Oprea Laurentiu-Cristian
Gr 242
Proiect SGBD

Introducere

In acest proiect se vor furniza informatii despre **administrarea magazinelor online.**

Restrictii de functionare

- Pentru un magazin lucreaza unul sau mai multi angajati.
- Un angajat lucreaza pentru un magazin.
- Un magazin poate avea sau nu sedii.
- Un sediu este pazit de unul sau mai multi angajati de tip paznic.
- O reclama poate fi realizata de mai multe echipe de marketing si promoveaza unul sau mai multe magazine, iar un magazin poate avea mai multe reclame.
- Reclama este fie imagine, fie video.
- Pe o pagina web se pot vinde produsele mai multor distribuitori, pe mai multe magazine, iar un magazin poate avea mai multe pagini web.
- O pagina web este scrisa de unul sau mai multi programatori.
- La firma de IT lucreaza unul sau mai multi programatori.
- O pagina web este hostata pe un server.
- Firma IT poate sau nu sa inchirieze servere.
- Clientul cumpara unul sau mai multe produse.
- Produsul poate fi cumparat de mai multi clienti.
- Produsele pot fi bunuri sau servicii.
- Bunurile sunt produse de cate un producator in una sau mai multe fabrici.
- Serviciile sunt prestate de unul sau mai multi muncitori.
- Un muncitor poate presta mai multe servicii.
- O fabrica poate lucra cu mai multi producatori.

Lista de entitati:

MAGAZIN, SEDIU, ANGAJAT, CURIER, CALL_CENTER, PAZNIC, MANAGER, CLIENT_, PRODUS, BUN, SERVICIU, DISTRIBUITOR, PRODUCATOR, MUNCITOR, FABRICA, PAGINA_WEB, SERVER, PROGRAMATOR, FIRMA_IT, RECLAMA, IMAGINE, VIDEO, ECHIPA_MARKETING.

CURIER, CALL_CENTER, PAZNIC, MANAGER, IMAGINE, VIDEO, BUN, SERVICIU reprezinta subentitati in modelul relational.

Diagrama E-R

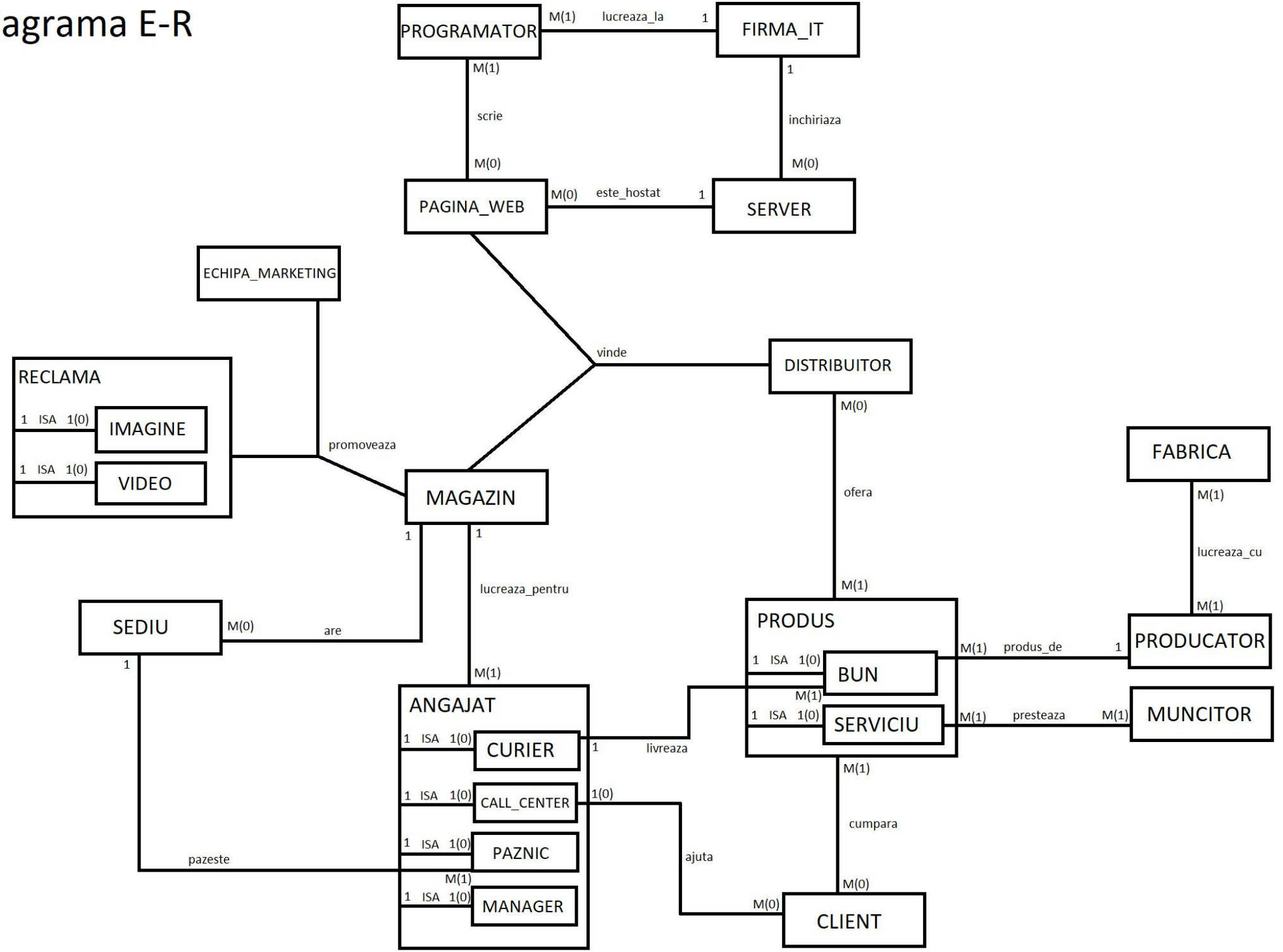
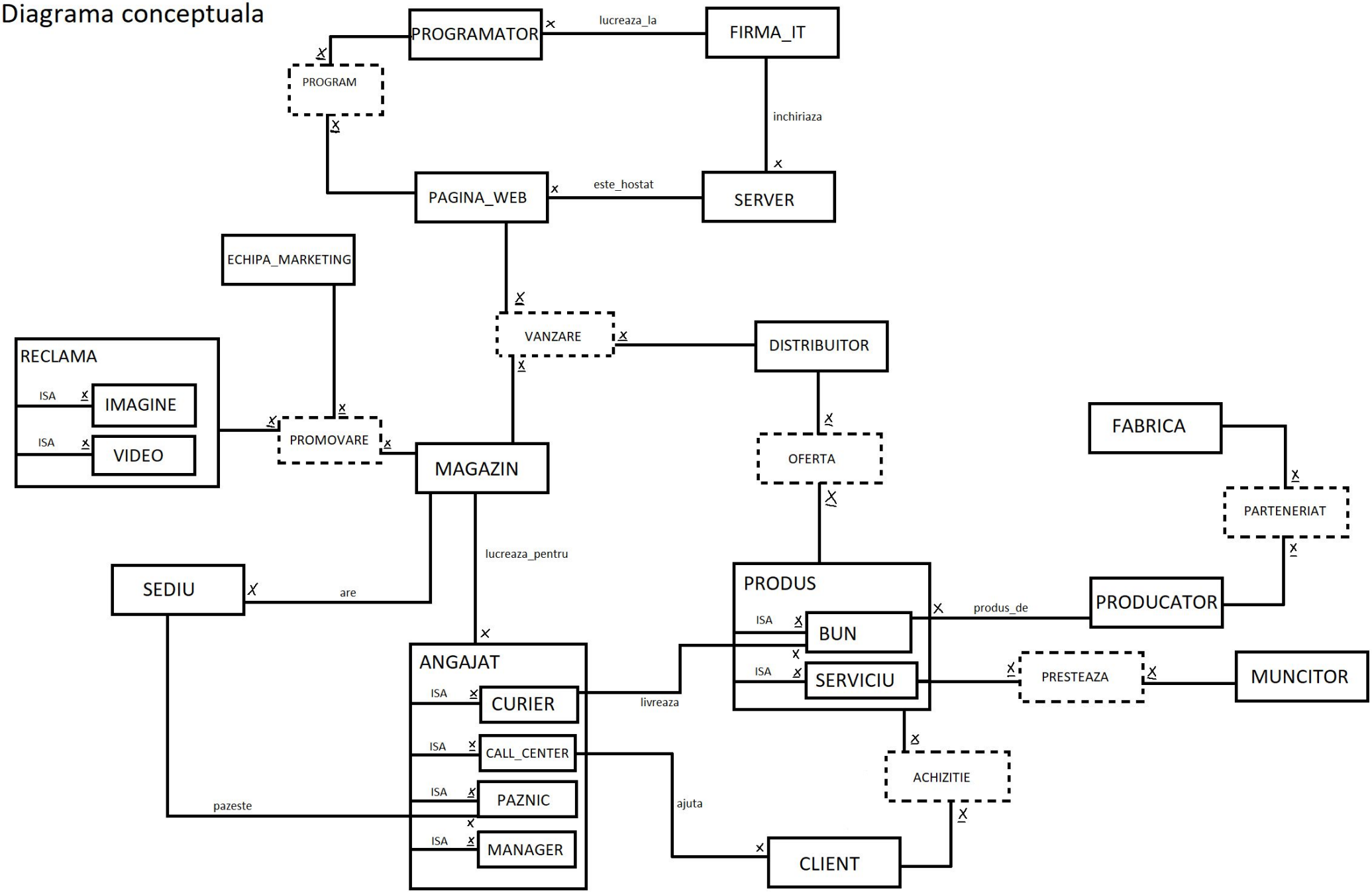

PROGRAMATOR
M(1)
lucreaza_la
1
FIRMA_IT
M(1)
scrie
M(0)
1
inchiriaza
M(0)
PAGINA_WEB
M(0)
este_hostat
1
SERVER
ECHIPA_MARKETING
vinde
DISTRIBUITOR
RECLAMA
1 ISA 1(0)
IMAGINE
1 ISA 1(0)
VIDEO
promoveaza
MAGAZIN
M(0)
ofera
M(1)
FABRICA
M(1)
lucreaza_cu
M(1)
1
1
lucreaza_pentru
SEDIU
M(0)
are
1
M(1)
PRODUS
1 ISA 1(0)
BUN
M(1)
1 ISA 1(0)
SERVICIU
M(1)
produs_de
1
PRODUCATOR
M(1)
presteaza
M(1)
MUNCITOR
ANGAJAT
1 ISA 1(0)
CURIER
1
livreaza
1 ISA 1(0)
CALL_CENTER
1(0)
1 ISA 1(0)
PAZNIC
M(1)
1 ISA 1(0)
MANAGER
pazeste
ajuta
M(1)
cumpara
M(0)
M(0)
CLIENT

