying the rental price of a movie based on the user's input and modifying it if the movie has been rented more than 3 times. The code uses a CASE expression to calculate a 10% discount for rentals over 3 times and a 20% discount otherwise.
- Script Output:** The output shows the execution results:
 - Filmul ales a fost inchiriat de 3 ori :)
 - Se modifica pretul...
 - PL/SQL procedure successfully completed.
- Messages - Log:** Shows the log messages for the executed statement.
- SQL History:** Shows the history of previous SQL statements.
- Components:** Shows the available components for the current session.

Exercitiul 16:

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD FILME

Worksheet Query Builder

```
--SA SE AFISEZE PRETUL DE INCHIRIERE AL FILMULUI CU ID-UL INTRODUS DE UTILIZATOR SI SA SE MODIFICE ASTFEL ****FOLOSIND CASE STATEMENT!!!!!!:  
--DACA A FOST INCHIRIAT DE MAI MULT DE 3 ORI, SA SE MAREASCA CU 10%  
--ALTFEL, SA SE SCADA MICSOREZE CU 20%  
SET SERVEROUTPUT ON  
DECLARE  
    V_PRET_INCHIRIERE FILME.PRET_INCHIRIERE%TYPE;  
    V_NR_INCHIRIERI NUMBER(2);  
    V_ID_FILM FILME.ID_FILM%TYPE;  
BEGIN  
    SELECT ID_FILM INTO V_ID_FILM FROM FILME WHERE ID_FILM = &IF_FILM;  
  
    IF V_ID_FILM IS NOT NULL THEN  
        SELECT COUNT(*) INTO V_NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM = V_ID_FILM;  
        DBMS_OUTPUT.PUT_LINE('Filmul ales a fost inchiriat de ' || V_NR_INCHIRIERI || ' ori :');  
        DBMS_OUTPUT.PUT_LINE('Se modifica pretul...');  
        DBMS_OUTPUT.PUT_LINE('DONE');  
        CASE  
            WHEN V_NR_INCHIRIERI > 3 THEN UPDATE FILME SET PRET_INCHIRIERE=1.1*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
            ELSE UPDATE FILME SET PRET_INCHIRIERE=0.8*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;  
        END CASE;  
    END IF;  
  
    EXCEPTION  
        WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film');  
    END;
```

Script Output x | Task completed in 1,529 seconds

END CASE;
END IF.

SQL History Messages - Log
Messages Logging Page Statements

Line 71 Column 5 | Insert | Modified | Windows: Cf

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD FILME

Worksheet Query Builder

```
SELECT ID_FILM INTO V_ID_FILM FROM FILME WHERE ID_FILM = &IF_FILM;

IF V_ID_FILM IS NOT NULL THEN
    SELECT COUNT(*) INTO V_NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM = V_ID_FILM;
    DBMS_OUTPUT.PUT_LINE('Filmul ales a fost inchiriat de ' || V_NR_INCHIRIERI || ' ori :)');
    DBMS_OUTPUT.PUT_LINE('Se modifica pretul...');
    DBMS_OUTPUT.PUT_LINE('DONE');
    CASE
        WHEN V_NR_INCHIRIERI > 3 THEN UPDATE FILME SET PRET_INCHIRIERE=1.1*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;
        ELSE UPDATE FILME SET PRET_INCHIRIERE=0.8*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;
    END CASE;
    END IF;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film');
    END;
```

Script Output x

```
| Task completed in 1,529 seconds
```

```
END CASE;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film');
END;
Nu exista acest film

PL/SQL procedure successfully completed.
```

SQL History Messages - Log

Line 65 Column 15 | Insert | Modified | Windows: Ci

Oracle SQL Developer : RotaruTeodorSGBD

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql RotaruTeodorSGBD FILME

Worksheet Query Builder

```
SELECT ID_FILM INTO V_ID_FILM FROM FILME WHERE ID_FILM = &IF_FILM;

IF V_ID_FILM IS NOT NULL THEN
    SELECT COUNT(*) INTO V_NR_INCHIRIERI FROM RAND_INCHIRIERI WHERE ID_FILM = V_ID_FILM;
    DBMS_OUTPUT.PUT_LINE('Filmul ales a fost inchiriat de ' || V_NR_INCHIRIERI || ' ori :');
    DBMS_OUTPUT.PUT_LINE('Se modifica pretul...');
    DBMS_OUTPUT.PUT_LINE('DONE');
    CASE
        WHEN V_NR_INCHIRIERI > 3 THEN UPDATE FILME SET PRET_INCHIRIERE=1.1*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;
        ELSE UPDATE FILME SET PRET_INCHIRIERE=0.8*PRET_INCHIRIERE WHERE ID_FILM = V_ID_FILM;
    END CASE;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film');
END;
```

Script Output

```
Task completed in 1,1 seconds
WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film');
END;
Filmul ales a fost inchiriat de 3 ori :
Se modifica pretul...
DONE

PL/SQL procedure successfully completed.
```

SQL History Messages - Log

Line 71 Column 5 | Insert | Modified | Windows: C:\

The screenshot shows Oracle SQL Developer interface. The left sidebar displays the database schema with tables like ACTORI, ANGAJATI, FILME, etc. The central workspace contains a PL/SQL script that checks if a movie exists and then updates its rental price based on the number of rentals. The 'Script Output' pane shows the execution results, including the output of DBMS_OUTPUT.PUT_LINE statements and a success message. The bottom status bar indicates the current line and column numbers, and various navigation and modification buttons.

Exercitiul 17:

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

- Connections:** RotaruTeodorSGBD
- Tables (Filtered):** ACTORI, ANGAJATI, ANGAJATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME_FI.
- Worksheet:** Query Builder tab selected. The code is a PL/SQL procedure named INCHIRIERI:

```
--SA SE AFISEZE TOTI ANGAJATII CU ID-UL DIN INTERVALUL INTRODUS DE UTILIZATOR DOAR DACA AU MAI MULT DE 2 FILME VANDUTE (INCHIRIATE)
SET SERVEROUTPUT ON
DECLARE
    CAPAT_INFERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    CAPAT_SUPERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    INFO_ANGAJAT ANGAJATI_FILME%ROWTYPE;
BEGIN
    CAPAT_INFERIOR := &CAPAT_INFERIOR;
    CAPAT_SUPERIOR := &CAPAT_SUPERIOR;
    LOOP
        SELECT * INTO INFO_ANGAJAT FROM ANGAJATI_FILME A WHERE A.ID_ANGAJAT = CAPAT_INFERIOR AND 2 < (SELECT COUNT(*) FROM INCHIRIERI I WHERE I.ID_
        DBMS_OUTPUT.PUT_LINE('Pentru iteratia(id-ul) ' || CAPAT_INFERIOR || ' : ' || INFO_ANGAJAT.NUME || ' ' || INFO_ANGAJAT.PRENUME);
        CAPAT_INFERIOR := CAPAT_INFERIOR+1;
    EXIT WHEN CAPAT_INFERIOR > CAPAT_SUPERIOR;
END LOOP;

EXCEPTION WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Pentru iteratia(id-ul) ' || CAPAT_INFERIOR || ' Acest angajat nu respecta conditiile. Iesire din iteratie...');
END;
```
- Script Output:** Task completed in 2,365 seconds. Output:

```
END;
Pentru iteratia(id-ul) 1 : Constantin Rares
Pentru iteratia(id-ul) 2 Acest angajat nu respecta conditiile. Iesire din iteratie...

PL/SQL procedure successfully completed.
```
- Messages - Log:** Filter, Messages, Logging Page, Statements.
- Bottom Status Bar:** Line 77 Column 8, Insert, Modified, Windows: Cf

Exercitiul 18:

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

+ RotaruTeodorSGBD

- Tables (Filtered)
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories
- Editions
- Application Express
- Java
- XML Schemas
- XML DB Repository
- OLAP Option
- Scheduler
- Recycle Bin
- Other Users

Oracle NoSQL Connections

Database Schema Service Connections

RotaruTeodorSGBD.sql x TEMA3.sql x RotaruTeodorSGBD x

SQL Worksheet History

Worksheet Query Builder

```
--SA SE AFISEZE TOTI ANGAJATII CU ID-UL DIN INTERVALUL INTRODUS DE UTILIZATOR DOAR DACA AU MAI MULT DE 2 FILME VANDUTE (INCHIRIATE)
-- ****FOLOSIND WHILE!***
SET SERVEROUTPUT ON
DECLARE
    CAPAT_INFERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    CAPAT_SUPERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    INFO_ANGAJAT ANGAJATI_FILME%ROWTYPE;
BEGIN
    CAPAT_INFERIOR := &CAPAT_INFERIOR;
    CAPAT_SUPERIOR := &CAPAT_SUPERIOR;
    WHILE CAPAT_INFERIOR < CAPAT_SUPERIOR LOOP
        SELECT * INTO INFO_ANGAJAT FROM ANGAJATI_FILME A WHERE A.ID_ANGAJAT = CAPAT_INFERIOR AND 2 < (SELECT COUNT(*) FROM INCHIRIERI I WHERE I.ID_
DBMS_OUTPUT.PUT_LINE('Pentru iteratia(id-ul) ' || CAPAT_INFERIOR || ' : ' || INFO_ANGAJAT.NUME || ' ' || INFO_ANGAJAT.PRENUME);
        CAPAT_INFERIOR := CAPAT_INFERIOR+1;
    END LOOP;

    EXCEPTION WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Pentru iteratia(id-ul) ' || CAPAT_INFERIOR || ' Acest angajat nu respecta conditiile. I');
END;
```

Script Output x

Task completed in 2,762 seconds

Pentru iteratia(id-ul) 1 : Constantin Rares
Pentru iteratia(id-ul) 2 Acest angajat nu respecta conditiile. Iesire din iteratie...

PL/SQL procedure successfully completed.

SQL History x Filter

Messages - Log x

Messages Logging Page Statements

Line 113 Column 1 | Insert | Modified | Windows: C

Exercitiul 19:

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql TEMA3.sql RotaruTeodorSGBD

SQL Worksheet History 1,9789997 seconds RotaruTeodorSGBD

Worksheet Query Builder

```
--SA SE AFISEZE TOTI ANGAJATII CU ID-UL DIN INTERVALUL INTRODUS DE UTILIZATOR DOAR DACA AU MAI MULT DE 2 FILME VANDUTE(INCHIRIATE)
-- ****FOLOSIND FOR!***
SET SERVEROUTPUT ON
DECLARE
    CAPAT_INFERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    CAPAT_SUPERIOR ANGAJATI_FILME.ID_ANGAJAT%TYPE;
    INFO_ANGAJAT ANGAJATI_FILME%ROWTYPE;
BEGIN
    CAPAT_INFERIOR := &CAPAT_INFERIOR;
    CAPAT_SUPERIOR := &CAPAT_SUPERIOR;
    FOR ID IN CAPAT_INFERIOR..CAPAT_SUPERIOR LOOP
        SELECT * INTO INFO_ANGAJAT FROM ANGAJATI_FILME A WHERE A.ID_ANGAJAT = ID AND 2 < (SELECT COUNT(*) FROM INCHIRIERI I WHERE I.ID_ANGAJAT = II
        DBMS_OUTPUT.PUT_LINE('Pentru iteratia(id-ul) ' || CAPAT_INFERIOR || ' : ' || INFO_ANGAJAT.NUME || ' ' || INFO_ANGAJAT.PRENUME);
    END LOOP;

    EXCEPTION WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Pentru iteratia curenta: Acest angajat nu respecta conditiile. Iesire din iteratie...');
END;
```

Script Output Task completed in 1,979 seconds

```
Pentru iteratia(id-ul) 1 : Constantin Rares
Pentru iteratia curenta: Acest angajat nu respecta conditiile. Iesire din iteratie...

PL/SQL procedure successfully completed.
```

