ACADEMIA DE STUDII ECONOMICE,BUCUREȘTI

FACULTATEA DE CIBERNETICĂ,STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ

Proiect

Baze de date

Nume: Manolache Andrei-Stefan

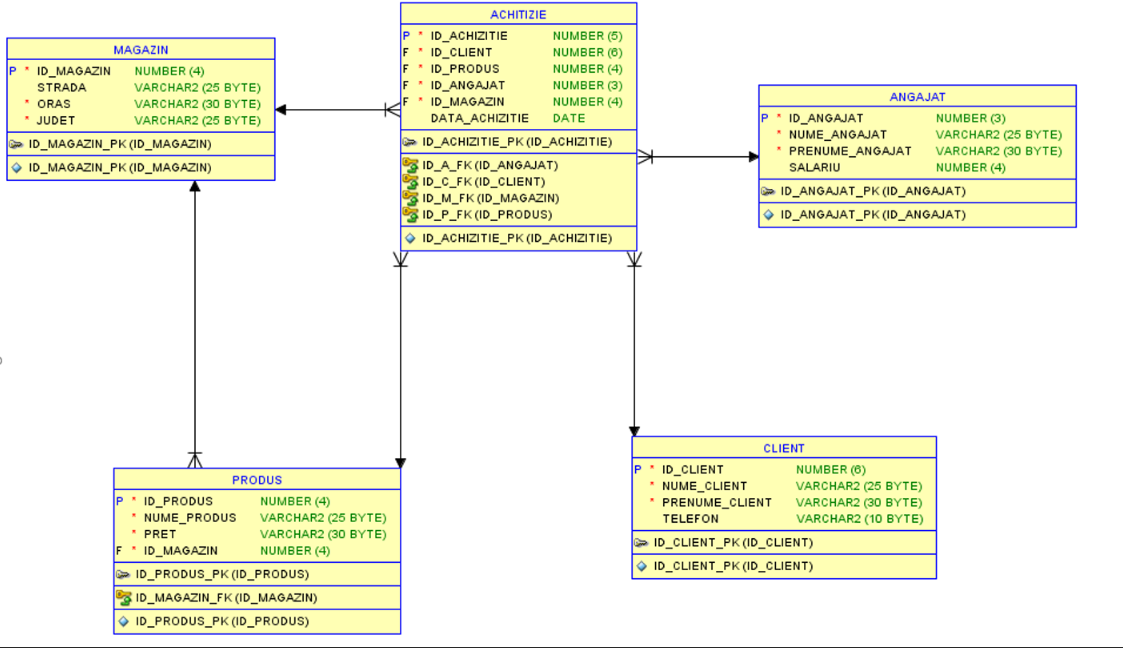
Grupa:1045, Seria:A

**Descrierea firmei folosita in proiect:**



Firma folosita este Media Galaxy, o firmă românească detinuta de Altex, care comercializează produse electronice, electrocasnice, echipamente si componente IT. Am construit o baza de date care conține informații despre angajați, sucursalele deschise, clienți si achizițiile acestora.

**Schema Bazei de date:**



**Comenzi LDD (Data Definition Language – Limbaj de Definire a Datelor)**

**1.** **Să se creeze tabelele ANGAJAT, MAGAZIN, PRODUS, CLIENT și ACHIZITIE în care sa fie precizate restricțiile de integritate:**

**Tabele:**

Angajat:

CREATE TABLE ANGAJAT

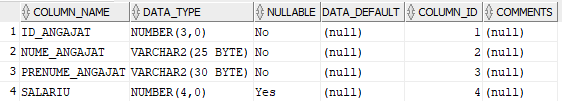
(id\_angajat NUMBER(3) CONSTRAINT id\_angajat\_pk PRIMARY KEY,

nume\_angajat VARCHAR2(25) NOT NULL,

prenume\_angajat VARCHAR2(30) NOT NULL,

salariu number(4)

);



Magazin

CREATE TABLE MAGAZIN

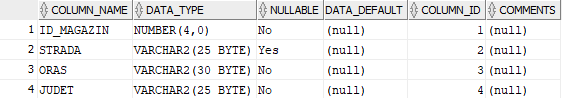
(id\_magazin NUMBER(4) CONSTRAINT id\_magazin\_pk PRIMARY KEY,

strada VARCHAR2(25),

oras VARCHAR2(30) NOT NULL,

judet VARCHAR2(25) NOT NULL

);



Produs

CREATE TABLE PRODUS

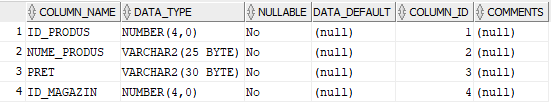
(id\_produs NUMBER(4) CONSTRAINT id\_produs\_pk PRIMARY KEY,

nume\_produs VARCHAR2(25) NOT NULL,

pret VARCHAR2(30) NOT NULL,

id\_magazin NUMBER(4) NOT NULL,

CONSTRAINT id\_magazin\_fk FOREIGN KEY (ID\_MAGAZIN) REFERENCES MAGAZIN(ID\_MAGAZIN) );