Diagrama conceptuala



Atribute

Entitatea MAGAZIN are ca atribute:

- id_magazin =variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare a magazinului
- nume= variabila de tip caracter, de lungime maxima 30

Entitatea SEDIU are ca atribute:

- id_sediu = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare a sediului
- adresa = variabila de tip caracter, de lungime maxima 100, reprezentand adresa fizica a sediului
- id_magazin

Entitatea ANGAJAT are ca atribute:

- cod_angajat = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al angajatului in magazin
- id_magazin
- tip_angajat= poate fi curier, call_center, paznic sau manager
- data_angajare= variabila de tip data, in format DD/MM/YYYY
- id_sediu
- nume

Subentitatea CURIER are ca atribute:

- cod_angajat

Subentitatea CALL_CENTER are ca atribute:

- cod_angajat

Subentitatea PAZNIC are ca atribute:

- cod_angajat
- id_sediu

Subentitatea MANAGER are ca atribute:

- cod_angajat

Entitatea CLIENT_ are ca atribute:

- id_client = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al clientului
- cod_angajat = codul angajatului de tip call_center care ajuta respectivul client

- platit = variabila de tip numar real, reprezentand suma totala platita de client pe produse cumparate in trecut

Entitatea PRODUS are ca atribute:

- cod_produs = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al produsului
- tip_produs= bun sau serviciu
- pret = variabila de tip numar real
- cod_curier = codul curierului care livreaza acest produs

Subentitatea BUN are ca atribute:

- cod_produs
- cod_angajat = codul curierului care livreaza bunul respectiv
- id_producator = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al producatorului bunului respectiv
- nume

Subentitatea SERVICIU are ca atribute:

- cod_produs
- nume

Entitatea DISTRIBUITOR are ca atribute:

- id_distribuitor = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al distribuitorului
- nume
- data_infiintare = variabila de tip data

Entitatea PRODUCATOR are ca atribute:

- id_producator
- nume
- data_infiintare = variabila de tip data

Entitatea MUNCITOR are ca atribute:

- cnp = variabila de tip numar de lungime 13; codul numeric personal al muncitorului
- data_nastere = variabila de tip data
- an_inceput_munca= variabila de tip numar de lungime 4, reprezentand anul din care muncitorul a inceput sa presteze serviciul
- nume
- prenume = variabila de tip caracter, lungime maxima 30

Entitatea FABRICA are ca atribute:

- cod_fabrica
- nume
- adresa = variabila de tip caracter, de lungime maxima 100, reprezentand adresa fizica a fabricii

Entitatea PAGINA_WEB are ca atribute:

- link_pagina = link de o lungime maxima de 100 de caractere
- an_creare = variabila de tip numar de lungime 4, reprezentand anul din care este activa pagina web
- id_server

Entitatea SERVER are ca atribute:

- id_server = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al serverului in baza de date a firmei ce il hosteaza
- maxsize = variabila de tip intreg de lungime maxima 12, reprezentand numarul maxim de pagini web ce pot fi hostate pe acel server
- id_firma

Entitatea PROGRAMATOR are ca atribute:

- cod_programator = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al angajatului(programatorului) in firma IT
- limbaje_cunoscute = **variabila de tip caracter, lungime maxima 500; reprezinta o lista de limbaje de programare cunoscute de catre programator preluata din CV-ul acestuia (nu este inca validat, el poate sa puna ce vrea acolo), de aceea este un string ci nu un tabel**
- id_firma

Entitatea FIRMA_IT are ca atribute:

- id_firma = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare a firmei
- nr_angajati = variabila de tip numar, lungime maxima 7

Entitatea RECLAMA are ca atribute:

- url_reclama =variabila de tip caracter, lungime maxima 100; link-ul generat de reclama catre site-ul pe care il promoveaza
- tip_reclama = poate fi imagine sau video

Subentitatea IMAGINE are ca atribute:

- url_reclama

- height = variabila de tip numar, lungime maxima 5, reprezinta inaltimea imaginii in pixeli
- width = variabila de tip numar, lungime maxima 5, reprezinta latimea imaginii in pixeli

Subentitatea VIDEO are ca atribute:

- url_reclama
- len = variabila de tip numar, lungime maxima 4; reprezinta durata in secunde a videoclipului
- width
- height
- quality = variabila de tip caracter, lungime maxima 5; poate fi 140p, 480p, 720p, 1080p

Entitatea ECHIPA_MARKETING are ca atribute:

- id_firma_marketing = variabila de tip intreg, de lungime maxima 7, reprezentand codul unic de identificare al firmei de marketing
- nr_angajati

4.

```
CREATE TABLE MAGAZIN(
    id_magazin number(7),
    nume varchar2(30),
    CONSTRAINT pk_mag primary key(id_magazin)
    );

CREATE TABLE SEDIU(
    id_sediu number(7),
    adresa varchar2(100),
    id_magazin number(7),
    CONSTRAINT pk_sediu primary key(id_sediu),
    CONSTRAINT fk_sediu_mag foreign key(id_magazin) references MAGAZIN(id_magazin)
    );

CREATE TABLE ANGAJAT(
    nume varchar2(30),
    cod_angajat number(7),
    id_magazin number(7),
    tip_angajat varchar2(20) CONSTRAINT tip_nn not null,
    data_angajare date,
    id_sediu number(7),
    CONSTRAINT pk_angajat primary key(cod_angajat),
    CONSTRAINT fk_ang_mag foreign key(id_magazin) references MAGAZIN(id_magazin),
    CONSTRAINT fk_paznic foreign key(id_sediu) references SEDIU(id_sediu)
    );

CREATE TABLE CLIENT_(
    id_client number(7),
```

```
    cod_angajat number(7),
    platit number(10,2),
    CONSTRAINT pk_client primary key(id_client),
    CONSTRAINT fk_client_callcenter foreign key(cod_angajat) references
ANGAJAT(cod_angajat)
    );

CREATE TABLE PRODUCATOR(
    id_producator number(7),
    nume varchar2(30),
    data_infiintare date,
    CONSTRAINT pk_producator primary key(id_producator)
    );



CREATE TABLE PRODUS(
    cod_produs number(7),
    tip_produs varchar2 (10) constraint produs_nn not null,
    pret number(10,2),
    cod_curier number(7),
    id_producator number(7),
    nume varchar2(30),
    CONSTRAINT pk_produs primary key(cod_produs),
    CONSTRAINT fk_prod_curier foreign key(cod_curier) references ANGAJAT(cod_angajat),
    CONSTRAINT fk_prod_produc foreign key(id_producator) references
PRODUCATOR(id_producator)
    );

CREATE TABLE DISTRIBUITOR(
    id_distribuitor number(7),
    nume varchar2(30),
    data_infiintare date,
    CONSTRAINT pk_distr primary key(id_distribuitor)
    );
```

```
CREATE TABLE MUNCITOR(
    cnp number(13),
    data_nastere date,
    an_inceput_munca number(4),
    nume varchar2(30),
    prenume varchar2(30),
    CONSTRAINT pk_munc primary key(cnp)
    );


CREATE TABLE FABRICA(
    cod_fabrica number(7),
    nume varchar2(30),
    adresa varchar2(100),
    CONSTRAINT pk_fabrica primary key(cod_fabrica)
    );

CREATE TABLE FIRMA_IT(
    id_firma number(7),
    nr_angajati number(7),
    CONSTRAINT pk_firma primary key(id_firma)
    );


CREATE TABLE SERVER(
    id_server number(7),
    id_firma number(7),
    maxsize number(12),
    CONSTRAINT pk_server primary key(id_server),
    CONSTRAINT fk_server_firma foreign key(id_firma) references FIRMA_IT(id_firma)
    );
```

```
CREATE TABLE PAGINA_WEB(
    link_pagina varchar2(100),
    an_creare number(4),
    id_server number(7),
    CONSTRAINT pk_pag_web primary key(link_pagina),
    CONSTRAINT fk_pag_web_server foreign key(id_server) references SERVER(id_server)
    );

CREATE TABLE PROGRAMATOR(
    cod_programator number(7),
    limbaje_cunoscute varchar2(500),
    id_firma number(7),
    CONSTRAINT pk_programator primary key(cod_programator),
    CONSTRAINT fk_prog_firma foreign key(id_firma) references FIRMA_IT(id_firma)
    );

CREATE TABLE ECHIPA_MARKETING(
    id_firma_marketing number(7),
    nr_angajati number(7),
    CONSTRAINT pk_ec_mark primary key(id_firma_marketing)
    );


CREATE TABLE RECLAMA(
    url_reclama varchar2(100),
    tip_reclama varchar2(30) not null,
    height number(5) not null,
    width number(5)not null,
    len number(4),
    quality varchar2(5),
    CONSTRAINT pk_reclama primary key(url_reclama)
    );
```

```
CREATE TABLE PROGRAM_(
    cod_programator number(7),
    link_pagina varchar2(100),
    CONSTRAINT pk_compus_program primary key(cod_programator,link_pagina),
    CONSTRAINT fk_program_prog foreign key(cod_programator) references
PROGRAMATOR(cod_programator),
    CONSTRAINT fk_program_pagina foreign key(link_pagina) references
PAGINA_WEB(link_pagina)
    );


CREATE TABLE VANZARE(
    id_magazin number(7),
    id_distribuitor number(7),
    link_pagina varchar2(100),
    CONSTRAINT pk_compus_vanzare primary key(id_magazin,id_distribuitor,link_pagina),
    CONSTRAINT fk_vanzare_magazin foreign key(id_magazin) references MAGAZIN(id_magazin),
    CONSTRAINT fk_vanzare_distr foreign key(id_distribuitor) references
DISTRIBUITOR(id_distribuitor),
    CONSTRAINT fk_vanzare_pagina foreign key(link_pagina) references
PAGINA_WEB(link_pagina)
    );


CREATE TABLE PROMOVARE(
    id_magazin number(7),
    id_firma_marketing number(7),
    url_reclama varchar2(100),
    CONSTRAINT pk_compus_promovare primary key(id_magazin, id_firma_marketing,
url_reclama),
    CONSTRAINT fk_promo_magazin foreign key(id_magazin) references MAGAZIN(id_magazin),
    CONSTRAINT fk_promo_firma_market foreign key(id_firma_marketing) references
ECHIPA_MARKETING(id_firma_marketing),
    CONSTRAINT fk_promo_reclama foreign key(url_reclama) references RECLAMA(url_reclama)
    );


CREATE TABLE OFERTA(
```

```
    cod_produs number(7),
    id_distribuitor number(7),
    CONSTRAINT pk_compus_oferta primary key(cod_produs,id_distribuitor),
    CONSTRAINT fk_oferta_produs foreign key(cod_produs) references PRODUS(cod_produs),
    CONSTRAINT fk_oferta_dist foreign key(id_distribuitor) references
DISTRIBUITOR(id_distribuitor)
    );

CREATE TABLE ACHIZITIE(
    id_client number(7),
    cod_produs number(7),
    suma number(10),
    CONSTRAINT pk_achizitie primary key(id_client, cod_produs),
    CONSTRAINT fk_ac_cli foreign key(id_client) references CLIENT_(id_client),
    CONSTRAINT fk_ac_prod foreign key(cod_produs) references PRODUS(cod_produs)
    );

CREATE TABLE PRESTEAZA(
    cod_produs number(7),
    cnp number(13),
    CONSTRAINT pk_presteaza primary key(cod_produs, cnp),
    CONSTRAINT fk_pres_prod foreign key(cod_produs) references PRODUS(cod_produs),
    CONSTRAINT fk_pres_munc foreign key(cnp) references MUNCITOR(cnp)
);

CREATE TABLE PARTENERIAT(
    id_producator number(7),
    cod_fabrica number(7),
    data_inceput date,
    CONSTRAINT pk_parteneriat primary key(id_producator, cod_fabrica),
    CONSTRAINT fk_part_prod foreign key(id_producator) references
PRODUCATOR(id_producator),
    CONSTRAINT fk_part_fabrica foreign key(cod_fabrica) references FABRICA(cod_fabrica)
);
```

```
commit;
```