SQL History Messages - Log Filter Messages Logging Page Statements

| Line 132 Column 5 | Insert | Modified | Windows: C|

Exercitiul 20:

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD TEMA3.sql

SQL Worksheet History 0,39500001 seconds RotaruTeodorSGBD

Worksheet Query Builder

```
--SA SE INTRODUCĂ TOTI ANGAJATII ÎNTR-UN TABLOU INDEXAT, SA SE AFISEZE ACESTI ANGAJATI SI NUMARUL TOTAL DE ANGAJATI
SET SERVEROUTPUT ON
DECLARE
    TYPE ROW_NUM IS TABLE OF ANGAJATI_FILME.NUME%TYPE INDEX BY PLS_INTEGER;
    V_TABELA ROW_NUM;
    INDEX_TABELA ANGAJATI_FILME.ID_ANGAJAT%TYPE := 1;
    INDEX_FINAL ANGAJATI_FILME.ID_ANGAJAT%TYPE;
BEGIN
    SELECT MAX(ID_ANGAJAT) INTO INDEX_FINAL FROM ANGAJATI_FILME;
    LOOP
        SELECT NUME INTO V_TABELA(INDEX_TABELA) FROM ANGAJATI_FILME WHERE ID_ANGAJAT = INDEX_TABELA;
        INDEX_TABELA := INDEX_TABELA+1;
        EXIT WHEN INDEX_TABELA > INDEX_FINAL;
    END LOOP;

    FOR INDEX_TABELA IN V_TABELA.FIRST..V_TABELA.LAST LOOP
        IF V_TABELA.EXISTS(INDEX_TABELA) THEN
            DBMS_OUTPUT.PUT_LINE('Numele angajatului cu id-ul ' || INDEX_TABELA || ' este ' || V_TABELA(INDEX_TABELA));
        END IF;
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Numarul total de angajati este ' || V_TABELA.COUNT);
END;
```

Script Output Task completed in 0,395 seconds

Messages - Log Messages Statements Logging Page

SQL History Filter

Line 134 Column 58 | Insert | Modified | Windows: Cf

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD TEMA3.sql

SQL Worksheet History 0,39500001 seconds RotaruTeodorSGBD

Worksheet Query Builder

```
--SA SE INTRODUCĂ TOTI ANGAJATII ÎNTR-UN TABLOU INDEXAT, SA SE AFISEZE ACESTI ANGAJATI SI NUMARUL TOTAL DE ANGAJATI
SET SERVEROUTPUT ON
DECLARE
    TYPE ROW_NUM IS TABLE OF ANGAJATI_FILME.NUME%TYPE INDEX BY PLS_INTEGER;
    V_TABELA ROW_NUM;
    INDEX_TABELA ANGAJATI_FILME.ID_ANGAJAT%TYPE := 1;
    INDEX_FINAL ANGAJATI_FILME.ID_ANGAJAT%TYPE;
BEGIN
    SELECT MAX(ID_ANGAJAT) INTO INDEX_FINAL FROM ANGAJATI_FILME;
    LOOP
        SELECT NUME INTO V_TABELA(INDEX_TABELA) FROM ANGAJATI_FILME WHERE ID_ANGAJAT = INDEX_TABELA;
        INDEX_TABELA := INDEX_TABELA+1;
END;
```

Script Output x Task completed in 0,395 seconds

```
Numele angajatului cu id-ul 1 este Constantin
Numele angajatului cu id-ul 2 este Marcus
Numele angajatului cu id-ul 3 este Gheorghita
Numele angajatului cu id-ul 4 este Avram
Numele angajatului cu id-ul 5 este Topor
Numele angajatului cu id-ul 6 este Clopot
Numele angajatului cu id-ul 7 este Jean
Numarul total de angajati este 7

PL/SQL procedure successfully completed.
```

SQL History Messages - Log
Messages Statements Logging Page

Line 134 Column 58 | Insert | Modified | Windows: Cf

Exercitiul 21:

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connec... x | RotaruTeodorSGBD.sql x TEM43.sql x

SQL Worksheet History

Worksheet Query Builder

```
--SA SE INTRODUCĂ TOTI ANGAJATII INTR-UN TABLOU INDEXAT CU TIPUL RANDURILOR DIN TABELA DE ANGAJATI, SA SE AFISEZE ACESTI ANGAJATI SI NUMARUL TOTAL DE ANGAJATI
SET SERVEROUTPUT ON
DECLARE
    TYPE ROW_NUM IS TABLE OF ANGAJATI_FILME%ROWTYPE INDEX BY PLS_INTEGER;
    V_TABELA ROW_NUM;
    INDEX_TABELA ANGAJATI_FILME.ID_ANGAJAT%TYPE := 1;
    INDEX_FINAL ANGAJATI_FILME.ID_ANGAJAT%TYPE;
BEGIN
    SELECT MAX(ID_ANGAJAT) INTO INDEX_FINAL FROM ANGAJATI_FILME;
    LOOP
        SELECT * INTO V_TABELA(INDEX_TABELA) FROM ANGAJATI_FILME WHERE ID_ANGAJAT = INDEX_TABELA;
        INDEX_TABELA := INDEX_TABELA+1;
        EXIT WHEN INDEX_TABELA > INDEX_FINAL;
    END LOOP;

    FOR INDEX_TABELA IN V_TABELA.FIRST..V_TABELA.LAST LOOP
        IF V_TABELA.EXISTS(INDEX_TABELA) THEN
            DBMS_OUTPUT.PUT_LINE('Numele angajatului cu id-ul ' || INDEX_TABELA || ' este ' || V_TABELA(INDEX_TABELA).NUME);
            DBMS_OUTPUT.PUT_LINE('Salariul acestuia este de ' || V_TABELA(INDEX_TABELA).SALARIUL || ' RON');
        END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Numarul total de angajati este ' || V_TABELA.COUNT);
END;
```

Script Output x

Task completed in 0,385 seconds

Salariul acestuia este de 2400 RON

Database Schema

SQL History

Messages - Log

Filter Messages Statements Logging Page

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

Line 182 Column 14 | Insert | Modified | Windows: Cf

Oracle SQL Developer : D:\fac anul 2 sem 2\SGBD Oracle\TEMA 3\TEMA3.sql

File Edit View Navigate Run Source Team Tools Window Help

Connec... x RotaruTeodorSGBD.sql x TEMA3.sql x

SQL Worksheet History

Oracle Connection RotaruTeodorSGBD

Tables (F) RotaruTeodor

Views

Indexes

Package

Procedure

Function

Operator

Queues

Queues

Triggers

Types

Sequences

Materialized Views

Materialized Views

Synonym

Public Synonyms

Databases

Public Database

Directories

Editions

Applications

Java

XML Schemas

XML DB Functions

OLAP Objects

Schedules

Recycle Bin

Other Us

Oracle NoSQL Configuration

Database Schema

Worksheet Query Builder

```
--SA SE INTRODUCĂ TOTI ANGAJATII INTR-UN TABLOU INDEXAT CU TIPUL RANDURILOR DIN TABELA DE ANGAJATI, SA SE AFISEZE ACESTI ANGAJATI SI NUMARUL TOTAL DE ANGAJATI
SET SERVEROUTPUT ON
DECLARE
    TYPE ROW_NUM IS TABLE OF ANGAJATI_FILME%ROWTYPE INDEX BY PLS_INTEGER;
    V_TABELA ROW_NUM;
    INDEX TABELA ANGAJATI FILME.ID ANGAJAT%TYPE := 1;
```

Script Output x

Task completed in 0,385 seconds

Numele angajatului cu id-ul 1 este Constantin
Salariul acestuia este de 2400 RON
Numele angajatului cu id-ul 2 este Marcus
Salariul acestuia este de 3500 RON
Numele angajatului cu id-ul 3 este Gheorghita
Salariul acestuia este de 2300 RON
Numele angajatului cu id-ul 4 este Avram
Salariul acestuia este de 2300 RON
Numele angajatului cu id-ul 5 este Topor
Salariul acestuia este de 2200 RON
Numele angajatului cu id-ul 6 este Clopot
Salariul acestuia este de 2200 RON
Numele angajatului cu id-ul 7 este Jean
Salariul acestuia este de 4000 RON
Numarul total de angajati este 7

PL/SQL procedure successfully completed.

SQL History x

Messages - Log x

Filter Messages Statements Logging Page

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

Line 182 Column 14 | Insert | Modified | Windows: Cf

Exercitiul 22:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays the 'Connections' tree, which is expanded to show the 'Tables (Filtered)' section under 'RotaruTeodorSGBD'. The tables listed include ACTORI, ANGAJATI, ANGAJATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUȚIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII, REGIUNI, and TARI. Below the connections tree are sections for Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, and Triggers.

The central workspace contains two tabs: 'Tema4.sql' (active) and 'RotaruTeodorSGBD.sql'. The 'Tema4.sql' tab displays a PL/SQL block:

```
--SA SE AFISEZE FIECARE ANGAJAT CARE ARE SALARIUL MAXIM AL FUNCTIEI SALE

SET SERVEROUTPUT ON
DECLARE
    CURSOR INFO_ANG IS SELECT ANGAJATI_FILME.ID_ANGAJAT, ANGAJATI_FILME.NUME, ANGAJATI_FILME.PRENUME,
                           FROM ANGAJATI_FILME JOIN FUNCTII_FILME USING(ID_FUNCTIE) /*ON ANGAJATI_FILME.ID_FUNCTIE = FUNCTII_FILME.ID_FUNCTIE*/
                           WHERE ANGAJATI_FILME.SALARIUL = FUNCTII_FILME.SALARIUL_MAX;
BEGIN
    FOR ANGAJAT IN INFO_ANG LOOP
        DBMS_OUTPUT.PUT_LINE('Angajat ' || ANGAJAT.ID_ANGAJAT || ':' || ANGAJAT.NUME || ' ' || ANGAJAT.PRENUME);
    END LOOP;
END;
```

The 'Script Output' panel at the bottom shows the results of the execution:

```
Task completed in 0,26 seconds
Angajat 1: Constantin Rares

PL/SQL procedure successfully completed.
```

The bottom right corner of the interface shows status information: Line 90 Column 5, Insert, Modified, and Windows: Cf.

Exercitiul 23:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays the 'Connections' tree, which is expanded to show the 'RotaruTeodorSGBD' database connection and its tables: ACTORI, ANGAJATI, ANGAJATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUTIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, and RECENZII. The main workspace contains a 'Worksheet' tab with the following PL/SQL code:

```
--SA SE AFISEZE TOTI ANGAJATII CARE AU NUMAR DE TELEFON APARTINAND UNEI TELEFONII MOBILE ROMANESTI

SET SERVEROUTPUT ON
DECLARE
    CURSOR INFO_ANG IS SELECT ID_ANAJAT, NUME, PRENUME FROM ANGAJATI_FILME WHERE TELEFON_ANAJAT LIKE '07%' OR TELEFON_ANAJAT LIKE '407%';
BEGIN
    FOR ANAJAT IN INFO_ANG LOOP
        DBMS_OUTPUT.PUT_LINE('Angajat ' || ANAJAT.ID_ANAJAT || ':' || ANAJAT.NUME || ' ' || ANAJAT.PRENUME);
    END LOOP;
END;
```

The 'Script Output' window below the worksheet shows the results of the execution:

```
Angajat 1: Constantin Rares
Angajat 2: Marcus Stefan
Angajat 3: Gheorghita Vlad
Angajat 4: Avram Diana
Angajat 5: Topor Ana
Angajat 6: Clopot Marius
Angajat 7: Jean Mohammed
```

At the bottom of the output window, it says "PL/SQL procedure successfully completed."

The bottom status bar indicates "Line 74 Column 5" and "Windows: Cf".