Client

CREATE TABLE Client

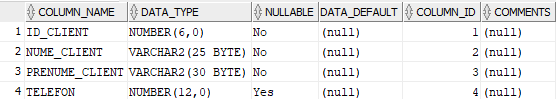
(id\_client NUMBER(6) CONSTRAINT id\_client\_pk PRIMARY KEY,

nume\_client VARCHAR2(25) NOT NULL,

prenume\_client VARCHAR2(30) NOT NULL,

Telefon number(12)

);



Achizitie:

CREATE TABLE ACHIZITIE

(id\_achizitie NUMBER(5) CONSTRAINT id\_achizitie\_pk PRIMARY KEY,

id\_client number(6) NOT NULL,

id\_produs number(4) not null,

id\_angajat number(3) not null,

id\_magazin number(4) not null,

Data\_Achizitie Date,

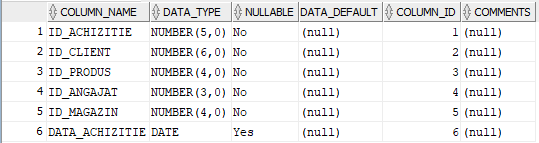
CONSTRAINT id\_c\_fk FOREIGN KEY (ID\_CLIENT) REFERENCES Client(ID\_CLIENT),

CONSTRAINT id\_p\_fk FOREIGN KEY (ID\_PRODUS) REFERENCES PRODUS(ID\_PRODUS),

CONSTRAINT id\_a\_fk FOREIGN KEY (ID\_ANGAJAT) REFERENCES Angajat(ID\_ANGAJAT),

CONSTRAINT id\_m\_fk FOREIGN KEY (ID\_MAGAZIN) REFERENCES Magazin(ID\_MAGAZIN)

);



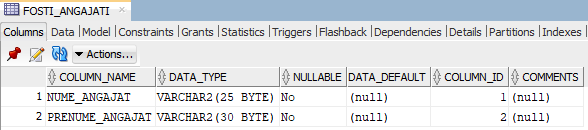
Cu ajutorul comenzilor de mai sus am efectuat tabelele, cu restricții de integritate necesare.

**2.** **Să se creeze tabelul FOSTI\_ANGAJATI pe baza tabelei ANGAJAT ce va conține doar o parte din coloanele tabelei initiale (nume\_angajat, prenume\_angajat).**

CREATE TABLE Fosti\_Angajati

AS

SELECT nume\_angajat, prenume\_angajat FROM ANGAJAT;



Prin intermediul tabelei Angajat a fost creat un nou tabel ce conține doar doua din atributele tabelului inițial.

**3.** **Să se redenumească tabelul FOSTI\_ANGAJATI cu FOSTI\_LUCRATORI, să se adauge coloanele Email,telefon și funcția și sa se inactiveze coloana funcția.**

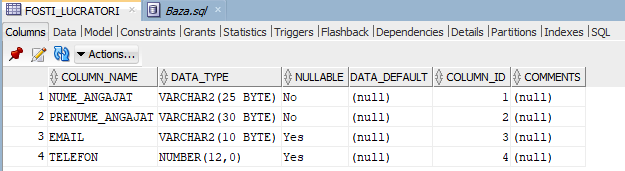
ALTER TABLE FOSTI\_ANGAJATI RENAME TO FOSTI\_LUCRATORI;

ALTER TABLE FOSTI\_LUCRATORI

ADD (email VARCHAR2(10), telefon NUMBER(12), functia VARCHAR2(20));

ALTER TABLE FOSTI\_LUCRATORI

SET UNUSED COLUMN functia;

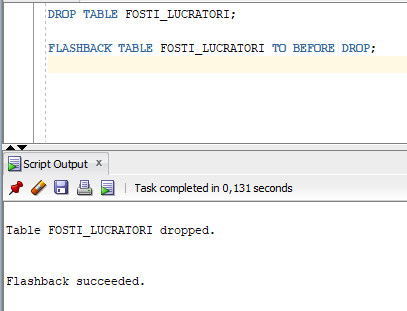


Noul tabel creat la exercițiul doi a fost redenumit, fiindu-i adăugate incă doua coloane:Email și telefon, funcția fiind inactivă.