proiect~1.sql

0.13 seconds

Worksheet | Query Builder

```
    id_client number(7),
    cod_produs number(7),
    suma number(10),
    CONSTRAINT pk_achizitie primary key(id_client, cod_produs),
    CONSTRAINT fk_ac_cli foreign key(id_client) references CLIENT_(id_client),
    CONSTRAINT fk_ac_prod foreign key(cod_produs) references PRODUS(cod_produs)
    );

CREATE TABLE PRESTEAZA(
    cod_produs number(7),
    cnp number(13),
    CONSTRAINT pk_presteaza primary key(cod_produs, cnp),
    CONSTRAINT fk_pres_prod foreign key(cod_produs) references PRODUS(cod_produs),
    CONSTRAINT fk_pres_munc foreign key(cnp) references MUNCITOR(cnp)
);

CREATE TABLE PARTENERIAT(
    id_producator number(7),
    cod_fabrica number(7),
    data_inceput date,
    CONSTRAINT pk_parteneriat primary key(id_producator, cod_fabrica),
    CONSTRAINT fk_part_prod foreign key(id_producator) references PRODUCATOR(id_producator),
    CONSTRAINT fk_part_fabrica foreign key(cod_fabrica) references FABRICA(cod_fabrica)
);

commit;
```

Script Output

Task completed in 0.13 seconds

```
Table ACHIZITIE created.

Table PRESTEAZA created.

Table PARTENERIAT created.

Commit complete.
```

- ACHIZITIE
- ANGAJAT
- CLIENT_
- DISTRIBUITOR
- ECHIPA_MARKETING
- FABRICA
- FIRMA_IT
- MAGAZIN
- MUNCITOR
- OFERTA
- PAGINA_WEB
- PARTENERIAT
- PRESTEAZA
- PRODUCATOR
- PRODUS
- PROGRAM_
- PROGRAMATOR
- PROMOVARE
- RECLAMA
- SEDIU
- SERVER
- VANZARE

proiect4.sql | ACHIZITIE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_CLIENT	NUMBER(7,0)	No	(null)	1	(null)
2	COD_PRODUS	NUMBER(7,0)	No	(null)	2	(null)
3	SUMA	NUMBER(10,0)	Yes	(null)	3	(null)

proiect4.sql | ANGAJAT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	NUME	VARCHAR2(30 BYTE)	Yes	(null)	1	(null)
2	COD_ANGAJAT	NUMBER(7,0)	No	(null)	2	(null)
3	ID_MAGAZIN	NUMBER(7,0)	Yes	(null)	3	(null)
4	TIP_ANGAJAT	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	DATA_ANGAJARE	DATE	Yes	(null)	5	(null)
6	ID_SEDIU	NUMBER(7,0)	Yes	(null)	6	(null)

proiect4.sql | CLIENT_

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_CLIENT	NUMBER(7,0)	No	(null)	1	(null)
2	COD_ANGAJAT	NUMBER(7,0)	Yes	(null)	2	(null)
3	PLATIT	NUMBER(10,2)	Yes	(null)	3	(null)

proiect4.sql | DISTRIBUITOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_DISTRIBUITOR	NUMBER(7,0)	No	(null)	1	(null)
2	NUME	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3	DATA_INFIINTARE	DATE	Yes	(null)	3	(null)

proiect4.sql | ECHIPA_MARKETING

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_FIRMA_MARKETING	NUMBER(7,0)	No	(null)	1	(null)
2	NR_ANGAJATI	NUMBER(7,0)	Yes	(null)	2	(null)

proiect4.sql | FABRICA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_FABRICA	NUMBER(7,0)	No	(null)	1	(null)
2	NUME	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3	ADRESA	VARCHAR2(100 BYTE)	Yes	(null)	3	(null)

proiect4.sql | FIRMA_IT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_FIRMA	NUMBER(7,0)	No	(null)	1	(null)
2	NR_ANGAJATI	NUMBER(7,0)	Yes	(null)	2	(null)

proiect4.sql | MAGAZIN

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MAGAZIN	NUMBER(7,0)	No	(null)	1	(null)
2	NUME	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)

proiect4.sql | MUNCITOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CNP	NUMBER(13,0)	No	(null)	1	(null)
2	DATA_NASTERE	DATE	Yes	(null)	2	(null)
3	AN_INCEPUT_MUNCA	NUMBER(4,0)	Yes	(null)	3	(null)
4	NUME	VARCHAR2(30 BYTE)	Yes	(null)	4	(null)
5	PRENUME	VARCHAR2(30 BYTE)	Yes	(null)	5	(null)

proiect4.sql | OFERTA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PRODUS	NUMBER(7,0)	No	(null)	1	(null)
2	ID_DISTRIBUITOR	NUMBER(7,0)	No	(null)	2	(null)

proiect4.sql | PAGINA_WEB

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LINK_PAGINA	VARCHAR2(100 BYTE)	No	(null)	1	(null)
2	AN_CREARE	NUMBER(4,0)	Yes	(null)	2	(null)
3	ID_SERVER	NUMBER(7,0)	Yes	(null)	3	(null)

proiect4.sql | PARTENERIAT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PRODUCATOR	NUMBER(7,0)	No	(null)	1	(null)
2	COD_FABRICA	NUMBER(7,0)	No	(null)	2	(null)
3	DATA_INCEPUT	DATE	Yes	(null)	3	(null)

proiect4.sql | PRESTEAZA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PRODUS	NUMBER(7,0)	No	(null)	1	(null)
2	CNP	NUMBER(13,0)	No	(null)	2	(null)

proiect4.sql | PRODUCATOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PRODUCATOR	NUMBER(7,0)	No	(null)	1	(null)
2	NUME	VARCHAR2(30 BYTE)	Yes	(null)	2	(null)
3	DATA_INFIINTARE	DATE	Yes	(null)	3	(null)

proiect4.sql | PRODUS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PRODUS	NUMBER(7,0)	No	(null)	1	(null)
2	TIP_PRODUS	VARCHAR2(10 BYTE)	No	(null)	2	(null)
3	PRET	NUMBER(10,2)	Yes	(null)	3	(null)
4	COD_CURIER	NUMBER(7,0)	Yes	(null)	4	(null)
5	ID_PRODUCATOR	NUMBER(7,0)	Yes	(null)	5	(null)
6	NUME	VARCHAR2(30 BYTE)	Yes	(null)	6	(null)

proiect4.sql | PROGRAM_

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PROGRAMATOR	NUMBER(7,0)	No	(null)	1	(null)
2	LINK_PAGINA	VARCHAR2(100 BYTE)	No	(null)	2	(null)

proiect4.sql | PROGRAMATOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	COD_PROGRAMATOR	NUMBER(7,0)	No	(null)	1	(null)
2	LIMBAJE_CUNOSCUTE	VARCHAR2(500 BYTE)	Yes	(null)	2	(null)
3	ID_FIRMA	NUMBER(7,0)	Yes	(null)	3	(null)

proiect4.sql | PROMOVARE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MAGAZIN	NUMBER(7,0)	No	(null)	1	(null)
2	ID_FIRMA_MARKETING	NUMBER(7,0)	No	(null)	2	(null)
3	URL_RECLAMA	VARCHAR2(100 BYTE)	No	(null)	3	(null)

proiect4.sql | RECLAMA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	URL_RECLAMA	VARCHAR2(100 BYTE)	No	(null)	1	(null)
2	TIP_RECLAMA	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3	HEIGHT	NUMBER(5,0)	No	(null)	3	(null)
4	WIDTH	NUMBER(5,0)	No	(null)	4	(null)
5	LEN	NUMBER(4,0)	Yes	(null)	5	(null)
6	QUALITY	VARCHAR2(5 BYTE)	Yes	(null)	6	(null)

proiect4.sql | SEDIU

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_SEDIU	NUMBER(7,0)	No	(null)	1	(null)
2	ADRESA	VARCHAR2(100 BYTE)	Yes	(null)	2	(null)
3	ID_MAGAZIN	NUMBER(7,0)	Yes	(null)	3	(null)

proiect4.sql | SERVER

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_SERVER	NUMBER(7,0)	No	(null)	1	(null)
2	ID_FIRMA	NUMBER(7,0)	Yes	(null)	2	(null)
3	MAXSIZE	NUMBER(12,0)	Yes	(null)	3	(null)

proiect4.sql | VANZARE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MAGAZIN	NUMBER(7,0)	No	(null)	1	(null)
2	ID_DISTRIBUITOR	NUMBER(7,0)	No	(null)	2	(null)
3	LINK_PAGINA	VARCHAR2(100 BYTE)	No	(null)	3	(null)