Exercitiul 24:

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql Tema4.sql

SQL Worksheet History 3,11999989 seconds RotaruTeodorSGBD Components

Worksheet Query Builder

```
--SA SE AFISEZE TOATE FILMELE CU MEDIA RECENZIILOR MAI MARE DECAT UN MINIM INTRODUS DE UTILIZATOR
SET SERVEROUTPUT ON

DECLARE
    CURSOR INFO_FILME(MINIM_RECENZIE NUMBER) IS SELECT FILME.DENUMIRE_FILM,
        FROM FILME JOIN RECENZII ON FILME.ID_FILM = RECENZII.ID_FILM GROUP BY FILME.DENUMIRE_FILM HAVING AVG(RECENZII.NOTA) > MINIM_RECENZIE;
BEGIN
    FOR FILM IN INFO_FILME(MINIM_RECENZIE) LOOP
        DBMS_OUTPUT.PUT_LINE('Filmul ' || FILM.DENUMIRE_FILM);
    END LOOP;
END;
```

Script Output Task completed in 3,12 seconds

```
END;
Filmul Cardinalul
Filmul Insidious
Filmul Gladiator
Filmul Bloodshot
Filmul James Bond
Filmul Schidlers List

PL/SQL procedure successfully completed.
```

SQL History Filter Messages - Log Messages Statements Logging Page

| Line 106 Column 1 | Insert | Modified | Windows: Cf

Exercitiul 25:

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql Tema4.sql ANGAJATI_FILME

SQL Worksheet History 0,421 seconds RotaruTeodorSGBD Components

Worksheet Query Builder

```
--SA SE CREEZE TABELA "ANGAJATI_PREMIATI_FILMME" CARE CONTINE ANGAJATII CARE AU VANDUT FILME DE CEL PUTIN 50 RON FOLOSIND CURSORI
CREATE TABLE ANGAJATI_PREMIATI_FILMME AS (SELECT ID_ANGAJAT, NUME, PRENUME FROM ANGAJATI WHERE 1=2);
SET SERVEROUTPUT ON
DECLARE
    CURSOR ANGAJATI_PREMIATI_CURSOR IS SELECT ANGAJATI_FILME.ID_ANGAJAT, ANGAJATI_FILME.NUME, ANGAJATI_FILME.PRENUME
        FROM ANGAJATI_FILME WHERE ANGAJATI_FILME.ID_ANGAJAT IN
            (SELECT INCHIRIERI.ID_ANGAJAT FROM INCHIRIERI JOIN RAND_INCHIRIERI ON INCHIRIERI.ID_INCHIRIERE = RAND_INCHIRIERI.ID_INCHIRIERE
             GROUP BY INCHIRIERI.ID_ANGAJAT HAVING SUM(RAND_INCHIRIERI.PRET*RAND_INCHIRIERI.ZILE_INCHIRIATE) > 50);
BEGIN
    FOR ANGAJAT IN ANGAJATI_PREMIATI_CURSOR LOOP
        INSERT INTO ANGAJATI_PREMIATI_FILMME VALUES ANGAJAT;
        DBMS_OUTPUT.PUT_LINE('Angajatul ' || ANGAJAT.ID_ANGAJAT || ':' || ANGAJAT.NUME || ' ' || ANGAJAT.PRENUME || ' a fost adaugat in tabela');
    END LOOP;
END;

SELECT * FROM ANGAJATI_PREMIATI_FILMME;
```

Script Output x Task completed in 0,421 seconds

Table ANGAJATI_PREMIATI_FILMME created.

Angajatul 4: Avram Diana a fost adaugat in tabela
Angajatul 3: Gheorghita Vlad a fost adaugat in tabela

PL/SQL procedure successfully completed.

SQL History Messages - Log Filter Messages Statements Logging Page

| Line 54 Column 5 | Insert | Modified | Windows: Cf

Exercitiul 26:

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has various icons for connection management, file operations, and code navigation.

The left sidebar displays the database schema under "RotaruTeodorSGBD", specifically the "Tables" section, listing ACTORI, ANGAJA, ANGAJA, CLIENTI, CLIENTI, COMENZ, CRITICI, DEPART, DISTRIB, FILME, FUNCTII, FUNCTII, INCHIRI, ISTORIC, ISTORIC, LOCATII, PRODUS, RAND_C, RAND_II, RECENZ, REGIUNI, TARI, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, and Queues Tabl.

The main workspace contains a "Worksheet" tab where a PL/SQL script is being run. The script retrieves information about films that have been rented at least 3 times. The output window shows the results and a message indicating the procedure was successfully completed.

```
--Sa se afiseze toate filmele care au fost comandate de cel putin 3 ori
SET SERVEROUTPUT ON
DECLARE
    CURSOR CURSOR_FILME IS SELECT F.ID_FILM, F.DENUMIRE_FILM, F.PRET_INCHIRIERE FROM FILME F WHERE 2 < (SELECT COUNT(*) FROM RAND_INCHIRIERI RI WHERE RI.ID_FILM = F.ID_FILM);
    FILME_INC_NR_ORI CURSOR_FILME%ROWTYPE;
BEGIN
    OPEN CURSOR_FILME;
    LOOP
        FETCH CURSOR_FILME INTO FILME_INC_NR_ORI;
        EXIT WHEN CURSOR_FILME%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(' -- ID film: ' || FILME_INC_NR_ORI.ID_FILM || ' -- Denumire film: ' || FILME_INC_NR_ORI.DENUMIRE_FILM || ' -- Pret inchiriere: ' || FILME_INC_NR_ORI.PRET_INCHIRIERE);
    END LOOP;
    CLOSE CURSOR_FILME;
END;
```

Script Output x | Task completed in 0,351 seconds

```
-- ID film: 1 -- Denumire film: Insidious -- Pret inchiriere: 2,99
-- ID film: 2 -- Denumire film: James Bond -- Pret inchiriere: 2,99
-- ID film: 3 -- Denumire film: Cabin in the woods -- Pret inchiriere: 2,19
-- ID film: 4 -- Denumire film: A walk to remember -- Pret inchiriere: 2,59
-- ID film: 17 -- Denumire film: Schidlers List -- Pret inchiriere: 2,19

PL/SQL procedure successfully completed.
```

SQL History x | Filter

Messages - Log x | Filter

Messages Logging Page Statements

| Line 38 Column 5 | Insert | Modified | Windows: C|

Exercitiul 27:

The screenshot shows the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar has icons for New, Open, Save, Run, Stop, and Refresh. The Connections sidebar lists various database objects: TELEFON, EMAIL_CLIENT, SEX, PREMIUM, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUTIE, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI (with sub-items ID_INCHIRIERE, ID_CLIENT, DATA, ID_ANGAJAT), ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI (with sub-items ID_INCHIRIERE, ID_FILM, ZILE_INCHIRIATE, DATA_INCHIRIERE, DATA_RETURNARE, PRET), RECENZII, REGIUNI, and TARI. The current tab is "Tema4.sql" under the "SQL Worksheet" tab. The code in the worksheet is:

```
--SA SE AFISEZE TOTI CLIENTII ORDONATI DESCRESATOR DUPA VARSTA SI TOATE INFORMATIILE DESPRE INCHIRIERILE ACESTORA FOLOSING CURSORI IMBRICATI
SET SERVEROUTPUT ON
DECLARE
    CURSOR INFO_CLIENTI IS SELECT * FROM CLIENTI_FILME ORDER BY VARSTA DESC;
    CURSOR INFO_INCHIRIERI(ID_CLI CLIENTI_FILME.ID_CLIENT%TYPE) IS SELECT INCHIRIERI.ID_ANGAJAT, RAND_INCHIRIERI.*
    FROM INCHIRIERI JOIN RAND_INCHIRIERI ON INCHIRIERI.ID_INCHIRIERE = RAND_INCHIRIERI.ID_INCHIRIERE
    WHERE INCHIRIERI.ID_CLIENT = ID_CLI;
BEGIN
    FOR CLIENT IN INFO_CLIENTI LOOP
        DBMS_OUTPUT.PUT_LINE('Clientul: ' || CLIENT.NUME || ' ' || CLIENT.PRENUME);
        FOR INCHIRIERE IN INFO_INCHIRIERI(CLIENT.ID_CLIENT) LOOP
            DBMS_OUTPUT.PUT_LINE('      Comanda ' || INCHIRIERE.ID_INCHIRIERE ||
                ', Data ' || INCHIRIERE.DATA_INCHIRIERE ||
                ', Valoare linie ' || INCHIRIERE.PRET*INCHIRIERE.ZILE_INCHIRIATE);
        END LOOP;
    END LOOP;
END;
```

The "Script Output" window below the worksheet shows the message: "PL/SQL procedure successfully completed." The status bar at the bottom indicates "Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql".

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql Tema4.sql

TELEFON EMAIL_CLIENT SEX PREMIUM COMENZI CRITICI DEPARTAMENTE DISTRIBUTIE FILME FUNCTII FUNCTII_FILME INCHIRIERI ID_INCHIRIERE ID_CLIENT DATA ID_ANGAJAT ISTORIC_FUNCTII ISTORIC_FUNCTII_FILME LOCATII PRODUSE RAND_COMENZI RAND_INCHIRIERI ID_INCHIRIERE ID_FILM ZILE_INCHIRIATE DATA_INCHIRIERE DATA_RETURNARE PRET RECENZII REGIUNI TARI Views Indexes

SQL Worksheet History Worksheet Query Builder Script Output Task completed in 0,683 seconds

Comanda 12, Data 24-12-2019, Valoare linie 3,98
Clientul: Mihai Topor
Comanda 3, Data 11-09-2019, Valoare linie 3,87
Comanda 6, Data 10-07-2020, Valoare linie 7,98
Comanda 6, Data 10-07-2020, Valoare linie 11,97
Comanda 8, Data 11-10-2020, Valoare linie 3,99
Clientul: Valentin Afrim
Comanda 11, Data 24-12-2019, Valoare linie 6,57
Clientul: Mihaela Chirila
Comanda 19, Data 31-12-2020, Valoare linie 20,93
Clientul: Rotaru Teodor-Gabriel
Comanda 1, Data 11-12-2019, Valoare linie 8,97
Comanda 1, Data 11-12-2019, Valoare linie 6,57
Comanda 1, Data 11-12-2019, Valoare linie 8,76
Comanda 2, Data 01-12-2019, Valoare linie 18,13
Comanda 2, Data 01-12-2019, Valoare linie 6,93
Comanda 4, Data 11-01-2020, Valoare linie 6,93
Comanda 9, Data 12-10-2020, Valoare linie 11,97
Clientul: Irina Damaschin
Comanda 23, Data 12-12-2019, Valoare linie 1,99

PL/SQL procedure successfully completed.