**4.** **Să se șteargă tabela FOSTI\_LUCRATORI și apoi să se recupereze:**

DROP TABLE FOSTI\_LUCRATORI;

FLASHBACK TABLE FOSTI\_LUCRATORI TO BEFORE DROP;



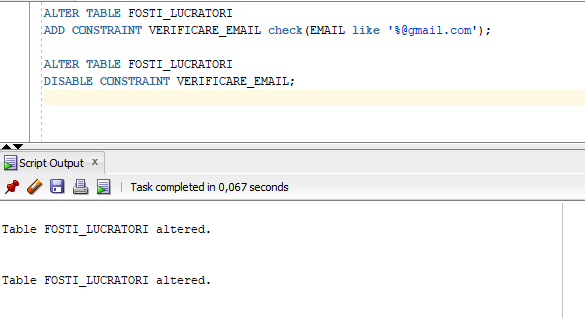
**5. Adaugati o restrictie astfel incât câmpul EMAIL din tabela FOSTI\_LUCRATORI sa fie de forma %@gmail.com și apoi să se dezactiveze restricția.**

ALTER TABLE FOSTI\_LUCRATORI

ADD CONSTRAINT VERIFICARE\_EMAIL check(EMAIL like '%@gmail.com');

ALTER TABLE FOSTI\_LUCRATORI

DISABLE CONSTRAINT VERIFICARE\_EMAIL;



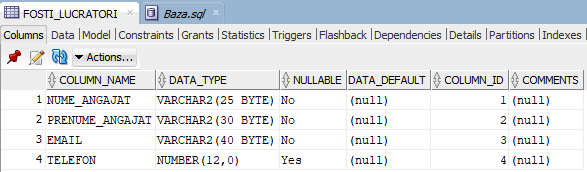
**6. Sa se modifice tabela FOSTI\_LUCRĂTORI astfel incăt câmpul EMAIL sa nu poată fi NULL iar lungimea să fie de 40.**

ALTER TABLE FOSTI\_LUCRATORI

MODIFY EMAIL NOT NULL;

ALTER TABLE FOSTI\_LUCRATORI

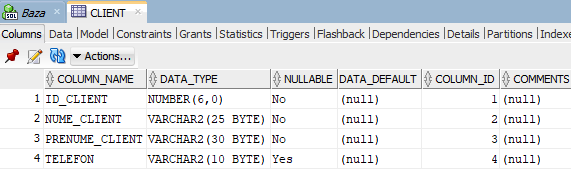
MODIFY(EMAIL VARCHAR2(40));



**7.Să se modifice tabela Client astfel încat câmpul TELEFON sa fie de tip VARCHAR2 de lungime 10.**

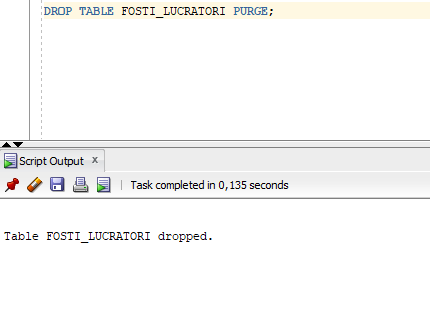
ALTER TABLE CLIENT

MODIFY(Telefon VARCHAR2(10));



**8. Să se șteargă fara posibilitate de recuperare tabela FOSTI\_LUCRATORI:**

DROP TABLE FOSTI\_LUCRATORI PURGE;

****

**COMENZI LMD (Data Manipulation Language – Limbaj de Manipulare a Datelor)**

**Adăugarea înregistrărilor:**

**Angajat:**

INSERT INTO ANGAJAT VALUES( 001, 'Ionescu', 'Sebastian', 1500);

INSERT INTO ANGAJAT VALUES( 002, 'Popescu', 'Andrei', 1000);

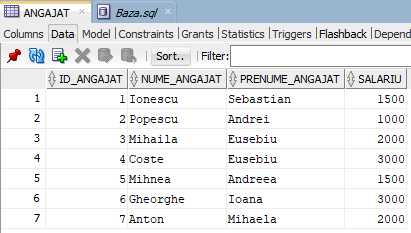
INSERT INTO ANGAJAT VALUES( 003, 'Mihaila', 'Eusebiu', 2000);

INSERT INTO ANGAJAT VALUES( 004, 'Coste', 'Eusebiu', 3000);

INSERT INTO ANGAJAT VALUES( 005, 'Mihnea', 'Andreea', 1500);

INSERT INTO ANGAJAT VALUES( 006, 'Gheorghe', 'Ioana', 3000);

INSERT INTO ANGAJAT VALUES( 007, 'Anton', 'Mihaela', 2000);



**Magazin:**

INSERT INTO MAGAZIN VALUES(0001,'Bucuriei 14','Arges','Pitesti');

INSERT INTO MAGAZIN VALUES(0002,'Trandafir 3','Constanta','Constanta');

INSERT INTO MAGAZIN VALUES(0003,'Ion Creanga 7','Brasov','Brasov');