5.

```
INSERT INTO MAGAZIN VALUES(1, 'eMag');

INSERT INTO MAGAZIN VALUES(2, 'Cel.ro');

INSERT INTO MAGAZIN VALUES(3, 'Altex');

INSERT INTO MAGAZIN VALUES(4, 'Flanco');

INSERT INTO MAGAZIN VALUES(5, 'PcGarage');


INSERT INTO SEDIU VALUES(7,'Bvd Iuliu Maniu 200',1);

INSERT INTO SEDIU VALUES(8,'Timpuri Noi 24',2);

INSERT INTO SEDIU VALUES(9,'Tineretului 210',3);

INSERT INTO SEDIU VALUES(10,'Lipscani 3',4);

INSERT INTO SEDIU VALUES(11,'Nordului 55',5);


INSERT INTO ANGAJAT VALUES(1,1,'manager',TO_DATE('23/08/2001','DD/MM/YYYY'),null,'Rayyan
Holmes');

INSERT INTO ANGAJAT VALUES(2,1,'curier',TO_DATE('21/12/2015','DD/MM/YYYY'),null,'Ayub
Shannon');

INSERT INTO ANGAJAT VALUES(3,1,'curier',TO_DATE('04/07/2013','DD/MM/YYYY'),null,'Bobbie
Rollins');

INSERT INTO ANGAJAT VALUES(4,1,'curier',TO_DATE('23/12/2007','DD/MM/YYYY'),null,'Sabiha
Li');

INSERT INTO ANGAJAT VALUES(5,1,'paznic',TO_DATE('07/03/2014','DD/MM/YYYY'),7,'Devon
Higgins');

INSERT INTO ANGAJAT VALUES(6,1,'call-
center',TO_DATE('31/10/2002','DD/MM/YYYY'),null,'Josef Cobb');

INSERT INTO ANGAJAT
VALUES(7,2,'manager',TO_DATE('17/06/2014','DD/MM/YYYY'),null,'Winifred Cornish');

INSERT INTO ANGAJAT VALUES(8,2,'curier',TO_DATE('28/06/2008','DD/MM/YYYY'),null,'Elen
Avalos');

INSERT INTO ANGAJAT VALUES(9,2,'paznic',TO_DATE('23/10/2001','DD/MM/YYYY'),8,'Usamah
Hastings');

INSERT INTO ANGAJAT VALUES(10,2,'paznic',TO_DATE('27/04/2005','DD/MM/YYYY'),8,'Nahla
Duke');
```

```
INSERT INTO ANGAJAT VALUES(11,2,'call-
center',TO_DATE('29/04/2020','DD/MM/YYYY'),null,'Jade Douglas');

INSERT INTO ANGAJAT
VALUES(12,3,'manager',TO_DATE('11/01/2005','DD/MM/YYYY'),null,'Kameron Haigh');

INSERT INTO ANGAJAT VALUES(13,3,'curier',TO_DATE('19/11/2015','DD/MM/YYYY'),null,'Ashley
Povey');

INSERT INTO ANGAJAT VALUES(14,3,'curier',TO_DATE('08/01/2005','DD/MM/YYYY'),null,'Ellie-
Mai Erickson');

INSERT INTO ANGAJAT VALUES(15,3,'paznic',TO_DATE('09/06/2009','DD/MM/YYYY'),9,'Javan
Ashton');

INSERT INTO ANGAJAT VALUES(16,3,'call-
center',TO_DATE('10/01/2005','DD/MM/YYYY'),null,'Alistair Aldred');

INSERT INTO ANGAJAT VALUES(17,4,'manager',TO_DATE('11/08/2002','DD/MM/YYYY'),null,'Rafi
Sellers');

INSERT INTO ANGAJAT VALUES(18,4,'curier',TO_DATE('19/09/2017','DD/MM/YYYY'),null,'Rikki
Bouvet');

INSERT INTO ANGAJAT VALUES(19,4,'paznic',TO_DATE('28/05/2018','DD/MM/YYYY'),10,'Precious
Tait');

INSERT INTO ANGAJAT VALUES(20,4,'call-
center',TO_DATE('20/04/2016','DD/MM/YYYY'),null,'Elize Devlin');

INSERT INTO ANGAJAT VALUES(21,4,'call-
center',TO_DATE('09/08/2016','DD/MM/YYYY'),null,'Eva Kelley');

INSERT INTO ANGAJAT
VALUES(22,5,'manager',TO_DATE('25/07/2003','DD/MM/YYYY'),null,'Steffan King');

INSERT INTO ANGAJAT
VALUES(23,5,'curier',TO_DATE('30/03/2004','DD/MM/YYYY'),null,'Courtney Wright');

INSERT INTO ANGAJAT VALUES(24,5,'curier',TO_DATE('10/04/2012','DD/MM/YYYY'),null,'Jo
Leal');

INSERT INTO ANGAJAT VALUES(25,5,'paznic',TO_DATE('23/08/2018','DD/MM/YYYY'),11,'Osman
Grainger');

INSERT INTO ANGAJAT VALUES(26,5,'call-
center',TO_DATE('19/09/2018','DD/MM/YYYY'),null,'Vera Hibbert');

INSERT INTO ANGAJAT VALUES(27,5,'call-
center',TO_DATE('25/01/2006','DD/MM/YYYY'),null,'Justine Franks');

ALTER TABLE ANGAJAT ADD salariu NUMBER;

UPDATE ANGAJAT SET salariu = 1500 WHERE tip_angajat = 'paznic';

UPDATE ANGAJAT SET salariu = 2500 WHERE tip_angajat = 'call-center';

UPDATE ANGAJAT SET salariu = 3500 WHERE tip_angajat = 'curier';

UPDATE ANGAJAT SET salariu = 7500 WHERE tip_angajat = 'manager';
```

```
INSERT INTO PRODUCATOR VALUES(1,'Producator1',TO_DATE('26/07/2002','DD/MM/YYYY'));
INSERT INTO PRODUCATOR VALUES(2,'Producator2',TO_DATE('03/05/2005','DD/MM/YYYY'));
INSERT INTO PRODUCATOR VALUES(3,'Producator3',TO_DATE('03/11/2005','DD/MM/YYYY'));
INSERT INTO PRODUCATOR VALUES(4,'Producator4',TO_DATE('04/08/2016','DD/MM/YYYY'));
INSERT INTO PRODUCATOR VALUES(5,'Producator5',TO_DATE('06/04/2018','DD/MM/YYYY'));
INSERT INTO PRODUCATOR VALUES(6,'Producator6',TO_DATE('08/11/2019','DD/MM/YYYY'));

INSERT INTO PRODUS VALUES(1,'bun',6000,2,1,'placa video');
INSERT INTO PRODUS VALUES(2,'bun',1000,2,2,'procesor');
INSERT INTO PRODUS VALUES(3,'bun',1500,3,3,'masina de spalat');
INSERT INTO PRODUS VALUES(4,'bun',1800,4,4,'aer conditionat');
INSERT INTO PRODUS VALUES(5,'bun',5,8,5,'o sticla de pepsi');
INSERT INTO PRODUS VALUES(6,'bun',300,13,6,'aspirator');
INSERT INTO PRODUS VALUES(7,'bun',200,14,5,'masina de ras');
INSERT INTO PRODUS VALUES(8,'bun',840,14,4,'licenta windows 10');
INSERT INTO PRODUS VALUES(9,'bun',50,14,3,'joc pe cd GTA');
INSERT INTO PRODUS VALUES(10,'bun',30,18,2,'cablu internet');
INSERT INTO PRODUS VALUES(11,'bun',10,23,1,'bec economic');
INSERT INTO PRODUS VALUES(12,'serviciu',100,null,null,'curatare aparat');
INSERT INTO PRODUS VALUES(13,'serviciu',150,null,null,'asamblare calculator');
INSERT INTO PRODUS VALUES(14,'serviciu',200,null,null,'reparatie calculator');
INSERT INTO PRODUS VALUES(15,'serviciu',50,null,null,'mentenanta');
INSERT INTO PRODUS VALUES(16,'serviciu',80,null,null,'soft-update calculator');

INSERT INTO CLIENT_ VALUES(1,6,100);
INSERT INTO CLIENT_ VALUES(2,6,1);
INSERT INTO CLIENT_ VALUES(3,11,1550);
INSERT INTO CLIENT_ VALUES(4,16,500);
INSERT INTO CLIENT_ VALUES(5,20,4490);
INSERT INTO CLIENT_ VALUES(6,20,5432);
INSERT INTO CLIENT_ VALUES(7,20,1234);
```

```
INSERT INTO CLIENT_ VALUES(8,27,556);

INSERT INTO ACHIZITIE VALUES(1,1,null);
INSERT INTO ACHIZITIE VALUES(1,2,null);
INSERT INTO ACHIZITIE VALUES(1,8,null);
INSERT INTO ACHIZITIE VALUES(1,13,null);
INSERT INTO ACHIZITIE VALUES(2,3,null);
INSERT INTO ACHIZITIE VALUES(2,4,null);
INSERT INTO ACHIZITIE VALUES(3,5,null);
INSERT INTO ACHIZITIE VALUES(4,3,null);
INSERT INTO ACHIZITIE VALUES(4,4,null);
INSERT INTO ACHIZITIE VALUES(4,6,null);
INSERT INTO ACHIZITIE VALUES(4,7,null);
INSERT INTO ACHIZITIE VALUES(4,11,null);
INSERT INTO ACHIZITIE VALUES(5,4,null);
INSERT INTO ACHIZITIE VALUES(5,12,null);
INSERT INTO ACHIZITIE VALUES(6,16,null);
INSERT INTO ACHIZITIE VALUES(7,14,null);
INSERT INTO ACHIZITIE VALUES(7,16,null);
INSERT INTO ACHIZITIE VALUES(8,10,null);
INSERT INTO ACHIZITIE VALUES(8,15,null);
INSERT INTO ACHIZITIE VALUES(8,16,null);

INSERT INTO MUNCITOR
VALUES(2881224098035,TO_DATE('24/12/1988','DD/MM/YYYY'),2011,'Pompiliu','Olga');
INSERT INTO MUNCITOR
VALUES(1940223365590,TO_DATE('23/02/1994','DD/MM/YYYY'),2015,'Horatiu','Andrei');
INSERT INTO MUNCITOR
VALUES(1871205464650,TO_DATE('05/12/1987','DD/MM/YYYY'),2013,'Teodor','Florin');
INSERT INTO MUNCITOR
VALUES(1890313140275,TO_DATE('13/03/1989','DD/MM/YYYY'),2017,'David','Victor');
INSERT INTO MUNCITOR
VALUES(1860909035887,TO_DATE('09/09/1986','DD/MM/YYYY'),2006,'Dacian','Felix');

INSERT INTO PRESTEAZA VALUES(12,2881224098035);
```

```
INSERT INTO PRESTEAZA VALUES(13,2881224098035);
INSERT INTO PRESTEAZA VALUES(14,2881224098035);
INSERT INTO PRESTEAZA VALUES(15,2881224098035);
INSERT INTO PRESTEAZA VALUES(16,2881224098035);
INSERT INTO PRESTEAZA VALUES(15,1940223365590);
INSERT INTO PRESTEAZA VALUES(16,1940223365590);
INSERT INTO PRESTEAZA VALUES(13,1871205464650);
INSERT INTO PRESTEAZA VALUES(15,1871205464650);
INSERT INTO PRESTEAZA VALUES(16,1871205464650);
INSERT INTO PRESTEAZA VALUES(15,1890313140275);
INSERT INTO PRESTEAZA VALUES(13,1860909035887);
INSERT INTO PRESTEAZA VALUES(14,1860909035887);
INSERT INTO PRESTEAZA VALUES(15,1860909035887);