SQL History Filter Messages - Log Messages Statements Logging Page

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql | Line 118 Column 5 | Insert | Modified | Windows: Cf

Exercitiul 28:

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

- Title Bar:** Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql
- File Menu:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Run, Stop, Refresh, and others.
- Connections Sidebar:** Shows the current connection is RotaruTeodorSGBD, with options for Tables, Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, Sequences, Materialized Views, Materialized View Logs, Synonyms, Public Synonyms, Database Links, Public Database Links, Directories, Editions, Application Express, Java, XML Schemas, XML DB Repository, OLAP Option, Scheduler, Recycle Bin, and Other Users.
- SQL Worksheet Tab:** The tab is titled "Tema4.sql" and contains the following PL/SQL code:

```
--SA SE SCADA CU 10% PRETURILE TUTUROR FILMELOR CARE NU AU FOST COMANDATE MAI MULT DE 2 ORI
SET SERVEROUTPUT ON
DECLARE
    CURSOR INFO_FILME IS SELECT * FROM FILME
    WHERE FILME.ID_FILM NOT IN (SELECT RAND_INCHIRIERI.ID_FILM FROM RAND_INCHIRIERI GROUP BY RAND_INCHIRIERI.ID_FILM HAVING COUNT(*) > 2)
    FOR UPDATE OF FILME.PRET_INCHIRIERE;
BEGIN
    FOR FILM IN INFO_FILME LOOP
        UPDATE FILME SET PRET_INCHIRIERE = 0.9*PRET_INCHIRIERE;
        DBMS_OUTPUT.PUT_LINE('Pretul filmului ' || FILM.DENUMIRE_FILM || ' a fost modificat');
    END LOOP;
END;
COMMIT;
```
- Script Output Tab:** Displays the output of the executed procedure, showing the modified titles of six movies:

```
Pretul filmului Warrior a fost modificat
Pretul filmului Saw a fost modificat
Pretul filmului The Mechanic a fost modificat
Pretul filmului Profu a fost modificat
Pretul filmului Animal Crackers a fost modificat
Pretul filmului Bloodshot a fost modificat
Pretul filmului A haunted house a fost modificat
```

PL/SQL procedure successfully completed.
- Bottom Status Bar:** Shows "Line 121 Column 92 | Insert | Modified | Windows: Cf"

Exercitiul 29:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema for 'RotaruTeodorSGBD' with various tables like ACTORI, ANGAJATI, and FUNCTII_FILME. The central workspace contains a SQL Worksheet tab with the following PL/SQL code:

```
--SA SE STEARGA TOATE FUNCTIILE DIN ISTORIC DACA SUNT MAI VECHI DE UN AN

SET SERVEROUTPUT ON
DECLARE
BEGIN
    DELETE FROM ISTORIC_FUNCTII_FILME IFF WHERE SYSDATE - IFF.DATA_SFARSIT > 365;
    DBMS_OUTPUT.PUT_LINE('Au fost sterse ' || SQL%ROWCOUNT || ' functii');
END;
rollback;
```

The 'Script Output' panel at the bottom shows the execution results:

- Rollback complete.
- Rollback complete.
- Au fost sterse 1 functii
- PL/SQL procedure successfully completed.

The status bar at the bottom right indicates 'Line 8 Column 5' and other window controls.

Exercitiul 30:

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema4.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql Tema4.sql

SQL Worksheet History 3,5859997 seconds RotaruTeodorSGBD

Worksheet Query Builder

```
--Sa se actualizeze salariul angajatului cu numele "John Smith" cu o suma primita de la utilizator
--In cazul in care nu exista acest angajat, sa se afiseze un mesaj corespunzator
SET SERVEROUTPUT ON
DECLARE
BEGIN
    UPDATE ANGAJATI_FILME AF SET AF.SALARIUL = &SALARIU_NOU WHERE AF.NUME LIKE 'John' AND AF.PRENUME LIKE 'Smith';
    IF SQL%NOTFOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista acest angajat');
    ELSE
        DBMS_OUTPUT.PUT_LINE('Salariul a fost actualizat cu succes');
    END IF;
END;
```

Script Output Task completed in 3,586 seconds

```
IF SQL%NOTFOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista acest angajat');
ELSE
    DBMS_OUTPUT.PUT_LINE('Salariul a fost actualizat cu succes');
END IF;
END;
Nu exista acest angajat

PL/SQL procedure successfully completed.
```

SQL History Messages - Log Filter Messages Logging Page Statements

Upper Keywords, Lower Identifiers Line 22 Column 5 Insert Modified Windows: Cf

Exercitiul 31:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the connection tree under 'RotaruTeodorSGBD' (Tables, Views, Indexes, Procedures, Functions, Operators, Queues, Sequences, Materialized Views, Materialized View Logs, Synonyms, Public Synonyms, Database Links, Public Database Links, Directories, Editions, Application Express, Java, XML Schemas, XML DB Repository, OLAP Option, Scheduler, Recycle Bin, Other Users). The main workspace contains a 'Worksheet' tab with the following PL/SQL code:

```
--SA SE AFISEZE TOATE FILMELE IN LIMBA CHINEZA. SA SE TRATEZE POSIBILELE EXCEPTII
SET SERVEROUTPUT ON
DECLARE
    CURSOR FILME_CURSOR IS SELECT DENUMIRE_FILM FROM FILME WHERE UPPER(LIMBA_FILM) LIKE 'CHINEZA';
    FILME_CHINEZA FILME.DENUMIRE_FILM%TYPE;
BEGIN
    SELECT DENUMIRE_FILM INTO FILME_CHINEZA FROM FILME WHERE UPPER(LIMBA_FILM) LIKE 'CHINEZA';
EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista niciun film in chineza');
    WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE('Eroare...nu se pot asa mai multe linii');
END;
```

The 'Script Output' window below shows the results of running the procedure:

```
Nu exista niciun film in chineza
PL/SQL procedure successfully completed.
```

The bottom status bar indicates 'Line 12 Column 5' and 'Modified'.

Exercitiul 32:

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The left sidebar displays the 'Connections' tree, which is expanded to show 'RotaruTeodorSGBD' and its various schema objects like Tables, Views, Indexes, etc. The main workspace contains two tabs: 'RotaruTeodorSGBD.sql' and 'Tema 5.sql'. The 'Tema 5.sql' tab is active and contains the following PL/SQL code:

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR FILME_CURSOR IS SELECT DENUMIRE_FILM FROM FILME WHERE UPPER(LIMBA_FILM) LIKE 'CHINEZA';
BEGIN
    OPEN FILME_CURSOR;
    FOR FILM IN FILME_CURSOR LOOP
        DBMS_OUTPUT.PUT_LINE(FILM.DENUMIRE_FILM);
    END LOOP;

    EXCEPTION
        WHEN CURSOR_ALREADY_OPEN THEN DBMS_OUTPUT.PUT_LINE('Eroare...cursorul a fost deschis de doua sau mai multe ori');
END;
```

The 'Script Output' window at the bottom shows the results of the execution:

```
Eroare...cursorul a fost deschis de doua sau mai multe ori
PL/SQL procedure successfully completed.
```

The status bar at the bottom right indicates 'Line 26 Column 5 | Insert | Modified | Windows: Cf'.

Exercitiul 33:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays a tree view of database tables under 'Tables (Filtered)'. The 'FILME' table is expanded, showing columns: ID_FILM, DENUMIRE_FILM, CATEGORIE_FILM, LIMBA_FILM, DURATA_FILM_MINUTE, DATA_LANSARE, PRET_FURNIZOR, PRET_INCHIRIERE, and STOC.

The central workspace contains a 'Worksheet' tab with the following PL/SQL code:

```
--SA SE INSEREZE IN TABELA FILME UN FILM CU ID-UL DEJA EXISTENT. SA SE TRATEZE POSIBILELE EXCEPTII
SET SERVEROUTPUT ON
DECLARE
    EXCEPTIE_CHEIE_UNICA EXCEPTION;
    PRAGMA EXCEPTION_INIT(EXCEPTIE_CHEIE_UNICA, -00001);
BEGIN
    INSERT INTO FILME VALUES(10, 'John Doe', 'Drama', 'Engleza', 120, SYSDATE, 19.99, 1.59, 100);
    EXCEPTION
        WHEN EXCEPTIE_CHEIE_UNICA THEN DBMS_OUTPUT.PUT_LINE('Acest id deja exista');
END;
```

The 'Script Output' panel at the bottom shows the execution results:

```
Acest id deja exista

PL/SQL procedure successfully completed.
```

The status bar at the bottom right indicates: Line 39 Column 5, Insert, Modified, Windows-CE.

Exercitiul 34:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the database schema with various tables like ACTORI, ANGAJATI, FILME, etc. The central workspace contains a SQL Worksheet with the following PL/SQL code:

```
--SA SE STEARGA FILMELE CU ID-UL NUMAR PAR. IN CAZUL ERORILOR, SA FIE ADAUGATE INTR-O TABELA 'ERORI'  
DROP TABLE erori CASCADE CONSTRAINTS;  
CREATE TABLE ERORI(  
    UTILIZATOR VARCHAR2(150),  
    DATA_DATE,  
    COD_EROARE NUMBER,  
    MESAJ_EROARE VARCHAR2(200)  
);  
SET SERVEROUTPUT ON  
DECLARE  
    COD_EROARE NUMBER;  
    MESAJ VARCHAR2(200);  
EXCEPTIE_STERGERE EXCEPTION;  
    PRAGMA EXCEPTION_INIT(EXCEPTIE_STERGERE, -2292);  
BEGIN  
    DELETE FROM FILME WHERE MOD(FILME.ID_FILM, 2) = 0;  
    EXCEPTION  
    WHEN EXCEPTIE_STERGERE THEN DBMS_OUTPUT.PUT_LINE('Nu puteti sterge filmul');  
    COD_EROARE := SQLCODE;  
    MESAJ := SQLERRM;  
    INSERT INTO ERORI VALUES(USER, SYSDATE, COD_EROARE, MESAJ);  
END;
```

The script output window below shows the message "Table ERORI created." and indicates the task completed in 0,378 seconds.

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections RotaruTeodorSGBD.sql Tema 5.sql ERORI

SQL Worksheet History 0,3779999 seconds RotaruTeodorSGBD

Components

Tables (Filtered)

- ACTORI
- ANGAJATI
- ANGAJATI_FILME
- ANGAJATI_PREMIATI_FILMME
- CLIENTI
- CLIENTI_FILME
- COMENZI
- CRITICI
- DEPARTAMENTE
- DISTRIBUTIE
- ERORI
- FILME
- FUNCTII
- FUNCTII_FILME
- INCHIRIERI
- ISTORIC_FUNCTII
- ISTORIC_FUNCTII_FILME
- LOCATII
- PRODUSE
- RAND_COMENZI
- RAND_INCHIRIERI
- RECENTII
- REGIUNI
- TARI

Views

Indexes

Packages

Procedures

Functions

Operators

--SA SE STEARGA FILMELE CU ID-UL NUMAR PAR. IN CAZUL ERORILOR, SA FIE ADAUGATE INTR-O TABELA 'ERORI'

DROP TABLE erori CASCADE CONSTRAINTS;

CREATE TABLE ERORI(
 UTILIZATOR VARCHAR2(150),
 DATA_DATE,
 COD_EROARE NUMBER,
 MESAJ_EROARE VARCHAR2(200)
)
SET SERVEROUTPUT ON

DECLARE
 COD_EROARE NUMBER;
 MESAJ VARCHAR2(200);

EXCEPTIE_STERGERE EXCEPTION;
PRAGMA EXCEPTION_INIT(EXCEPTIE_STERGERE, -2292);

BEGIN

Table ERORI created.

Nu puteti sterge filmul

PL/SQL procedure successfully completed.

SQL History

Messages - Log Filter Messages Logging Page Statements

Line 60 Column 22 | Insert | Modified | Windows: CI

Exercitiul 35:

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

- Title Bar:** Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql
- Menu Bar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help
- Toolbars:** Standard toolbar with icons for Open, Save, Print, etc.
- Connections Sidebar:** Shows "RotaruTeodorSGBD" connection and its tables: ACTORI, ANGAJATI, ANGAJATI_FILME, ANGAJATI_PREMIATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUIE, ERORI, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII, REGIUNI, TARI, Views, Indexes, Packages, Procedures, Functions, Operators.
- SQL Worksheet Tab:** RotaruTeodorSGBD.sql (selected), Tema 5.sql
- Worksheet Area:** Displays a PL/SQL script:

```
--SA SE STEARGA TOATE INCHIRIERILE CLIENTILOR MAI VECHI DE UN AN. SA NU SE STEARGA DELOC INCHIRIERILE CARE AU TOTALUL PER CLIENT  
--MAI MARE DE 100 LEI, INDIFERENT DE DATA ACESTORA  
CREATE OR REPLACE VIEW INCHIRIERI_100 AS  
SELECT INCHIRIERI.ID_CLIENT AS "ID_CL", SUM(RAND_INCHIRIERI.PRET*RAND_INCHIRIERI.ZILE_INCHIRIATE) AS "SUMA"  
FROM INCHIRIERI JOIN RAND_INCHIRIERI ON  
INCHIRIERI.ID_INCHIRIERE = RAND_INCHIRIERI.ID_INCHIRIERE  
GROUP BY INCHIRIERI.ID_CLIENT  
HAVING SUM(RAND_INCHIRIERI.PRET*RAND_INCHIRIERI.ZILE_INCHIRIATE) > 100;  
  