INSERT INTO MAGAZIN VALUES(0004,'Primaverii 1','Harghita','Miercurea Ciuc');

INSERT INTO MAGAZIN VALUES(0005,'Jupiter 16','Buzau','Buzau');

INSERT INTO MAGAZIN VALUES(0006,'Panselutelor 11','Cluj','Cluj-Napoca');

INSERT INTO MAGAZIN VALUES(0007,'Mihai Eminescu 12','Gord','Targu Jiu');



**Produs:**

INSERT INTO PRODUS VALUES(0001,'Samsung Galaxy S21', '4500',0002);

INSERT INTO PRODUS VALUES(0002,'Laptop ASUS ROG G15', '6500',0001);

INSERT INTO PRODUS VALUES(0003,'Fier de calcat Whirlpool', '500',0002);

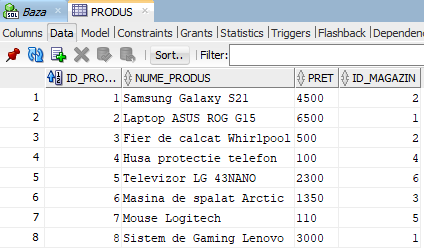
INSERT INTO PRODUS VALUES(0004,'Husa protectie telefon', '100',0004);

INSERT INTO PRODUS VALUES(0005,'Televizor LG 43NANO', '2300',0006);

INSERT INTO PRODUS VALUES(0007,'Mouse Logitech', '110',0005);

INSERT INTO PRODUS VALUES(0008,'Sistem de Gaming Lenovo', '3000',0001);

INSERT INTO PRODUS VALUES(0006,'Masina de spalat Arctic', '1350',0003);



**Client:**

INSERT INTO CLIENT VALUES(106015,'Neagoe','Mihail','0789123413');

INSERT INTO CLIENT VALUES(106016,'Darius','Andrei','0768423461');

INSERT INTO CLIENT VALUES(106018,'Rotaru','Sebastian','0745161954');

INSERT INTO CLIENT VALUES(106019,'Anton','Andreea','0730202677');

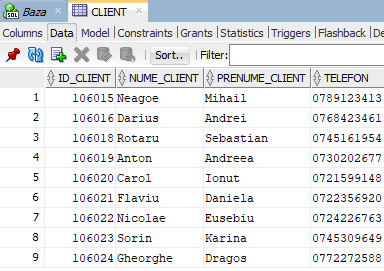
INSERT INTO CLIENT VALUES(106020,'Carol','Ionut','0721599148');

INSERT INTO CLIENT VALUES(106021,'Flaviu','Daniela','0722356920');

INSERT INTO CLIENT VALUES(106022,'Nicolae','Eusebiu','0724226763');

INSERT INTO CLIENT VALUES(106023,'Sorin','Karina','0745309649');

INSERT INTO CLIENT VALUES(106024,'Gheorghe','Dragos','0772272588');



**Achizitie:**

INSERT INTO ACHIZITIE VALUES(21051,106015,0001,003,0002,TO\_DATE('01-FEB-2021', 'DD MON YYYY'));

INSERT INTO ACHIZITIE VALUES(21052,106016,0004,002,0002,TO\_DATE('16-MART-2021', 'DD-MON-YYYY'));

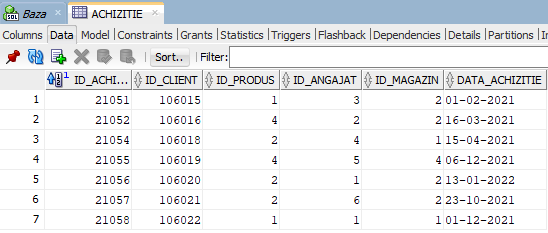
INSERT INTO ACHIZITIE VALUES(21054,106018,0002,004,0001,TO\_DATE('15-APR-2021', 'DD-MON-YYYY'));

INSERT INTO ACHIZITIE VALUES(21055,106019,0004,005,0004,TO\_DATE('06-DEC-2021', 'DD-MON-YYYY'));

INSERT INTO ACHIZITIE VALUES(21056,106020,0002,001,0002,TO\_DATE('13-IAN-2022', 'DD-MON-YYYY'));

INSERT INTO ACHIZITIE VALUES(21057,106021,0002,006,0002,TO\_DATE('23-OCT-2021', 'DD-MON-YYYY'));

INSERT INTO ACHIZITIE VALUES(21058,106022,0001,001,0001,TO\_DATE('01-DEC-2021', 'DD-MON-YYYY'));



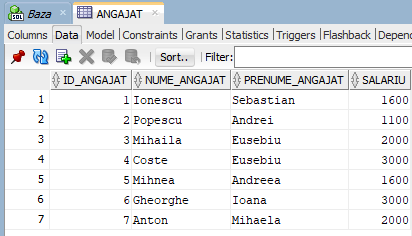
**1.** **Să se modifice salariile din tabela ANGAJAT persoanelor care au salariul mai mic decat 1600:**