INSERT INTO PRESTEAZA VALUES(16,1860909035887);

INSERT INTO FABRICA VALUES(1,'Fabrica de calculatoare','Grivitei 355');
INSERT INTO FABRICA VALUES(2,'Fabrica de mancare','Berceni 30');
INSERT INTO FABRICA VALUES(3,'Fabrica de becuri','Brancoveanu 12');
INSERT INTO FABRICA VALUES(4,'Fabrica de cabluri','Pantelimon 123');
INSERT INTO FABRICA VALUES(5,'Fabrica de laptopuri','Timisoarei 10');
INSERT INTO FABRICA VALUES(6,'Fabrica de frigidere','Giurgiului 300');

INSERT INTO PARTENERIAT VALUES(1,1,TO_DATE('14/06/2010','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(1,2,TO_DATE('08/07/2011','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(1,3,TO_DATE('11/07/2012','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(2,4,TO_DATE('26/08/2014','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(2,5,TO_DATE('01/01/2015','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(3,6,TO_DATE('01/01/2016','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(4,4,TO_DATE('08/04/2016','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(4,6,TO_DATE('27/02/2017','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(5,3,TO_DATE('22/05/2017','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(5,4,TO_DATE('16/05/2018','DD/MM/YYYY'));
INSERT INTO PARTENERIAT VALUES(5,5,TO_DATE('01/03/2019','DD/MM/YYYY'));
```

```
INSERT INTO PARTENERIAT VALUES(6,1,TO_DATE('11/10/2019','DD/MM/YYYY'));

INSERT INTO DISTRIBUITOR VALUES(1,'Nvidia',TO_DATE('10/04/2006','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(2,'AMD',TO_DATE('20/09/2000','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(3,'Pepsico',TO_DATE('04/04/2004','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(4,'Samsung',TO_DATE('15/12/2010','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(5,'MancareSRL',TO_DATE('22/02/1997','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(6,'Bosch',TO_DATE('30/05/1990','DD/MM/YYYY'));
INSERT INTO DISTRIBUITOR VALUES(7,'Adidas',TO_DATE('22/01/2002','DD/MM/YYYY'));

INSERT INTO OFERTA VALUES(1,1);
INSERT INTO OFERTA VALUES(1,2);
INSERT INTO OFERTA VALUES(2,2);
INSERT INTO OFERTA VALUES(3,4);
INSERT INTO OFERTA VALUES(3,6);
INSERT INTO OFERTA VALUES(4,4);
INSERT INTO OFERTA VALUES(4,6);
INSERT INTO OFERTA VALUES(5,3);
INSERT INTO OFERTA VALUES(6,4);
INSERT INTO OFERTA VALUES(6,6);
INSERT INTO OFERTA VALUES(7,4);
INSERT INTO OFERTA VALUES(8,1);
INSERT INTO OFERTA VALUES(8,4);
INSERT INTO OFERTA VALUES(9,4);
INSERT INTO OFERTA VALUES(10,2);
INSERT INTO OFERTA VALUES(11,4);
INSERT INTO OFERTA VALUES(11,6);
INSERT INTO OFERTA VALUES(12,4);
INSERT INTO OFERTA VALUES(12,6);
INSERT INTO OFERTA VALUES(13,1);
INSERT INTO OFERTA VALUES(14,2);
INSERT INTO OFERTA VALUES(15,2);
INSERT INTO OFERTA VALUES(16,1);
```

```
INSERT INTO FIRMA_IT VALUES(1,100);
INSERT INTO FIRMA_IT VALUES(2,10);
INSERT INTO FIRMA_IT VALUES(3,1);
INSERT INTO FIRMA_IT VALUES(4,2000);
INSERT INTO FIRMA_IT VALUES(5,335);
INSERT INTO FIRMA_IT VALUES(6,14);
INSERT INTO FIRMA_IT VALUES(7,57);

INSERT INTO SERVER VALUES(1,1,10000);
INSERT INTO SERVER VALUES(2,2,26000);
INSERT INTO SERVER VALUES(3,1,1000);
INSERT INTO SERVER VALUES(4,2,5555);
INSERT INTO SERVER VALUES(5,3,4885);
INSERT INTO SERVER VALUES(6,3,123456);
INSERT INTO SERVER VALUES(7,4,5235);
INSERT INTO SERVER VALUES(8,4,100);
INSERT INTO SERVER VALUES(9,4,9500);
INSERT INTO SERVER VALUES(10,4,40);

INSERT INTO VANZARE VALUES(1,1,'emag/calculatoare');
INSERT INTO VANZARE VALUES(1,2,'emag/calculatoare');
INSERT INTO VANZARE VALUES(1,3,'emag/mancare');
INSERT INTO VANZARE VALUES(1,4,'emag/electrocasnice');
INSERT INTO VANZARE VALUES(1,4,'emag/mancare');
INSERT INTO VANZARE VALUES(1,5,'emag/mancare');
INSERT INTO VANZARE VALUES(1,7,'emag/electrocasnice');
INSERT INTO VANZARE VALUES(2,4,'cel.ro/electrocasnice');
INSERT INTO VANZARE VALUES(2,6,'cel.ro/electrocasnice');
INSERT INTO VANZARE VALUES(3,1,'altex/calculatoare');
INSERT INTO VANZARE VALUES(3,2,'altex/calculatoare');
INSERT INTO VANZARE VALUES(3,4,'altex/electrocasnice');
INSERT INTO VANZARE VALUES(3,6,'altex/electrocasnice');
```

```
INSERT INTO VANZARE VALUES(4,4,'flanco/electrocasnice');
INSERT INTO VANZARE VALUES(4,6,'flanco/electrocasnice');
INSERT INTO VANZARE VALUES(5,1,'pcgarage/calculatoare');
INSERT INTO VANZARE VALUES(5,2,'pcgarage/calculatoare');

INSERT INTO PAGINA_WEB VALUES('emag/electrocasnice',2009,1);
INSERT INTO PAGINA_WEB VALUES('emag/calculatoare',2014,1);
INSERT INTO PAGINA_WEB VALUES('altex/electrocasnice',2005,2);
INSERT INTO PAGINA_WEB VALUES('altex/calculatoare',2006,2);
INSERT INTO PAGINA_WEB VALUES('cel.ro/electrocasnice',2015,3);
INSERT INTO PAGINA_WEB VALUES('flanco/electrocasnice',2009,4);
INSERT INTO PAGINA_WEB VALUES('pcgarage/calculatoare',2009,5);
INSERT INTO PAGINA_WEB VALUES('emag/mancare',2005,6);

INSERT INTO PROGRAMATOR VALUES(1,'cpp',1);
INSERT INTO PROGRAMATOR VALUES(2,'sql',1);
INSERT INTO PROGRAMATOR VALUES(3,'javascript cpp',1);
INSERT INTO PROGRAMATOR VALUES(4,'python javascript cpp sql',2);
INSERT INTO PROGRAMATOR VALUES(5,'html css node',2);
INSERT INTO PROGRAMATOR VALUES(6,'sql node cpp javascript',3);
INSERT INTO PROGRAMATOR VALUES(7,'python',4);
INSERT INTO PROGRAMATOR VALUES(8,'sql',5);
INSERT INTO PROGRAMATOR VALUES(9,'html css',6);
INSERT INTO PROGRAMATOR VALUES(10,'html css javascript',7);

INSERT INTO PROGRAM_ VALUES(1,'emag/electrocasnice');
INSERT INTO PROGRAM_ VALUES(3,'emag/electrocasnice');
INSERT INTO PROGRAM_ VALUES(4,'emag/calculatoare');
INSERT INTO PROGRAM_ VALUES(1,'emag/mancare');
INSERT INTO PROGRAM_ VALUES(3,'altex/electrocasnice');
INSERT INTO PROGRAM_ VALUES(9,'altex/calculatoare');
INSERT INTO PROGRAM_ VALUES(4,'cel.ro/electrocasnice');
INSERT INTO PROGRAM_ VALUES(5,'cel.ro/electrocasnice');
```

```
INSERT INTO PROGRAM_ VALUES(7,'cel.ro/electrocasnice');
INSERT INTO PROGRAM_ VALUES(5,'flanco/electrocasnice');
INSERT INTO PROGRAM_ VALUES(6,'flanco/electrocasnice');
INSERT INTO PROGRAM_ VALUES(10,'pcgarage/calculatoare');
INSERT INTO PROGRAM_ VALUES(7,'pcgarage/calculatoare');

INSERT INTO PROMOVARE VALUES(1,1,'www.reclama1.ro');
INSERT INTO PROMOVARE VALUES(1,1,'www.reclama2.ro');
INSERT INTO PROMOVARE VALUES(2,1,'www.reclama2.ro');
INSERT INTO PROMOVARE VALUES(2,2,'www.yt/rec5.com');
INSERT INTO PROMOVARE VALUES(2,3,'www.yt/rec6.com');
INSERT INTO PROMOVARE VALUES(3,3,'www.reclama3.ro');
INSERT INTO PROMOVARE VALUES(3,4,'www.yt/rec7.com');
INSERT INTO PROMOVARE VALUES(4,5,'www.reclama3.ro');
INSERT INTO PROMOVARE VALUES(5,6,'www.reclama4.ro');
INSERT INTO PROMOVARE VALUES(5,6,'www.yt/rec7.com');

INSERT INTO ECHIPA_MARKETING VALUES(1,120);
INSERT INTO ECHIPA_MARKETING VALUES(2,20);
INSERT INTO ECHIPA_MARKETING VALUES(3,10);
INSERT INTO ECHIPA_MARKETING VALUES(4,5);
INSERT INTO ECHIPA_MARKETING VALUES(5,1);
INSERT INTO ECHIPA_MARKETING VALUES(6,2000);

INSERT INTO RECLAMA VALUES('www.reclama1.ro','imagine',540,400,null,null);
INSERT INTO RECLAMA VALUES('www.reclama2.ro','imagine',840,300,null,null);
INSERT INTO RECLAMA VALUES('www.reclama3.ro','imagine',330,330,null,null);
INSERT INTO RECLAMA VALUES('www.reclama4.ro','imagine',540,540,null,null);
INSERT INTO RECLAMA VALUES('www.yt/rec5.com','video',1920,1260,30,'1080p');
INSERT INTO RECLAMA VALUES('www.yt/rec6.com','video',1400,800,22,'720p');
INSERT INTO RECLAMA VALUES('www.yt/rec7.com','video',720,400,5,'360p');

commit;
```