SET SERVEROUTPUT ON  
DECLARE  
    EXCEPTIE_COMANDA EXCEPTION;  
BEGIN  
    DELETE (SELECT * FROM RAND_INCHIRIERI JOIN INCHIRIERI ON RAND_INCHIRIERI.ID_INCHIRIERE = INCHIRIERI.ID_INCHIRIERE  
    WHERE SYSDATE - RAND_INCHIRIERI.DATA_INCHIRIERE > 365 AND INCHIRIERI.ID_CLIENT NOT IN  
    (SELECT INCHIRIERI_100.ID_CL FROM INCHIRIERI_100));  
  
    IF SQL%NOTFOUND THEN  
        RAISE EXCEPTIE_COMANDA;  
    END IF;  
  
    EXCEPTION  
    WHEN EXCEPTIE_COMANDA THEN DBMS_OUTPUT.PUT_LINE('Nu a fost gasit nicio comanda de sters');  
END;
```
- Script Output Window:** Task completed in 0,279 seconds
- SQL History Window:**
- Messages - Log Window:** Messages, Logging Page, Statements

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections x | RotaruTeodorSGBD.sql x Tema 5.sql x

SQL Worksheet History

Worksheet Query Builder

```
SET SERVEROUTPUT ON
DECLARE
    EXCEPTIE_COMANDA EXCEPTION;
BEGIN
    DELETE (SELECT * FROM RAND_INCHIRIERI JOIN INCHIRIERI ON RAND_INCHIRIERI.ID_INCHIRIERE = INCHIRIERI.ID_INCHIRIERE
            WHERE SYSDATE - RAND_INCHIRIERI.DATA_INCHIRIERE > 365 AND INCHIRIERI.ID_CLIENT NOT IN
            (SELECT INCHIRIERI_100.ID_CL FROM INCHIRIERI_100));

    IF SQL%NOTFOUND THEN
        RAISE EXCEPTIE_COMANDA;
    END IF;

    EXCEPTION
        WHEN EXCEPTIE_COMANDA THEN DBMS_OUTPUT.PUT_LINE('Nu a fost gasit nicio comanda de sters');
    END;
    ROLLBACK;
```

Script Output x

Task completed in 0,279 seconds

View INCHIRIERI_100 created.

PL/SQL procedure successfully completed.

SQL History x | Filter

Messages - Log

Messages | Logging Page | Statements

Components

Exercitiul 37:

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections RotaruTeodorSGBD

Tables (Filtered)

- ACTORI
- ANGAJATI
- ANGAJATI_FILME
- ANGAJATI_PREMIATI_FILM
- CLIENTI
- CLIENTI_FILME
- COMENZI
- CRITICI
- DEPARTAMENTE
- DISTRIBUȚIE
- ERORI
- FILME
- FUNCTII
- FUNCTII_FILME
- INCHIRIERI
- ISTORIC_FUNCTII
- ISTORIC_FUNCTII_FILME
- LOCATII
- PRODUSE
- RAND_COMENZI
- RAND_INCHIRIERI

 - ID_INCHIRIERE
 - ID_FILM
 - ZILE_INCHIRIATE
 - DATA_INCHIRIERE
 - DATA_RETURNARE
 - PRET

- RECENTII
- REGIUNI
- TARI

Tema 5.sql x

SQL Worksheet History

Worksheet Query Builder

```
--SA SE CREEZE O PROCEDURA PENTRU MODIFICAREA PRETULUI DE INCHIRIERE A UNUI FILM CU UN NOU PRET
--SA SE FOLOSEACA PROCEDURA PENTRU A ACTUALIZA PRETUL TUTUROR FILMELOR INCHIRIATE DE MAI PUTIN DE 3 ORI, SCAZAND 1 LEU DIN PRETUL CURENT

CREATE OR REPLACE
PROCEDURE MODIFICA_PRET_INCHIRIERE_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, P_PRET_INC_NOU IN FILME.PRET_INCHIRIERE%TYPE) IS
BEGIN
    UPDATE FILME SET PRET_INCHIRIERE = P_PRET_INC_NOU WHERE ID_FILM = P_ID_FILM;

    IF SQL%NOTFOUND THEN
        RAISE NO_DATA_FOUND;
    END IF;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film. Incercati alt ID');
    END;
    SET SERVEROUTPUT ON
DECLARE
    CURSOR CURSOR_FILME IS SELECT FILME.ID_FILM, FILME.PRET_INCHIRIERE
    FROM FILME WHERE 3 > (SELECT COUNT(*) FROM RAND_INCHIRIERI WHERE RAND_INCHIRIERI.ID_FILM = FILME.ID_FILM);
BEGIN
    FOR FILM IN CURSOR_FILME LOOP
        MODIFICA_PRET_INCHIRIERE_FILM(FILM.ID_FILM, FILM.PRET_INCHIRIERE-1);
    END LOOP;
END;
```

Script Output x

Task completed in 0,22 seconds

SQL History x

Compiler - Log
Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

Line 115 Column 5 | Insert | Modified | Windows: Cf

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Tema 5.sql

Oracle Connections RotaruTeodorSGBD

Tables (Filtered) ACTORI ANGAJATI ANGAJATI_FILME ANGAJATI_PREMIATI_FILME CLIENTI CLIENTI_FILME COMENZI CRITICI DEPARTAMENTE DISTRIBUTIE ERORI FILME FUNCTII FUNCTII_FILME INCHIRIERI ISTORIC_FUNCTII ISTORIC_FUNCTII_FILME LOCATII PRODUSE RAND_COMENZI RAND_INCHIRIERI ID_INCHIRIERE ID_FILM ZILE_INCHIRIATE DATA_INCHIRIERE DATA_RETURNARE PRET RECENZII REGIUNI TARI

SQL Worksheet History

Worksheet Query Builder

```
IF SQL%NOTFOUND THEN
    RAISE NO_DATA_FOUND;
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nu exista acest film. Incercati alt ID');
END;
SET SERVEROUTPUT ON
DECLARE
    CURSOR CURSOR_FILME IS SELECT FILME.ID_FILM, FILME.PRET_INCHIRIERE
        FROM FILME WHERE 3 > (SELECT COUNT(*) FROM RAND_INCHIRIERI WHERE RAND_INCHIRIERI.ID_FILM = FILME.ID_FILM);
BEGIN
    FOR FILM IN CURSOR_FILME LOOP
        MODIFICA_PRET_INCHIRIERE_FILM(FILM.ID_FILM, FILM.PRET_INCHIRIERE-1);
    END LOOP;
END;
```

Script Output x

Task completed in 0,22 seconds

Procedure MODIFICA_PRET_INCHIRIERE_FILM compiled

PL/SQL procedure successfully completed.

SQL History Compiler - Log

Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql | Line 115 Column 5 | Insert | Modified | Windows: CI

Exercitiul 38:

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

- Title Bar:** Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql
- Menu Bar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help
- Toolbars:** Standard toolbar with icons for New, Open, Save, Print, Run, Stop, Refresh, Undo, Redo, Find, Replace, and Help.
- Connections:** Shows a connection to "RotaruTeodorSGBD".
- Sidebar (Oracle Connections):** Lists various tables and objects:
 - Tables (Filtered): ACTORI, ANGAJATI, ANGAJATI_FILME, ANGAJATI_PREMIATI_FILME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUIE, ERORI, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI (selected), ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII, REGIUNI, TARI.
 - Views, Indexes.
- SQL Worksheet:** The current tab is "Worksheet". The code is as follows:

```
--SA SE CREEZE O PROCEDURA CARE PRIMESTE CA PARAMETRU UN ID DE FILM SI RETURNEAZA NUMELE ACESTUIA SI DE CATE ORI A FOST INCHIRIAT  
--PRIN PARAMETRII DE IESIRE  
CREATE OR REPLACE  
PROCEDURE AFISARE_INFO_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, NR_INC OUT NUMBER, P_DENUMIRE_FILM OUT FILME.DENUMIRE_FILM%TYPE) IS  
BEGIN  
    SELECT COUNT(*), FILME.DENUMIRE_FILM INTO NR_INC, P_DENUMIRE_FILM  
    FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM = RAND_INCHIRIERI.ID_FILM WHERE FILME.ID_FILM = P_ID_FILM GROUP BY FILME.DENUMIRE_FILM;  
  
    EXCEPTION  
        WHEN NO_DATA_FOUND THEN SELECT DENUMIRE_FILM INTO P_DENUMIRE_FILM FROM FILME WHERE FILME.ID_FILM = P_ID_FILM;  
        NR_INC := 0;  
END;  
SET SERVEROUTPUT ON  
DECLARE  
    V_ID_FLM FILME.ID_FILM%TYPE;  
    V_NR_INC NUMBER(3);  
    V_DENUMIRE_FILM FILME.DENUMIRE_FILM%TYPE;  
BEGIN  
    V_ID_FLM := &ID_FLM;  
    AFISARE_INFO_FILM(V_ID_FLM, V_NR_INC, V_DENUMIRE_FILM);  
    DBMS_OUTPUT.PUT_LINE('ID Film: ' || V_ID_FLM);  
    DBMS_OUTPUT.PUT_LINE('Denumire: ' || V_DENUMIRE_FILM);  
    DBMS_OUTPUT.PUT_LINE('Numar inchirieri ' || V_NR_INC);  
END;
```

Script Output: Task completed in 1,703 seconds

```
DBMS_OUTPUT.PUT_LINE('Numar inchirieri ' || V_NR_INC);
```

SQL History:

Compiler - Log: Messages, Logging Page, Statements, Compiler

Bottom status bar: Line 128 Column 5 | Insert | Modified | Windows: Cf

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Tema 5.sql RAND_INCHIRIERI

SQL Worksheet History RotaruTeodorSGBD

Components

Worksheet Query Builder