UPDATE ANGAJAT

SET salariu=salariu+100

WHERE salariu<1600;

SELECT \* FROM ANGAJAT;

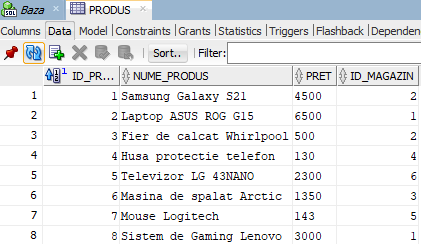


**2.** **Să se mărească prețul produselor cu o valoare mai mica de 500:**

UPDATE PRODUS SET pret=pret+pret\*0.3

WHERE pret<500;

SELECT \* FROM PRODUS;



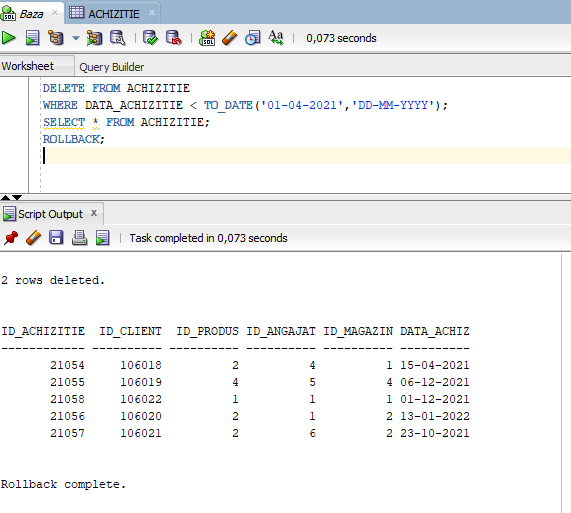
**3.** **Să se șteargă toate achizițiile efectuate înainte de aprilie 2021. Să se anuleze procedura.**

DELETE FROM ACHIZITIE

WHERE DATA\_ACHIZITIE < TO\_DATE('01-04-2021','DD-MM-YYYY');

SELECT \* FROM ACHIZITIE;

ROLLBACK;

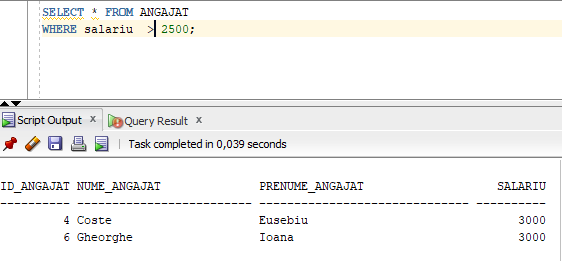


**INTEROGAREA TABELELOR**

**1.Să se selecteze angajații care au salariul mai mare de 2500:**

SELECT \* FROM ANGAJAT

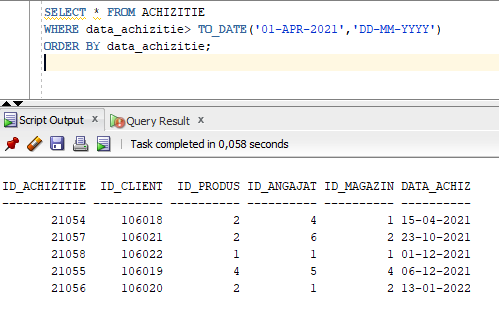
WHERE salariu > 2500;



**2.Să se selecteze toate achizițiile realizate dupa 1 aprilie 2021:**

SELECT \* FROM ACHIZITIE

WHERE data\_achizitie> TO\_DATE('01-APR-2021','DD-MM-YYYY')

ORDER BY data\_achizitie;

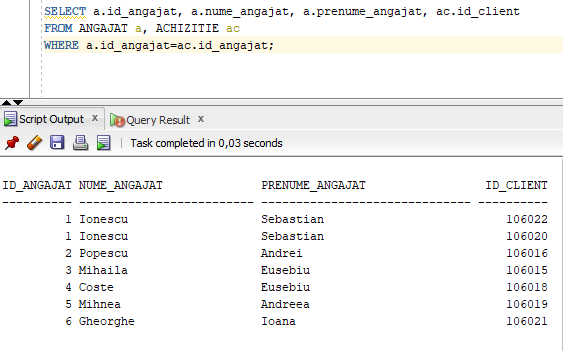
**JONCȚIUNI ÎNTRE RELAȚII**

**Sa se selecteze id\_angajat, nume\_angajat si prenume\_angajat din tabela ANGAJAT si id\_client din tabela ACHIZITIE si sa se realizeze jonctiunea dintre cele doua tabele:**