proiect~1.sql

Worksheet | Query Builder

```sql
INSERT INTO ECHIPA_MARKETING VALUES(1,120);
INSERT INTO ECHIPA_MARKETING VALUES(2,20);
INSERT INTO ECHIPA_MARKETING VALUES(3,10);
INSERT INTO ECHIPA_MARKETING VALUES(4,5);
INSERT INTO ECHIPA_MARKETING VALUES(5,1);
INSERT INTO ECHIPA_MARKETING VALUES(6,2000);

INSERT INTO RECLAMA VALUES('www.reclama1.ro','imagine',540,400,null,null);
INSERT INTO RECLAMA VALUES('www.reclama2.ro','imagine',840,300,null,null);
INSERT INTO RECLAMA VALUES('www.reclama3.ro','imagine',330,330,null,null);
INSERT INTO RECLAMA VALUES('www.reclama4.ro','imagine',540,540,null,null);
INSERT INTO RECLAMA VALUES('www.yt/rec5.com','video',1920,1260,30,'1080p');
INSERT INTO RECLAMA VALUES('www.yt/rec6.com','video',1400,800,22,'720p');
INSERT INTO RECLAMA VALUES('www.yt/rec7.com','video',720,400,5,'360p');

INSERT INTO PROMOVARE VALUES(1,1,'www.reclama1.ro');
INSERT INTO PROMOVARE VALUES(1,1,'www.reclama2.ro');
INSERT INTO PROMOVARE VALUES(2,1,'www.reclama2.ro');
INSERT INTO PROMOVARE VALUES(2,2,'www.yt/rec5.com');
INSERT INTO PROMOVARE VALUES(2,3,'www.yt/rec6.com');
INSERT INTO PROMOVARE VALUES(3,3,'www.reclama3.ro');
INSERT INTO PROMOVARE VALUES(3,4,'www.yt/rec7.com');
INSERT INTO PROMOVARE VALUES(4,5,'www.reclama3.ro');
INSERT INTO PROMOVARE VALUES(5,6,'www.reclama4.ro');
INSERT INTO PROMOVARE VALUES(5,6,'www.yt/rec7.com');

commit;
```

Script Output | Query Result

Task completed in 0.305 seconds

```
1 row inserted.


1 row inserted.


1 row inserted.


Commit complete.
```

proiect4.sql | ACHIZITIE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers

Sort.. | Filter:

	ID_CLIENT	COD_PRODUS	SUMA
1	1	1	(null)
2	1	2	(null)
3	1	8	(null)
4	1	13	(null)
5	2	3	(null)
6	2	4	(null)
7	3	5	(null)
8	4	3	(null)
9	4	4	(null)
10	4	6	(null)
11	4	7	(null)
12	4	11	(null)
13	5	4	(null)
14	5	12	(null)
15	6	16	(null)
16	7	14	(null)
17	7	16	(null)
18	8	10	(null)
19	8	15	(null)
20	8	16	(null)

proiect5.sql | ANGAJAT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	NUME	COD_ANG...	ID_MAGAZIN	TIP_ANGAJAT	DATA_ANGAJARE	ID_SEDIU	SALARIU
1	Rayyan Holmes	1	1	manager	23-AUG-01	(null)	7500
2	Ayub Shannon	2	1	curier	21-DEC-15	(null)	3500
3	Bobbie Rollins	3	1	curier	04-JUL-13	(null)	3500
4	Sabiha Li	4	1	curier	23-DEC-07	(null)	3500
5	Devon Higgins	5	1	paznic	07-MAR-14	7	1500
6	Josef Cobb	6	1	call-center	31-OCT-02	(null)	2500
7	Winifred Cornish	7	2	manager	17-JUN-14	(null)	7500
8	Elen Avalos	8	2	curier	28-JUN-08	(null)	3500
9	Usamah Hastings	9	2	paznic	23-OCT-01	8	1500
10	Nahla Duke	10	2	paznic	27-APR-05	8	1500
11	Jade Douglas	11	2	call-center	29-APR-20	(null)	2500
12	Kameron Haigh	12	3	manager	11-JAN-05	(null)	7500
13	Ashley Povey	13	3	curier	19-NOV-15	(null)	3500
14	Ellie-Mai Erickson	14	3	curier	08-JAN-05	(null)	3500
15	Javan Ashton	15	3	paznic	09-JUN-09	9	1500
16	Alistair Aldred	16	3	call-center	10-JAN-05	(null)	2500
17	Rafi Sellers	17	4	manager	11-AUG-02	(null)	7500
18	Rikki Bouvet	18	4	curier	19-SEP-17	(null)	3500
19	Precious Tait	19	4	paznic	28-MAY-18	10	1500
20	Elize Devlin	20	4	call-center	20-APR-16	(null)	2500
21	Eva Kelley	21	4	call-center	09-AUG-16	(null)	2500
22	Steffan King	22	5	manager	25-JUL-03	(null)	7500
23	Courtney Wright	23	5	curier	30-MAR-04	(null)	3500
24	Jo Leal	24	5	curier	10-APR-12	(null)	3500
25	Osman Grainger	25	5	paznic	23-AUG-18	11	1500
26	Vera Hibbert	26	5	call-center	19-SEP-18	(null)	2500
27	Justine Franks	27	5	call-center	25-JAN-06	(null)	2500

proiect4.sql | CLIENT_

Columns | Data | Model | Constraints | Grants | Statistics | Triggers

Sort.. | Filter:

	ID_CLIENT	COD_ANGAJAT	PLATIT
1	1	6	100
2	2	6	1
3	3	11	1550
4	4	16	500
5	5	20	4490
6	6	20	5432
7	7	20	1234
8	8	27	556

proiect4.sql | DISTRIBUITOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback

Sort.. | Filter:

	ID_DISTRIBUIT...	NUME	DATA_INFIINTARE
1	1	Nvidia	10-APR-06
2	2	AMD	20-SEP-00
3	3	Pepsico	04-APR-04
4	4	Samsung	15-DEC-10
5	5	MancareSRL	22-FEB-97
6	6	Bosch	30-MAY-90
7	7	Adidas	22-JAN-02

proiect4.sql | ECHIPA_MARKETING

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flash

Sort.. | Filter:

	ID_FIRMA_MARKETING	NR_ANGAJATI
1	1	120
2	2	20
3	3	10
4	4	5
5	5	1
6	6	2000

proiect4.sql | FABRICA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details

Sort.. | Filter:

	COD_FAB...	NUME	ADRESA
1	1	Fabrica de calculatoare	Grivitei 355
2	2	Fabrica de mancare	Berceni 30
3	3	Fabrica de becuri	Brancoveanu 12
4	4	Fabrica de cabluri	Pantelimon 123
5	5	Fabrica de laptopuri	Timisoarei 10
6	6	Fabrica de frigidere	Giurgiului 300
7	7	Fabrica de mancare	Jiului 19

proiect4.sql | FIRMA_IT

Columns | Data | Model | Constraints | Grants | Statistics

Sort.. | Filter:

	ID_FIRMA	NR_ANGAJATI
1	1	100
2	2	10
3	3	1
4	4	2000
5	5	335
6	6	14
7	7	57

proiect4.sql | MAGAZIN

Columns | Data | Model | Constraints | Gra

Sort..

	ID_MAGAZIN	NUME
1	1	eMag
2	2	Cel.ro
3	3	Altex
4	4	Flanco
5	5	PcGarage

proiect4.sql | OFERTA

Columns | Data | Model | Constraints | Grants | Statistics | T

Sort.. | Filter:

	COD_PRODUS	ID_DISTRIBUITOR
1	1	1
2	1	2
3	2	2
4	3	4
5	3	6
6	4	4
7	4	6
8	5	3
9	6	4
10	6	6
11	7	4
12	8	1
13	8	4
14	9	4
15	10	2
16	11	4
17	11	6
18	12	4
19	12	6
20	13	1
21	14	2
22	15	2
23	16	1

proiect4.sql | MUNCITOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partition

Sort.. | Filter:

	CNP	DATA_NASTERE	AN_INCEPUT_MUNCA	NUME	PRENUME
1	2881224098035	24-DEC-88	2011	Pompiliu	Olga
2	1940223365590	23-FEB-94	2015	Horatiu	Andrei
3	1871205464650	05-DEC-87	2013	Teodor	Florin
4	1890313140275	13-MAR-89	2017	David	Victor
5	1860909035887	09-SEP-86	2006	Dacian	Felix

proiect4.sql | PARTENERIAT

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback

Sort.. | Filter:

	ID_PRODUCATOR	COD_FABRICA	DATA_INCEPUT
1	1	1	14-JUN-10
2	1	2	08-JUL-11
3	1	3	11-JUL-12
4	2	4	26-AUG-14
5	2	5	01-JAN-15
6	3	6	01-JAN-16
7	4	4	08-APR-16
8	4	6	27-FEB-17
9	5	3	22-MAY-17
10	5	4	16-MAY-18
11	5	5	01-MAR-19
12	6	1	11-OCT-19

proiect4.sql | PAGINA_WEB

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dep

Sort.. | Filter:

	LINK_PAGINA	AN_CREARE	ID_SERVER
1	emag/electrocasnice	2009	1
2	emag/calculatoare	2014	1
3	altex/electrocasnice	2005	2
4	altex/calculatoare	2006	2
5	cel.ro/electrocasnice	2015	3
6	flanco/electrocasnice	2009	4
7	pcgarage/calculatoare	2009	5
8	emag/mancare	2005	6

proiect4.sql | PRESTEAZA

Columns | Data | Model | Constraints | Grants | Statistics

Sort.. | Filter:

	COD_PRODUS	CNP
1	12	2881224098035
2	13	2881224098035
3	14	2881224098035
4	15	2881224098035
5	16	2881224098035
6	15	1940223365590
7	16	1940223365590
8	13	1871205464650
9	15	1871205464650
10	16	1871205464650
11	15	1890313140275
12	13	1860909035887
13	14	1860909035887
14	15	1860909035887
15	16	1860909035887

proiect4.sql | PRODUCATOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback

Sort.. | Filter:

	ID_PRODUCATOR	NUME	DATA_INFIINTARE
1	1	Producator1	26-JUL-02
2	2	Producator2	03-MAY-05
3	3	Producator3	03-NOV-05
4	4	Producator4	04-AUG-16
5	5	Producator5	06-APR-18
6	6	Producator6	08-NOV-19

proiect4.sql | PRODUS

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL

Sort.. | Filter:

	COD_PRODUS	TIP_PRODUS	PRET	COD_CURIER	ID_PRODUCATOR	NUME
1	1	bun	6000	2	1	placa video
2	2	bun	1000	2	2	procesor
3	3	bun	1500	3	3	masina de spalat
4	4	bun	1800	4	4	aer conditionat
5	5	bun	5	8	5	o sticla de pepsi
6	6	bun	300	13	6	aspirator
7	7	bun	200	14	5	masina de ras
8	8	bun	840	14	4	licenta windows 10
9	9	bun	50	14	3	joc pe cd GTA
10	10	bun	30	18	2	cablu internet
11	11	bun	10	23	1	bec economic
12	12	serviciu	100	(null)	(null)	curatare aparat
13	13	serviciu	150	(null)	(null)	asamblare calculator
14	14	serviciu	200	(null)	(null)	reparatie calculator
15	15	serviciu	50	(null)	(null)	mentenanta
16	16	serviciu	80	(null)	(null)	soft-update calculator

proiect4.sql | PROGRAM_

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dep

Sort.. | Filter:

	COD_PROGRAMATOR	LINK_PAGINA
1	1	emag/electrocasnice
2	3	emag/electrocasnice
3	4	emag/calculatoare
4	1	emag/mancare
5	3	altex/electrocasnice
6	9	altex/calculatoare
7	4	cel.ro/electrocasnice
8	5	cel.ro/electrocasnice
9	7	cel.ro/electrocasnice
10	5	flanco/electrocasnice
11	6	flanco/electrocasnice
12	10	pcgarage/calculatoare
13	7	pcgarage/calculatoare

proiect4.sql | PROGRAMATOR

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details

Sort.. | Filter:

	COD_PROGRAMATOR	LIMBAJE_CUNOSCUTE	ID_FIRMA
1	1	cpp	1
2	2	sql	1
3	3	javascript cpp	1
4	4	python javascript cpp sql	2
5	5	html css node	2
6	6	sql node cpp javascript	3
7	7	python	4
8	8	sql	5
9	9	html css	6
10	10	html css javascript	7

proiect4.sql | PROMOVARE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Depend

Sort.. | Filter:

	ID_MAGAZIN	ID_FIRMA_MARKETING	URL_RECLAMA
1	1	1	www.reclama1.ro
2	1	1	www.reclama2.ro
3	2	1	www.reclama2.ro
4	2	2	www.yt/rec5.com
5	2	3	www.yt/rec6.com
6	3	3	www.reclama3.ro
7	3	4	www.yt/rec7.com
8	4	5	www.reclama3.ro
9	5	6	www.reclama4.ro
10	5	6	www.yt/rec7.com

proiect4.sql | RECLAMA

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Pa

Sort.. | Filter:

	URL_RECLAMA	TIP_RECLAMA	HEIGHT	WIDTH	LEN	QUALITY
1	www.reclama1.ro	imagine	540	400	(null)	(null)
2	www.reclama2.ro	imagine	840	300	(null)	(null)
3	www.reclama3.ro	imagine	330	330	(null)	(null)
4	www.reclama4.ro	imagine	540	540	(null)	(null)
5	www.yt/rec5.com	video	1920	1260	30	1080p
6	www.yt/rec6.com	video	1400	800	22	720p
7	www.yt/rec7.com	video	720	400	5	360p

proiect4.sql | SERVER

Columns | Data | Model | Constraints | Grants | Statistics | Trig

Sort.. | Filter:

	ID_SERVER	ID_FIRMA	MAXSIZE
1	1	1	10000
2	2	2	26000
3	3	1	1000
4	4	2	5555
5	5	3	4885
6	6	3	123456
7	7	4	5235
8	8	4	100
9	9	4	9500
10	10	4	40

proiect4.sql | SEDIU

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback

Sort.. | Filter:

	ID_SEDIU	ADRESA	ID_MAGAZIN
1	7	Bvd Iuliu Maniu 200	1
2	8	Timpuri Noi 24	2
3	9	Tineretului 210	3
4	10	Lipscani 3	4
5	11	Nordului 55	5

proiect4.sql | VANZARE

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies

Sort.. | Filter:

	ID_MAGAZIN	ID_DISTRIBUITOR	LINK_PAGINA
1	1	1	emag/calculatoare
2	1	2	emag/calculatoare
3	1	3	emag/mancare
4	1	4	emag/electrocasnice
5	1	4	emag/mancare
6	1	5	emag/mancare
7	1	7	emag/electrocasnice
8	2	4	cel.ro/electrocasnice
9	2	6	cel.ro/electrocasnice
10	3	1	altex/calculatoare
11	3	2	altex/calculatoare
12	3	4	altex/electrocasnice
13	3	6	altex/electrocasnice
14	4	4	flanco/electrocasnice
15	4	6	flanco/electrocasnice
16	5	1	pcgarage/calculatoare
17	5	2	pcgarage/calculatoare

6.

Problema:

Creati o colectie ce contine fiecare limbaj de programare cunoscut de cel putin unul dintre programatori. Pentru fiecare limbaj enumerati programatorii (codul lor) ce cunosc limbajul , alaturi de codul firmei la care sunt angajati.

Codul:

```
CREATE OR REPLACE PROCEDURE ex6
IS
    TYPE tablou_numere IS TABLE OF NUMBER;
    TYPE tablou_indexat_string IS TABLE OF tablou_numere INDEX BY STRING(3000);
    TYPE tablou_limbaje IS TABLE OF STRING(3000);
    programatori_dupa_limbaj tablou_indexat_string;
    limbaje tablou_limbaje := tablou_limbaje();
    aux_limbaje tablou_limbaje := tablou_limbaje();
    aux_lim VARCHAR(50);
    nr NUMBER := 1;
    firma NUMBER;
BEGIN

    FOR rand IN (SELECT * FROM PROGRAMATOR) LOOP
         SELECT trim(regexp_substr(rand.limbaje_cunoscute, '[^ ]+', 1, LEVEL))
         BULK COLLECT INTO aux_limbaje
         FROM DUAL
         CONNECT BY regexp_substr(rand.limbaje_cunoscute , '[^ ]+', 1, LEVEL) IS NOT
NULL;



         FOR i in aux_limbaje.FIRST..aux_limbaje.LAST LOOP
            IF aux_limbaje(i) member OF limbaje THEN
                NULL;
```

```
            ELSE
                limbaje.EXTEND;
                limbaje(limbaje.LAST) := aux_limbaje(i);
            END IF;
         END LOOP;

         aux_limbaje.DELETE;
    END LOOP;

    FOR i in limbaje.first..limbaje.last LOOP
        SELECT cod_programator
        BULK COLLECT INTO programatori_dupa_limbaj(limbaje(i))
        FROM programator
        WHERE INSTR(limbaje_cunoscute, limbaje(i)) != 0;
    END LOOP;



    aux_lim := programatori_dupa_limbaj.FIRST;
    WHILE aux_lim IS NOT NULL LOOP
        DBMS_OUTPUT.PUT_LINE(aux_lim);
        DBMS_OUTPUT.PUT_LINE('~~~~~~~~~~');
        FOR j in
programatori_dupa_limbaj(aux_lim).FIRST..programatori_dupa_limbaj(aux_lim).LAST LOOP
            SELECT id_firma INTO firma FROM programator WHERE cod_programator =
programatori_dupa_limbaj(aux_lim)(j);
            DBMS_OUTPUT.PUT_LINE(nr || ') Programatorul cu codul ' ||
programatori_dupa_limbaj(aux_lim)(j) || ' de la firma cu codul ' || firma);
            nr := nr+1;
        END LOOP;
        nr := 1;
        aux_lim := programatori_dupa_limbaj.NEXT(aux_lim);
        DBMS_OUTPUT.PUT_LINE('');
    END LOOP;


END ex6;
```

```
/

BEGIN
    ex6();
END;
```

proiect~1.sql

SQL Worksheet | History

Worksheet | Query Builder

```
CREATE OR REPLACE PROCEDURE ex6
IS
    TYPE tablou_numere IS TABLE OF NUMBER;
    TYPE tablou_indexat_string IS TABLE OF tablou_numere INDEX BY STRING(3000);
    TYPE tablou_limbaje IS TABLE OF STRING(3000);
    programatori_dupa_limbaj tablou_indexat_string;
    limbaje tablou_limbaje := tablou_limbaje();
    aux_limbaje tablou_limbaje := tablou_limbaje();
    aux_lim VARCHAR(50);
    nr NUMBER := 1;
    firma NUMBER;
BEGIN

    FOR rand IN (SELECT * FROM PROGRAMATOR) LOOP
         SELECT trim(regexp_substr(rand.limbaje_cunoscute, '[^ ]+', 1, LEVEL))
         BULK COLLECT INTO aux_limbaje
         FROM DUAL
         CONNECT BY regexp_substr(rand.limbaje_cunoscute , '[^ ]+', 1, LEVEL) IS NOT NULL;


         FOR i in aux_limbaje.FIRST..aux_limbaje.LAST LOOP
            IF aux_limbaje(i) member OF limbaje THEN
                NULL;
            ELSE
                limbaje.EXTEND;
                limbaje(limbaje.LAST) := aux_limbaje(i);
            END IF;
         END LOOP;

         aux_limbaje.DELETE;
    END LOOP;

    FOR i in limbaje.first..limbaje.last LOOP
        SELECT cod_programator
        BULK COLLECT INTO programatori_dupa_limbaj(limbaje(i))
        FROM programator
        WHERE INSTR(limbaje_cunoscute, limbaje(i)) != 0;
    END LOOP;
```