```
--SA SE CREEZE O PROCEDURA CARE PRIMESTE UN ID DE FILM SI RETURNEAZA NUMELE ACESTUIA SI DE CATE ORI A FOST INCHIRIAT
--PRIN PARAMETRII DE IESIRE
CREATE OR REPLACE
PROCEDURE AFISARE_INFO_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, NR_INC OUT NUMBER, P_DENUMIRE_FILM OUT FILME.DENUMIRE_FILM%TYPE) IS
BEGIN
    SELECT COUNT(*), FILME.DENUMIRE_FILM INTO NR_INC, P_DENUMIRE_FILM
    FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM = RAND_INCHIRIERI.ID_FILM WHERE FILME.ID_FILM = P_ID_FILM GROUP BY FILME.DENUMIRE_FILM;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN SELECT DENUMIRE_FILM INTO P_DENUMIRE_FILM FROM FILME WHERE FILME.ID_FILM = P_ID_FILM;
        NR_INC := 0;
END;
SET SERVEROUTPUT ON
DECLARE
    V_ID_FLM FILME.ID_FILM%TYPE;
    V_NR_INC NUMBER(3);
    V_DENUMIRE_FILM FILME.DENUMIRE_FILM%TYPE;
```

Script Output x
Task completed in 1,703 seconds

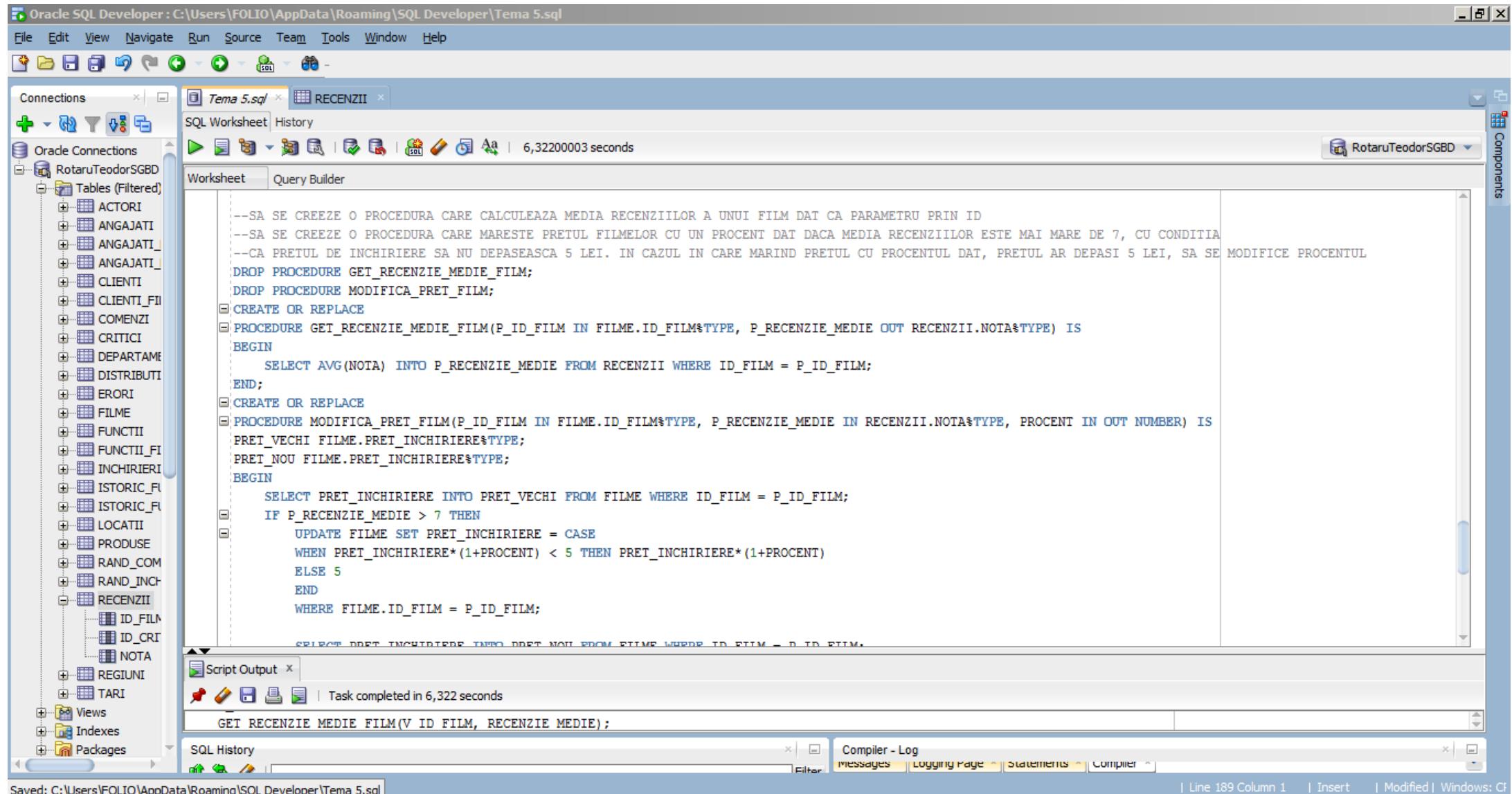
```
END;
ID Film: 4
Denumire: A walk to remember
Numar inchirieri 3

PL/SQL procedure successfully completed.
```

SQL History Compiler - Log
Messages Logging Page Statements Compiler

Line 128 Column 5 | Insert | Modified | Windows: CI

Exercitiul 39:



Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections RotaruTeodorSGBD Tables (Filtered) ACTORI ANGAJATI ANGAJATI_ ANGAJATI_ CLIENTI CLIENTI_FILM COMENZI CRITICI DEPARTAMENT DISTRIBUTII ERORI FILME FUNCTII FUNCTII_FILM INCHIRIERI ISTORIC_FILM ISTORIC_FILM LOCATII PRODUSE RAND_COM RAND_INCHIRIERI RECENZII ID_FILM ID_CRI NOTA REGIUNI TARI Views Indexes Packages

Tema 5.sql x RECENTII x SQL Worksheet History 6,32200003 seconds RotaruTeodorSGBD Components

Worksheet Query Builder

```
SELECT PRET_INCHIRIERE INTO PRET_NOU FROM FILME WHERE ID_FILM = P_ID_FILM;
DBMS_OUTPUT.PUT_LINE('Pretul vechi era de ' || PRET_VECI || ' lei');
DBMS_OUTPUT.PUT_LINE('Pretul nou este de ' || PRET_NOU || ' lei');
IF PRET_NOU = 5 THEN
    PROCENT := 1-PRET_NOU/PRET_VECI;
    DBMS_OUTPUT.PUT_LINE('Procentul a fost modificat (pretul depasea 5 lei). Procent nou = ' || ROUND(PROCENT,2));
ELSE
    DBMS_OUTPUT.PUT_LINE('Procentul nemodificat = ' || PROCENT);
END IF;
END IF;
END;
SET SERVEROUTPUT ON
DECLARE
    RECENZIE_MEDIE RECENZII.NOTA%TYPE;
    V_ID_FILM FILME.ID_FILM%TYPE;
    V_PROCENT NUMBER;
BEGIN
    V_ID_FILM := &ID_FILM;
    V_PROCENT := &PROCENT;
    GET_RECENZIE_MEDIE_FILM(V_ID_FILM, RECENZIE_MEDIE);
    MODIFICA_PRET_FILM(V_ID_FILM, RECENZIE_MEDIE, V_PROCENT);
END;
```

Script Output x Task completed in 6,322 seconds
GET RECENZIE MEDIE FILM(V ID FILM, RECENZIE MEDIE);

SQL History Compiler - Log Filter Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql | Line 189 Column 1 | Insert | Modified | Windows: CI

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Oracle Connections RotaruTeodorSGBD Tables (Filtered) ACTORI ANGAJATI ANGAJATI_ ANGAJATI_ CLIENTI CLIENTI_FILM COMENZI CRITICI DEPARTAMENT DISTRIBUTII ERORI FILME FUNCTII FUNCTII_FILM INCHIRIERI ISTORIC_FILM ISTORIC_FILM LOCATII PRODUSE RAND_COM RAND_INCHIRIERI RECENZII RECENZII ID_FILM ID_CRI NOTA REGIUNI TARI Views Indexes Packages

Tema 5.sql x RECENTII x SQL Worksheet History 6,32200003 seconds RotaruTeodorSGBD Components

Worksheet Query Builder

```
SELECT PRET_INCHIRIERE INTO PRET_NOU FROM FILME WHERE ID_FILM = P_ID_FILM;
DBMS_OUTPUT.PUT_LINE('Pretul vechi era de ' || PRET_VECHI ||' lei');
DBMS_OUTPUT.PUT_LINE('Pretul nou este de ' || PRET_NOU ||' lei');
IF PRET_NOU = 5 THEN
    PROCENT := 1-PRET_NOU/PRET_VECHI;
    DBMS_OUTPUT.PUT_LINE('Procentul a fost modificat (pretul depasea 5 lei). Procent nou = ' || ROUND(PROCENT,2));
ELSE
    DBMS_OUTPUT.PUT_LINE('Procentul nemodificat = ' || PROCENT);
END IF;
END IF;
END;
SET SERVEROUTPUT ON
DECLARE
    RECENZIE_MEDIE RECENZII.NOTA%TYPE;
```

Script Output x Task completed in 6,322 seconds

```
GET_RECENZIE_MEDIE_FILM(V_ID_FILM, RECENZIE_MEDIE);
MODIFICA_PRET_FILM(V_ID_FILM, RECENZIE_MEDIE, V_PROCENT);
END;
Pretul vechi era de 3,2 lei
Pretul nou este de 5 lei
Procentul a fost modificat (pretul depasea 5 lei). Procent nou = -,5625

PL/SQL procedure successfully completed.
```

SQL History Compiler - Log Filter Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql | Line 189 Column 1 | Insert | Modified | Windows: CI

Exercitiul 40:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes a connection to 'RotaruTeodorSGBD' containing various tables like ACTORI, ANGAJATI, ANGAJATI_FILME, etc. Below the connections are sections for Views, Indexes, Packages, Procedures, Functions, Operators, and Queues. The main workspace is titled 'Tema 5.sql' and contains a PL/SQL script. The script creates a function 'ESTE_ANGAJAT_1AN' that checks if an employee has been working for more than a year. It also updates the salary of employees whose age exceeds 364 days by 10%. The 'Script Output' pane at the bottom shows the message 'Salariul angajatului a fost modificat'. The 'Messages - Log' pane shows 'Task completed in 0,244 seconds'.

```
-- SA SE CREEZE O FUNCTIE CARE RETURNEAZA TRUE DACA UN ANGAJAT ARE MAI MULT DE UN AN VECHIME SI FALSE IN CAZ CONTRAR
-- SA SE FOLOSEASCA PENTRU A MARI SALARIUL ANGAJATILOR CU 10% DACA VECHEMIA ESTE DE PESTE UN AN
CREATE OR REPLACE
FUNCTION ESTE_ANGAJAT_1AN(P_ID_ANGAJAT IN ANGAJATI_FILME.ID_ANGAJAT%TYPE)
RETURN BOOLEAN
IS
    VECHIME NUMBER;
BEGIN
    SELECT SYSDATE-DATA_ANGAJARE INTO VECHIME FROM ANGAJATI_FILME WHERE ID_ANGAJAT = P_ID_ANGAJAT;
    IF VECHIME > 364 THEN RETURN TRUE;
    ELSE RETURN FALSE;
    END IF;
END;
SET SERVEROUTPUT ON
BEGIN
    FOR ANGAJAT IN (SELECT * FROM ANGAJATI_FILME) LOOP
        IF ESTE_ANGAJAT_1AN(ANGAJAT.ID_ANGAJAT) THEN
            UPDATE ANGAJATI_FILME SET SALARIUL = 1.1*SALARIUL WHERE ID_ANGAJAT = ANGAJAT.ID_ANGAJAT;
            DBMS_OUTPUT.PUT_LINE('Salariul angajatului ' || ANGAJAT.NUME || ' a fost modificat');
        ELSE DBMS_OUTPUT.PUT_LINE('Salariul angajatului ' || ANGAJAT.NUME || ' NU a fost modificat (vechime mai mica de un an)');
        END IF;
    END LOOP;
END;
```

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections x | Tema 5.sql x

SQL Worksheet History

Worksheet Query Builder

0,244 seconds RotaruTeodorSGBD Components

Oracle Connections RotaruTeodorSGBD

Tables (Filtered)

- ACTORI
- ANGAJATI
- ANGAJATI_FILME
- ANGAJATI_PREMIATI_FILMME
- CLIENTI
- CLIENTI_FILME
- COMENZI
- CRITICI
- DEPARTAMENTE
- DISTRIBUTIE
- ERORI
- FILME
- FUNCTII
- FUNCTII_FILME
- INCHIRIERI
- ISTORIC_FUNCTII
- ISTORIC_FUNCTII_FILME
- LOCATII
- PRODUSE
- RAND_COMENZI
- RAND_INCHIRIERI
- RECENZII
- REGIUNI
- TARI

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Tema 5.sql

-- SA SE CREEZE O FUNCTIE CARE RETURNEAZA TRUE DACA UN ANGAJAT ARE MAI MULT DE UN AN VECHIME SI FALSE IN CAZ CONTRAR
-- SA SE FOLOSEASCA PENTRU A MARI SALARIUL ANGAJATILOR CU 10% DACA VECHIMEA ESTE DE PESTE UN AN