SELECT a.id\_angajat, a.nume\_angajat, a.prenume\_angajat, ac.id\_client

FROM ANGAJAT a, ACHIZITIE ac

WHERE a.id\_angajat=ac.id\_angajat;



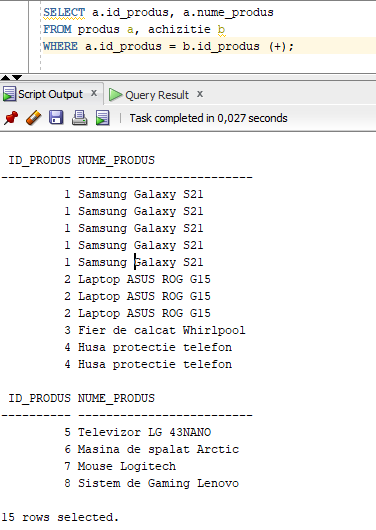
**JONCȚIUNE EXTERNĂ**

**Să se afișeze produsele chiar dacă nu au fost comandate**

SELECT a.id\_produs, a.nume\_produs

FROM produs a, achizitie b

WHERE a.id\_produs = b.id\_produs (+);

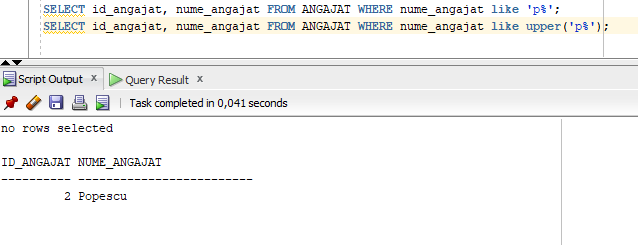


**FUNCȚII SINGLE-ROW**

**1.Să se afișeze salariații al căror nume începe cu litera ‘P’:**

SELECT id\_angajat, nume\_angajat FROM ANGAJAT WHERE nume\_angajat like 'p%';

SELECT id\_angajat, nume\_angajat FROM ANGAJAT WHERE nume\_angajat like upper('p%');

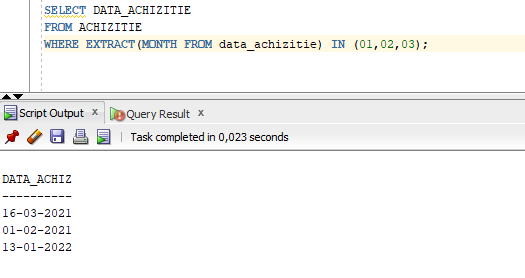


**2.Să se afișeze achizițiile efectuate în lunile ianuarie,februarie,martie.**

SELECT DATA\_ACHIZITIE

FROM ACHIZITIE

WHERE EXTRACT(MONTH FROM data\_achizitie) IN (01,02,03);

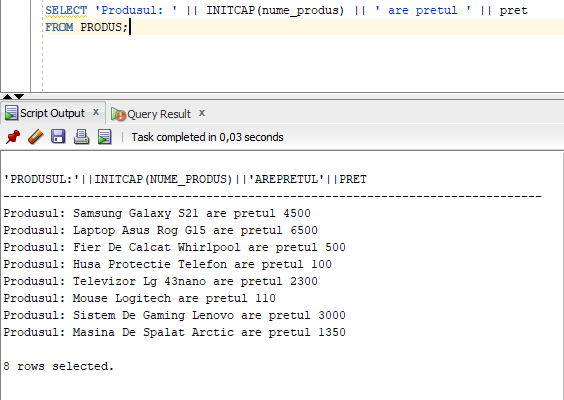


**Operatorul de concatenare (||)**

**3.Să se afișeze denumirea produsului si prețul acestuia:**

SELECT 'Produsul: ' || INITCAP(nume\_produs) || ' are pretul ' || pret

FROM PRODUS;

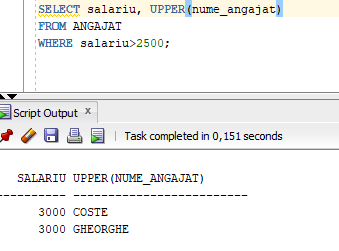


**4.Să se afișeze cu litere mari angajatii care au un salariu mai mare de 2500:**

SELECT salariu, UPPER(nume\_angajat)

FROM ANGAJAT

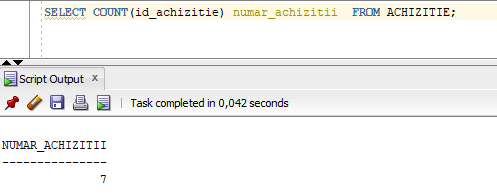
WHERE salariu>2500;



**FUNCȚII DE GRUP**

**1.Să se afișeze numărul total de comenzi încheiate:**

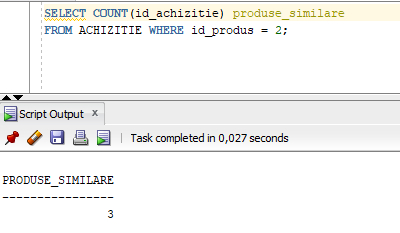
SELECT COUNT(id\_achizitie) numar\_achizitii FROM ACHIZITIE;



**2.** **Sa se afiseze pe cate comenzi apare produsul cu codul 2:**

SELECT COUNT(id\_achizitie) produse\_similare

FROM ACHIZITIE WHERE id\_produs = 2;



**PARCURGEREA PE UN ANUMIT NIVEL IERARHIC**

Deoarece în baza de date realizată toți au același nivel, voi realiza un camp nou id\_manager, in funcție de cel mai mare salariu:

ALTER TABLE Angajat

ADD id\_manager number(1);

UPDATE angajat

SET salariu=3500

where id\_angajat=4;

UPDATE angajat

SET id\_manager=7

where id\_angajat=1;

UPDATE angajat

SET id\_manager=1

where id\_angajat=2;

UPDATE angajat

SET id\_manager=6

where id\_angajat=3;

UPDATE angajat

SET id\_manager=0

where id\_angajat=4;

UPDATE angajat

SET id\_manager=7

where id\_angajat=5;

UPDATE angajat

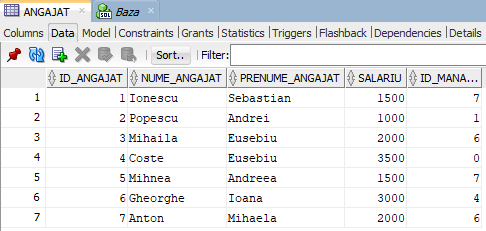
SET id\_manager=4

where id\_angajat=6;

UPDATE angajat

SET id\_manager=6

where id\_angajat=7;

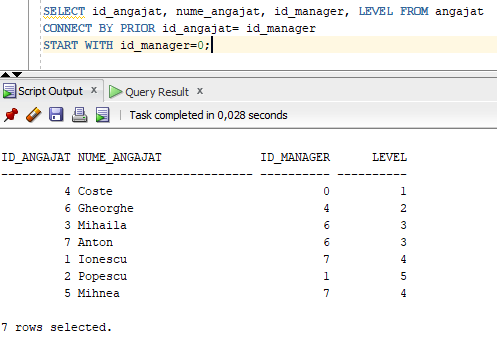


**Sa se afiseze angajatii si nivelul ierarhic al acestora pornind de la angajatul cu id-ul 1 (sa se ordoneze in functie de nivelul ierahic).**

SELECT id\_angajat, nume\_angajat, id\_manager, LEVEL FROM angajat

CONNECT BY PRIOR id\_angajat= id\_manager

START WITH id\_manager=0;



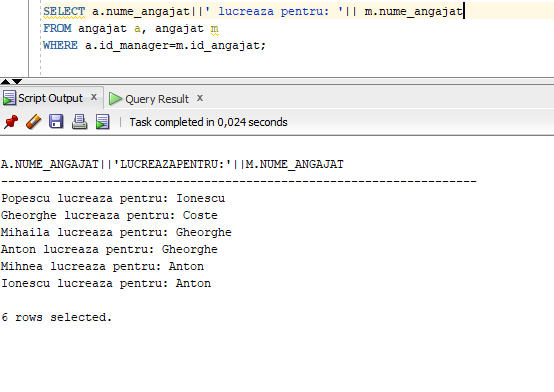
**JONCTIUNEA DINTRE TABELA IN ACEEASI TABELA**

**Să se afişeze numele fiecarui angajat şi numele sefului direct superior:**

**SELECT a.nume\_angajat||' lucreaza pentru: '|| m.nume\_angajat**

**FROM angajat a, angajat m**

**WHERE a.id\_manager=m.id\_angajat;**



Acum, am să șterg id\_manager pentru a reveni la baza de date inițială:

ALTER TABLE Angajat

DROP COLUMN id\_manager;

UPDATE angajat

SET salariu=300

where id\_angajat=4;

**FUNCȚIA DECODE ȘI EXPRESIA CASE**

**Să se realizeze o reducere în funcție de prețul produselor astfel:**

-5% din pret daca pretul este intre 50 si 500

-10% din pret daca pretul este intre 500 si 999

-20% din pret daca pretul este mai mare de 1000.

Pentru celelalte functii nu va fi aplicata o reducere.