```
CREATE OR REPLACE
FUNCTION ESTE_ANGAJAT_1AN(P_ID_ANGAJAT IN ANGAJATI_FILME.ID_ANGAJAT%TYPE)
RETURN BOOLEAN
IS
    VECHIME NUMBER;
BEGIN
    SELECT SYSDATE-DATA_ANGAJARE INTO VECHIME FROM ANGAJATI_FILME WHERE ID_ANGAJAT = P_ID_ANGAJAT;
    IF VECHIME > 364 THEN RETURN TRUE;
    ELSE RETURN FALSE;
END;
```

Script Output x

Task completed in 0,244 seconds

Function ESTE_ANGAJAT_1AN compiled

Salariul angajatului Constantin a fost modificat
Salariul angajatului Marcus a fost modificat
Salariul angajatului Gheorghita a fost modificat
Salariul angajatului Avram a fost modificat
Salariul angajatului Topor a fost modificat
Salariul angajatului Clopot a fost modificat
Salariul angajatului Jean a fost modificat

PL/SQL procedure successfully completed.

SQL History Filter

Messages - Log

Messages Logging Page Statements

Line 202 Column 5 | Insert | Modified | Windows: CI

Exercitiul 41:

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

- Title Bar:** Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql
- File Menu:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help
- Connections Sidebar:** Oracle Connections (RotaruTeodorSGBD) expanded, showing Tables (Filtered), Views, Indexes, Packages, Procedures, Functions, Operators, Queues, Queues Tables, Triggers, Types, Sequences, Materialized Views, Materialized View Logs, Synonyms, Public Synonyms, Database Links, Public Database Links, Directories, Editions, Application Express, Java, XML Schemas, XML DB Repository, OLAP Option, Scheduler, Recycle Bin, Other Users.
- Toolbar:** Standard SQL developer toolbar with icons for New Connection, Open Connection, Save, Run, Stop, Refresh, Undo, Redo, Find, Replace, Copy, Paste, Cut, Delete, Insert, Select, Sort, Filter, SQL, Query Builder, and Aa.
- Connections Tab:** RotaruTeodorSGBD selected.
- Components Sidebar:** Components tab visible on the right.
- Worksheet Tab:** Selected tab in the main editor area.
- Query Builder Tab:** Available tab in the main editor area.
- Code Area:** Contains the following PL/SQL code:

```
-- SA SE CREEZE O FUNCTIE CARE RETURNEAZA TOATE FILMELE INCHIRIATE INTR-O LUNA PRIMITA CA PARAMETRU
CREATE OR REPLACE
FUNCTION GET_FILME_INCHIRIATE(LUNA IN NUMBER)
RETURN SYS_REFCURSOR IS
FILME_INFO SYS_REFCURSOR;
BEGIN
OPEN FILME_INFO FOR
SELECT FILME.* FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM = RAND_INCHIRIERI.ID_FILM
WHERE EXTRACT(MONTH FROM RAND_INCHIRIERI.DATA_INCHIRIERE)=LUNA;
--SELECT FILME.* INTO FILME_INFO FROM FILME WHERE FILME.ID_FILM IN
--(SELECT RAND_INCHIRIERI.ID_FILM FROM RAND_INCHIRIERI WHERE EXTRACT(MONTH FROM RAND_INCHIRIERI.DATA_INCHIRIERE)=LUNA);
RETURN FILME_INFO;
END;
SET SERVEROUTPUT ON
DECLARE
LISTA_FILME SYS_REFCURSOR;
INFO_FILME FILME%ROWTYPE;
BEGIN
LISTA_FILME := GET_FILME_INCHIRIATE(:LUNA);
LOOP
FETCH LISTA_FILME INTO INFO_FILME;
EXIT WHEN LISTA_FILME%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(INFO_FILME.DENUMIRE_FILM);
END LOOP;
CLOSE LISTA_FILME;
END;
```
- SQL History Tab:** Available tab at the bottom.
- Messages - Log Tab:** Available tab at the bottom, showing tabs for Messages, Statements, and Logging Page.
- Status Bar:** Line 240 Column 1, Insert, Modified, Windows: Cf

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Tema 5.sql

SQL Worksheet History

Worksheet Query Builder

--SA SE CREEZE O FUNCTIE CARE RETURNEAZA TOATE FILMELE INCHIRIATE INTR-O LUNA PRIMITA CA PARAMETRU

CREATE OR REPLACE

FUNCTION GET_FILME_INCHIRIATE(LUNA IN NUMBER)

RETURN SYS_REFCURSOR IS

FILME_INFO SYS_REFCURSOR;

BEGIN

OPEN FILME_INFO FOR

SELECT FILME.* FROM FILME JOIN RAND_INCHIRIERI ON FILME.ID_FILM = RAND_INCHIRIERI.ID_FILM WHERE EXTRACT(MONTH FROM RAND_INCHIRIERI.DATA_INCHIRIERE)=LUNA;

--SELECT FILME.* INTO FILME_INFO FROM FILME WHERE FILME.ID_FILM IN

--(SELECT RAND_INCHIRIERI.ID_FILM FROM RAND_INCHIRIERI WHERE EXTRACT(MONTH FROM RAND_INCHIRIERI.DATA_INCHIRIERE)=LUNA);

RETURN FILME_INFO;

END;

Script SAVED/DOCKED ON

Script Output x | Query Result x

James Bond
Mulan
Mulan
The Pianist
Schindlers List
The Shawshank Redemption
Cabin in the woods
Schindlers List
Honest thief
Cabin in the woods

PL/SQL procedure successfully completed.

SQL History Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql | Line 224 Column 23 | Insert | Windows: C:\

Exercitiul 42:

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Tema6.sql Tema 5.sql

SQL Worksheet History

Worksheet Query Builder

```
-- SA SE CREEZE O FUNCTIE CARE PRIMESTE ID-UL UNUI CLIENT SI RETURNEAZA SUMA TOTALA CHELTUITA DE ACESTA IN MAGAZIN
CREATE OR REPLACE
FUNCTION GET_SUMA_CHELTUITA_CLIENT(ID_CL IN CLIENTI_FILME.ID_CLIENT%TYPE)
RETURN NUMBER
IS
SUMA_CHELTUITA_DE_CLIENT NUMBER;
BEGIN
SELECT SUM(RAND_INCHIRIERI.PRET*RAND_INCHIRIERI.ZILE_INCHIRIATE) INTO SUMA_CHELTUITA_DE_CLIENT
FROM RAND_INCHIRIERI JOIN INCHIRIERI ON RAND_INCHIRIERI.ID_INCHIRIERE = INCHIRIERI.ID_INCHIRIERE
WHERE INCHIRIERI.ID_CLIENT = ID_CL;
IF suma_cheltuita_de_client is null THEN RETURN 0;
ELSE RETURN SUMA_CHELTUITA_DE_CLIENT;
end if;
END;

SELECT ID_CLIENT,NUME, PRENUME, GET_SUMA_CHELTUITA_CLIENT(ID_CLIENT) AS "SUMA CHELTUITA" FROM CLIENTI_FILME;
```

Script Output x | Query Result x

All Rows Fetched: 22 in 0,037 seconds

ID_CLIENT	NUME	PRENUME	SUMA CHELTUITA	
1	1	Rotaru	Teodor-Gabriel	68,26
2	2	Mihai	Topor	27,81
3	3	Andrei	Clopot	15,96
4	4	Cosmin	Cantemir	0

SQL History Compiler - Log
Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

Line 256 Column 1 | Insert | Windows: Cf

Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections Tema6.sql Tema 5.sql

SQL Worksheet History

Worksheet Query Builder

--SA SE CREEZE O FUNCTIE CARE PRIMESTE ID-UL UNUI CLIENT SI RETURNEAZA SUMA TOTALA CHELTUITA DE ACESTA IN MAGAZIN

```
CREATE OR REPLACE
FUNCTION GET_SUMA_CHELTUITA_CLIENT(ID_CL IN CLIENTI_FILME.ID_CLIENT%TYPE)
RETURN NUMBER
IS
SUMA_CHELTUITA_DE_CLIENT NUMBER;
BEGIN
```

Script Output x | Query Result x

All Rows Fetched: 22 in 0,037 seconds

ID_CLIENT	NUME	PRENUME	SUMA CHELTUITA	
1	1	Rotaru	Teodor-Gabriel	68,26
2	2	Mihai	Topor	27,81
3	3	Andrei	Clopot	15,96
4	4	Cosmin	Cantemir	0
5	5	Rares	Dumitrescu	11,97
6	6	Stefan	Golescu	0
7	7	Adrian	Adamescu	0
8	8	Albert	Balanici	3,99
9	9	Valentin	Afrim	6,57
10	10	Calin	Barbulescu	3,98
11	11	Mircea	Banica	4,38
12	12	Vlad	Anghel	5,97
13	13	Radu	Cernea	5,98

SQL History Compiler - Log

Messages Logging Page Statements Compiler

Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql

Line 256 Column 1 | Insert | Windows: CI

Pachetul ,SUBPROGRAME_FILME' contine functiile si procedurile prezentate anterior.

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

- Title Bar:** Oracle SQL Developer : C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema6.sql
- Menu Bar:** File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help
- Toolbar:** Includes icons for New Connection, Open Connection, Save, Run, Stop, and Help.
- Connections Sidebar:** Lists various database objects including VARSTA, TELEFON, EMAIL_CLI, SEX, PREMIUM, COMENZI, CRITICI, DEPARTAMENT, DISTRIBUTIE, ERORI, FILME, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNC, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRI, ID_INCHIR, ID_FILM, ZILE_INCHI, DATA_INCHI, DATA_RETII, PRET, RECENZII, REGIUNI, TARI, Views, Indexes, Packages, Procedures.
- SQL Worksheet:** The current tab is "Tema6.sql". It contains the PL/SQL code for the package:

```
CREATE OR REPLACE PACKAGE SUBPROGRAME_FILME IS
    PROCEDURE MODIFICA_PRET_INCHIRIERE_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, P_PRET_INC_NOU IN FILME.PRET_INCHIRIERE%TYPE);
    PROCEDURE AFISARE_INFO_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, NR_INC OUT NUMBER, P_DENUMIRE_FILM OUT FILME.DENUMIRE_FILM%TYPE);
    PROCEDURE GET_RECENZIE_MEDIE_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, P_RECENZIE_MEDIE OUT RECENZII.NOTA%TYPE);
    PROCEDURE MODIFICA_PRET_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, P_RECENZIE_MEDIE IN RECENZII.NOTA%TYPE, PROCENT IN OUT NUMBER);
    FUNCTION ESTE_ANGAJAT_1AN(P_ID_ANGAJAT IN ANGAJATI_FILME.ID_ANGAJAT%TYPE) RETURN BOOLEAN;
    FUNCTION GET_FILME_INCHIRIATE(LUNA IN NUMBER) RETURN SYS_REFCURSOR;
    FUNCTION GET_SUMA_CHELTUITA_CLIENT(ID_CL IN CLIENTI_FILME.ID_CLIENT%TYPE) RETURN NUMBER;
END;
/
CREATE OR REPLACE PACKAGE BODY SUBPROGRAME_FILME IS
    PROCEDURE MODIFICA_PRET_INCHIRIERE_FILM(P_ID_FILM IN FILME.ID_FILM%TYPE, P_PRET_INC_NOU IN FILME.PRET_INCHIRIERE%TYPE) IS
        BEGIN
            UPDATE FILME SET PRET_INCHIRIERE = P_PRET_INC_NOU WHERE ID_FILM = P_ID_FILM;
```

- Script Output:** Shows the compilation results:
 - Package SUBPROGRAME_FILME compiled
 - Package Body SUBPROGRAME_FILME compiledTask completed in 1,546 seconds.
- Compiler - Log:** Shows tabs for Messages, Logging Page, Statements, and Compiler.
- Status Bar:** Saved: C:\Users\FOLIO\AppData\Roaming\SQL Developer\Tema 5.sql | Line 10 Column 75 | Insert | Modified | Windows: Cf

Trigger 1:

The screenshot shows the Oracle SQL Developer interface with a trigger creation script in the central workspace.

Connections pane (left): Shows the connection to "RotaruTeodorSGBD" and its tables: ACTORI, ANGAJATI, ANGAJATI_FILME, ANGAJATI_PREMIATI_FILMME, CLIENTI, CLIENTI_FILME, COMENZI, CRITICI, DEPARTAMENTE, DISTRIBUTIE, ERORI, FILME (selected), ID_FILM, DENUMIRE_FILM, CATEGORIE_FILM, LIMBA_FILM, DURATA_FILM_MINUTE, DATA_LANSARE, PRET_FURNIZOR, PRET_INCHIRIERE, STOC, FUNCTII, FUNCTII_FILME, INCHIRIERI, ISTORIC_FUNCTII, ISTORIC_FUNCTII_FILME, LOCATII, PRODUSE, RAND_COMENZI, RAND_INCHIRIERI, RECENZII.

SQL Worksheet pane (top right): The script content is as follows:

```
-- SE DECIDE ACHIZITIONAREA A 10 BUCATI SUPLIMENTARE DIN FIECARE FILM CU PRET DE CUMPARARE DE LA FURNIZOR MAI IEFTIN DE 40 DE LEI
-- SA SE ACTUALIZEZE STOCURILE FILMELOR CU PRETUL DE LA FURNIZOR MAI MIC DE 40 RON
-- SA SE CREEZE UN TRIGGER CARE AFISEAZA USERUL SI DATA LA CARE S-A REALIZAT ACTUALIZAREA, PRECUM SI VECHIUL SI NOUL STOC
SET SERVEROUTPUT ON;
CREATE OR REPLACE TRIGGER UPDATE_FILME_MESSAGE
BEFORE UPDATE ON FILME FOR EACH ROW ENABLE
DECLARE
USER_CURENT VARCHAR2(30);
DATA_CURENTA VARCHAR2(20);
BEGIN
SELECT USER, CURRENT_DATE INTO USER_CURENT, DATA_CURENTA FROM DUAL;
DBMS_OUTPUT.PUT_LINE('Film: ' || :new.denumire_film || '-- Modificare finalizata! User ' || USER_CURENT || ' Data ' || DATA_CURENTA);
DBMS_OUTPUT.PUT_LINE('Vecchiul stoc: ' || :old.stoc || ' Noul stoc: ' || :new.stoc);
END;
BEGIN
UPDATE FILME SET STOC = STOC + 10 WHERE PRET_FURNIZOR < 40;
END;
```

Script Output pane (bottom left):

```
Film: The Mechanic-- Modificare finalizata! User ROTARUT_57 Data 22-05-2021
Vecchiul stoc: 30 Noul stoc: 40
Film: Profu-- Modificare finalizata! User ROTARUT_57 Data 22-05-2021
Vecchiul stoc: 23 Noul stoc: 33

PL/SQL procedure successfully completed.
```

Compiler - Log pane (bottom right):

```
Messages | Logging Page | Statements | Compiler
```

Trigger 2:

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

- Connections:** Oracle Connections - RotaruTeodorSGBD
- SQL Worksheet:** triggers.sql (FILME tab) - Contains the PL/SQL code for the trigger.
- Script Output:** Shows the execution results of the trigger, displaying error messages for various movie titles.
- Compiler - Log:** Shows the status of the trigger compilation.

```
--SA SE MAREASCA CU 60% PRETURILE DE INCHIRIERE A FILMELOR
--SA SE CREEZE UN TRIGGER CARE ASIGURA CA PRETUL NU DEVINE MAI MARE DE 10 RON
ALTER TRIGGER UPDATE_FILME_MESSAGE DISABLE;
CREATE OR REPLACE TRIGGER UPDATE_FILME_PRICE_RESTRICTION
BEFORE UPDATE ON FILME FOR EACH ROW ENABLE
BEGIN
  IF(:NEW.PRET_INCHIRIERE > 10) THEN
    DBMS_OUTPUT.PUT_LINE('Film - ' || :new.denumire_film || ' - Eroare! Pretul nu poate fi mai mare de 10 RON!');
    :new.pret_inchiriere := 10;
  END IF;
END;
BEGIN
  UPDATE FILME SET PRET_INCHIRIERE = 1.6 * PRET_INCHIRIERE;
END;
```

Film - James Bond - Eroare! Pretul nu poate fi mai mare de 10 RON!
Film - Cabin in the woods - Eroare! Pretul nu poate fi mai mare de 10 RON!
Film - A walk to remember - Eroare! Pretul nu poate fi mai mare de 10 RON!
Film - Poarta Alba - Eroare! Pretul nu poate fi mai mare de 10 RON!
Film - Gladiator - Eroare! Pretul nu poate fi mai mare de 10 RON!
Film - Schidlers List - Eroare! Pretul nu poate fi mai mare de 10 RON!

PL/SQL procedure successfully completed.

Interactive report pentru tabela ,FILME':

App Rotaru Teodor Gabriel SGBD ? rotaruteodor19@stud.ase.ro

Interactive report - filme

Denumire Film	Categorie Film	Limba Film	Durata Film Minute	Data Lansare	Pret Furnizor	Pret Inchiriere	Stoc				
A walk to remember	Dragoste / Drama	Engleza	100	6/11/2008	52.99	2.7	6				
Animal Crackers	Animatie / Comedie	Engleza	104	11/21/2017	49.99	2.4	11				
Ava	Actiune / Crima	Engleza	95	9/11/2020	79.99	3.1	10				
Black Swan	Drama / Thriller	Engleza	107	9/11/2010	31.99	1.8	9				
Bloodshot	Actiune / SF / Drama	Engleza	109	7/31/2020	79.99	4.2	13				
Cabin in the woods	Horror	N/A	97	10/21/2010	49.99	2.1	11				
Cardinalul	Drama / Istoric	Romana	109	9/18/2019	29.99	1.1	30				
Forrest Gump	Comedie / Drama	Engleza	114	9/10/1994	39.99	2.3	11				
Gladiator	Actiune / Drama	Engleza	114	11/4/2000	39.99	1.2	13				
Greenland	Actiune / Thriller	Engleza	92	3/12/2020	75.99	3.1	7				
Honest thief	Actiune / Crima / Drama	Engleza	112	10/9/2020	79.99	4.2	11				
Inglourious Basterds	Actiune / Aventura	Engleza	94	11/11/2009	39.99	2.1	11				
Insidious	Horror	Engleza	104	4/1/2011	59.99	2.8	13				
James Bond	Actiune	Engleza	112	11/9/2009	69.99	2.8	10				
Made in Italy	Actiune / Aventura	Engleza	109	6/4/2020	69.99	3.1	9				
Mulan	Aventura	N/A	92	11/2/2020	67.99	3.6	2				
Poarta Alba	Drama	Romana	86	2/7/2014	25.99	1.0	2				
Profu	Documentar	Romana	82	12/11/2019	39.99	1.8	3				
Saw	Horror	Engleza	96	12/9/2004	29.99	1.6	16				
Schindlers List	Drama / Istoric	Engleza	104	11/12/1993	39.99	2.4	15				
Tenet	SF / Actiune	N/A	96	12/11/2020	79.99	3.1	10				
The Butterfly Effect	SF / Thriller	Engleză	105	11/1/2004	29.99	1.6	19				
The Departed	Drama	Home	Application 94533	Edit Page 2	Session	View Debug	No Debug	Page Info	Quick Edit	Customize	Reset

Master detail pentru tabelele ,INCHIRIERI' si ,RAND_INCHIRIERI':

Filme închiriate					
	Id Film	Zile Închiriate	Data Închiriere	Data Returnare	Pret
<input checked="" type="checkbox"/>	21 - Forrest Gump	2	7/10/2020	7/12/2020	3.99
<input type="checkbox"/>	10 - The war with grandpa	3	7/10/2020	7/13/2020	3.99

Pie Chart pentru a vedea media pretului de inchiriere a filmelor in functie de categorie:

≡ App Rotaru Teodor Gabriel SGBD

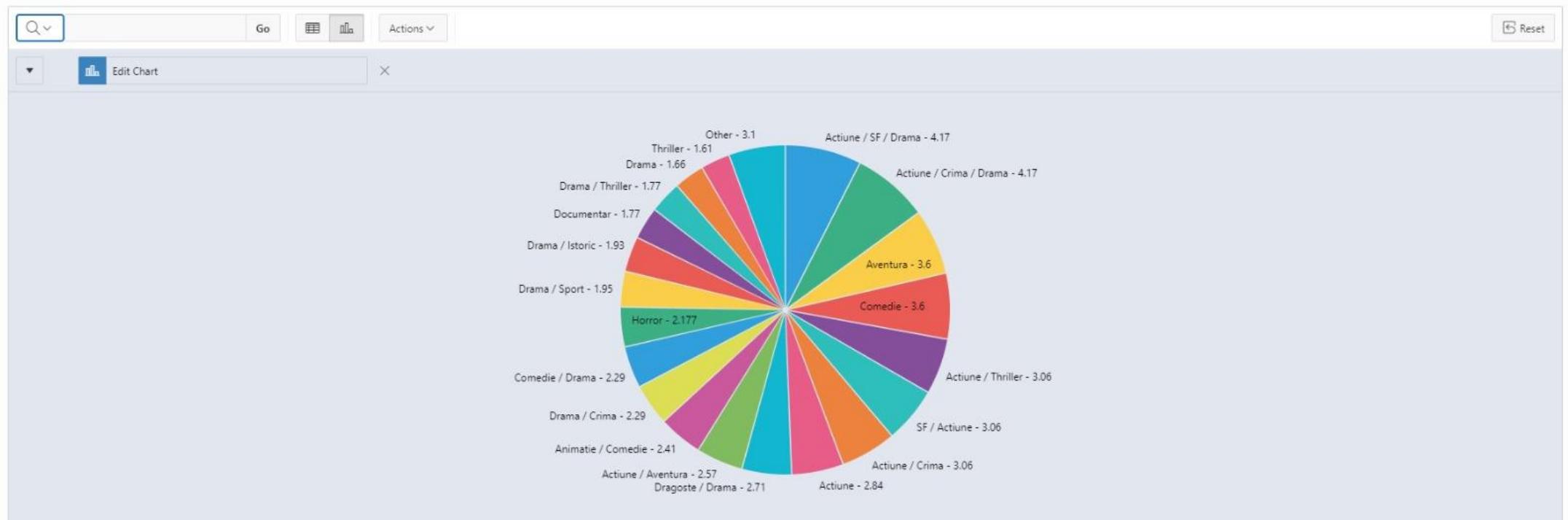
?

?

?

rotaruteodor19@stud.ase.ro

Interactive report - filme



Release 1.0

Home Application 94533 Edit Page 2 Session View Debug No Debug Page Info Quick Edit Customize

Classic report pentru tabelele ,RECENZII' si ,ACTORI':

App Rotaru Teodor Gabriel SGBD

Home Classic Report - Recenzii Interactive report - filme Master Detail Inchirieri Administration

Report recenzii

Id Film ↑↓	Id Critic	Nota
1	1	9.7
1	2	8.9
2	5	9.3
3	5	7.1
4	6	9.3
4	1	7.2
5	6	7.5
6	1	9.4
7	2	9.1
8	3	7.3
9	2	7.9
10	3	7.1
11	3	9.1
11	2	8.2
11	4	9.1

Download

1 - 15 Next *

Report actori

Id Actor ↑↓	Sex	Nume	Prenume
1	M	Robert	De Niro
2	M	Albert	Smith
3	M	Bruce	Willis
4	F	Maria	Carey
5	M	Jason	Statham
6	F	Judi	Dench
7	F	Nicole	Kidman
8	F	Sophie	Marceau
9	M	Shaun	Young
10	F	Anne	Dorval
11	M	Christian	Bale
12	F	Cate	Blanchett

Home Application 94533 Edit Page 4 Session View Debug No Debug Page Info Quick Edit Customize