SELECT id\_produs, nume\_produs, pret,

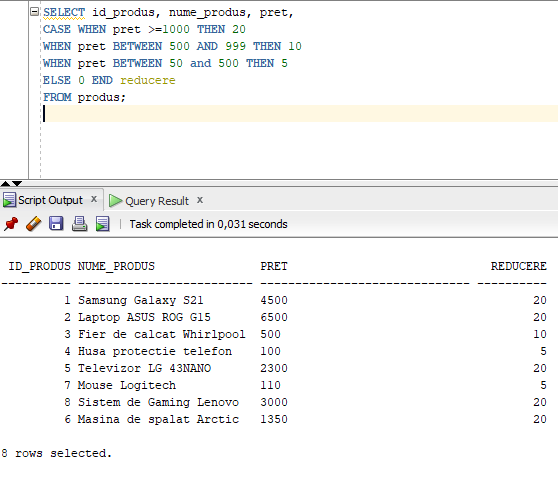
CASE WHEN pret >=1000 THEN 20

WHEN pret BETWEEN 500 AND 999 THEN 10

WHEN pret BETWEEN 50 and 500 THEN 5

ELSE 0 END reducere

FROM produs;



**TABELE VIRTUALE**

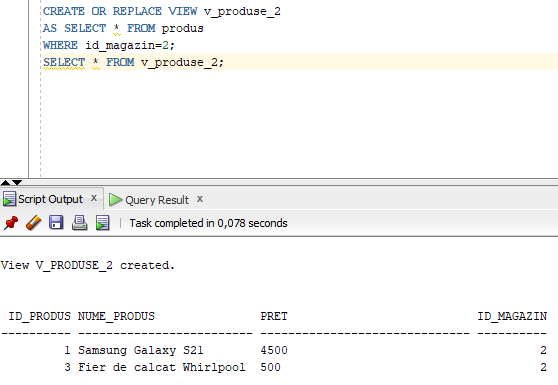
**Sa se realizeze o tabela virtuala cu toate produsele din magazinul 2.**

CREATE OR REPLACE VIEW v\_produse\_2

AS SELECT \* FROM produs

WHERE id\_magazin=2;

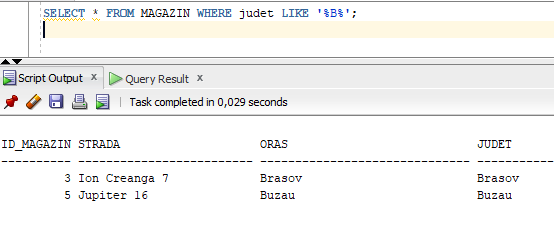
SELECT \* FROM v\_produse\_2;



**INDECSI**

**Să se creeze un index pe tabela Magazin pe coloana judet.**

SELECT \* FROM MAGAZIN WHERE judet LIKE '%B%';



**SECVENȚE**

**Sa se creeze o secvență pentru asigurarea unicitatii cheii primare din tabela Achizitie.**

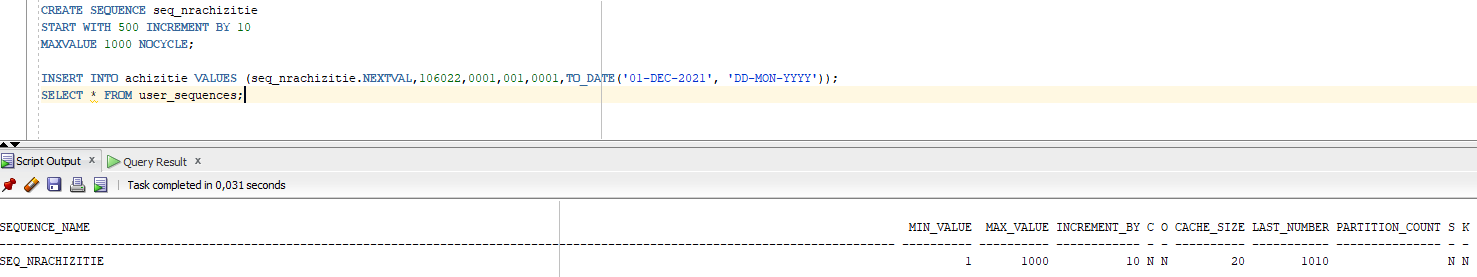
CREATE SEQUENCE seq\_nrachizitie

START WITH 500 INCREMENT BY 10

MAXVALUE 1000 NOCYCLE;

INSERT INTO achizitie VALUES (seq\_nrachizitie.NEXTVAL,106022,0001,001,0001,TO\_DATE('01-DEC-2021', 'DD-MON-YYYY'));

SELECT \* FROM user\_sequences;



**OPERATORII ALGEBREI RELAȚIONALE**

Sa se afișeze produsele care au prețul intre 1000 și 3000 fară acele produse care au prețul între 2000 si 2500.

SELECT \*FROM produs WHERE pret BETWEEN 1000 AND 3000

MINUS

SELECT \* FROM produs WHERE pret BETWEEN 2000 AND 2500;