Script Output | Query Result

Task completed in 0.05 seconds

```
PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

proiect

```
cpp
~~~~~~~~~~~
1) Programatorul cu codul 1 de la firma cu codul 1
2) Programatorul cu codul 3 de la firma cu codul 1
3) Programatorul cu codul 4 de la firma cu codul 2
4) Programatorul cu codul 6 de la firma cu codul 3

css
~~~~~~~~~~~
1) Programatorul cu codul 5 de la firma cu codul 2
2) Programatorul cu codul 9 de la firma cu codul 6
3) Programatorul cu codul 10 de la firma cu codul 7

html
~~~~~~~~~~~
1) Programatorul cu codul 5 de la firma cu codul 2
2) Programatorul cu codul 9 de la firma cu codul 6
3) Programatorul cu codul 10 de la firma cu codul 7

javascript
~~~~~~~~~~~
1) Programatorul cu codul 3 de la firma cu codul 1
2) Programatorul cu codul 4 de la firma cu codul 2
3) Programatorul cu codul 6 de la firma cu codul 3
4) Programatorul cu codul 10 de la firma cu codul 7

node
~~~~~~~~~~~
1) Programatorul cu codul 5 de la firma cu codul 2
2) Programatorul cu codul 6 de la firma cu codul 3

python
~~~~~~~~~~~
1) Programatorul cu codul 4 de la firma cu codul 2
2) Programatorul cu codul 7 de la firma cu codul 4

sql
~~~~~~~~~~~
1) Programatorul cu codul 2 de la firma cu codul 1
2) Programatorul cu codul 4 de la firma cu codul 2
3) Programatorul cu codul 6 de la firma cu codul 3
4) Programatorul cu codul 8 de la firma cu codul 5
```

7.

Problema:

Selectati toate magazinele care vand cel putin un produs de 2000 de lei si care isi fac reclama prin cel putin o imagine. Adaugati tabelului Magazin coloana 'taxa', in care veti introduce 10% din pretul celui mai scump produs vandut de magazin daca se afla printre cele selectate mai sus, sau 20% din cel mai ieftin produs in caz contrar. Filtrati-le dupa cat de mare este taxa: taxa este mare(> 199), medie (30 - 199) sau mica (< 30).

Codul:

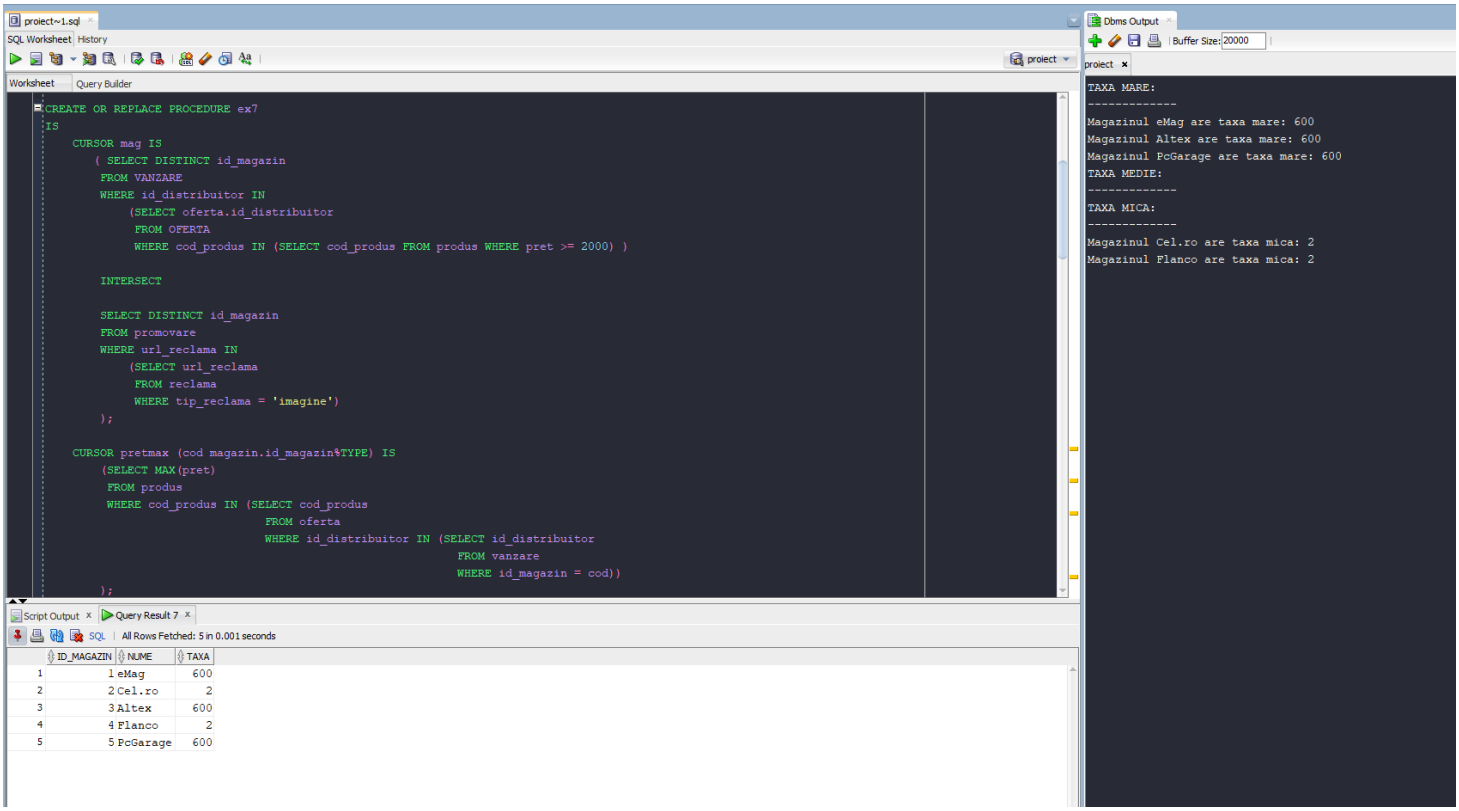
```
commit;

/

ALTER TABLE magazin
ADD taxa NUMBER;

    /

CREATE OR REPLACE PROCEDURE ex7
IS
    CURSOR mag IS
       ( SELECT DISTINCT id_magazin
        FROM VANZARE
        WHERE id_distribuitor IN
            (SELECT oferta.id_distribuitor
             FROM OFERTA
             WHERE cod_produs IN (SELECT cod_produs FROM produs WHERE pret >= 2000) )

        INTERSECT

        SELECT DISTINCT id_magazin
        FROM promovare
        WHERE url_reclama IN
```

```
        (SELECT url_reclama
         FROM reclama
         WHERE tip_reclama = 'imagine')
    );

    CURSOR pretmax (cod magazin.id_magazin%TYPE) IS
    (SELECT MAX(pret)
     FROM produs
     WHERE cod_produs IN (SELECT cod_produs
                          FROM oferta
                          WHERE id_distribuitor IN (SELECT id_distribuitor
                                                    FROM vanzare
                                                    WHERE id_magazin = cod))
    );
    CURSOR pretmin (cod magazin.id_magazin%TYPE) IS
    (SELECT MIN(pret)
     FROM produs
     WHERE cod_produs IN (SELECT cod_produs
                          FROM oferta
                          WHERE id_distribuitor IN (SELECT id_distribuitor
                                                    FROM vanzare
                                                    WHERE id_magazin = cod))
    );
    suma produs.pret%TYPE;
    TYPE afisare IS REF CURSOR RETURN magazin%ROWTYPE;
    afis afisare;
    aux magazin%ROWTYPE;
    nume magazin.nume%TYPE;
    taxa magazin.taxa%TYPE;
    cd magazin.id_magazin%TYPE;
BEGIN

    FOR m IN mag LOOP
```

```
        OPEN pretmax(m.id_magazin);
        FETCH pretmax INTO suma;
        UPDATE magazin
        SET taxa = suma*0.1
        WHERE id_magazin = m.id_magazin;
        CLOSE pretmax;
    END LOOP;

    FOR m IN  ( SELECT id_magazin FROM Magazin
               WHERE id_magazin NOT IN (SELECT DISTINCT id_magazin
                FROM VANZARE
                WHERE id_distribuitor IN
                    (SELECT oferta.id_distribuitor
                     FROM OFERTA
                     WHERE cod_produs IN (SELECT cod_produs FROM produs WHERE pret >=
2000) )

                INTERSECT

                SELECT DISTINCT id_magazin
                FROM promovare
                WHERE url_reclama IN
                    (SELECT url_reclama
                     FROM reclama
                     WHERE tip_reclama = 'imagine') ) ) LOOP

        OPEN pretmin(m.id_magazin);
        FETCH pretmin INTO suma;
        UPDATE magazin
        SET taxa = suma*0.2
        WHERE id_magazin = m.id_magazin;
        CLOSE pretmin;
    END LOOP;
```

```
    DBMS_OUTPUT.PUT_LINE('TAXA MARE:');
    DBMS_OUTPUT.PUT_LINE('-------------');
    OPEN afis FOR SELECT * FROM magazin WHERE taxa>199;
    LOOP
        FETCH afis into aux;
        EXIT WHEN afis%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa mare: ' || aux.taxa);
    END LOOP;
    CLOSE afis;
    DBMS_OUTPUT.PUT_LINE('TAXA MEDIE:');
    DBMS_OUTPUT.PUT_LINE('-------------');
    OPEN afis FOR SELECT * FROM magazin WHERE taxa BETWEEN 30 and 199;
    LOOP
        FETCH afis into aux;
        EXIT WHEN afis%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa medie: ' ||
aux.taxa);
    END LOOP;
    CLOSE afis;
    DBMS_OUTPUT.PUT_LINE('TAXA MICA:');
    DBMS_OUTPUT.PUT_LINE('-------------');
    OPEN afis FOR SELECT * FROM magazin WHERE taxa< 30;
    LOOP
        FETCH afis into aux;
        EXIT WHEN afis%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa mica: ' || aux.taxa);
    END LOOP;
    CLOSE afis;

END ex7;

/
```

```
BEGIN
    ex7();
END;

/

SELECT *
FROM Magazin;

/

rollback;
```



8.

Problema:

Scrieti o functie stocata ce primeste numele unei fabrici (verificati daca acesta exista, daca este unic, in caz contrar ridicati o eroare) ca parametru si intoarce o scurta statistica : cat la suta dintre produsele BUNURI sunt fabricate in aceasta fabrica si cat la suta dintre clienti au cumparat cel putin un produs dintre acestea.

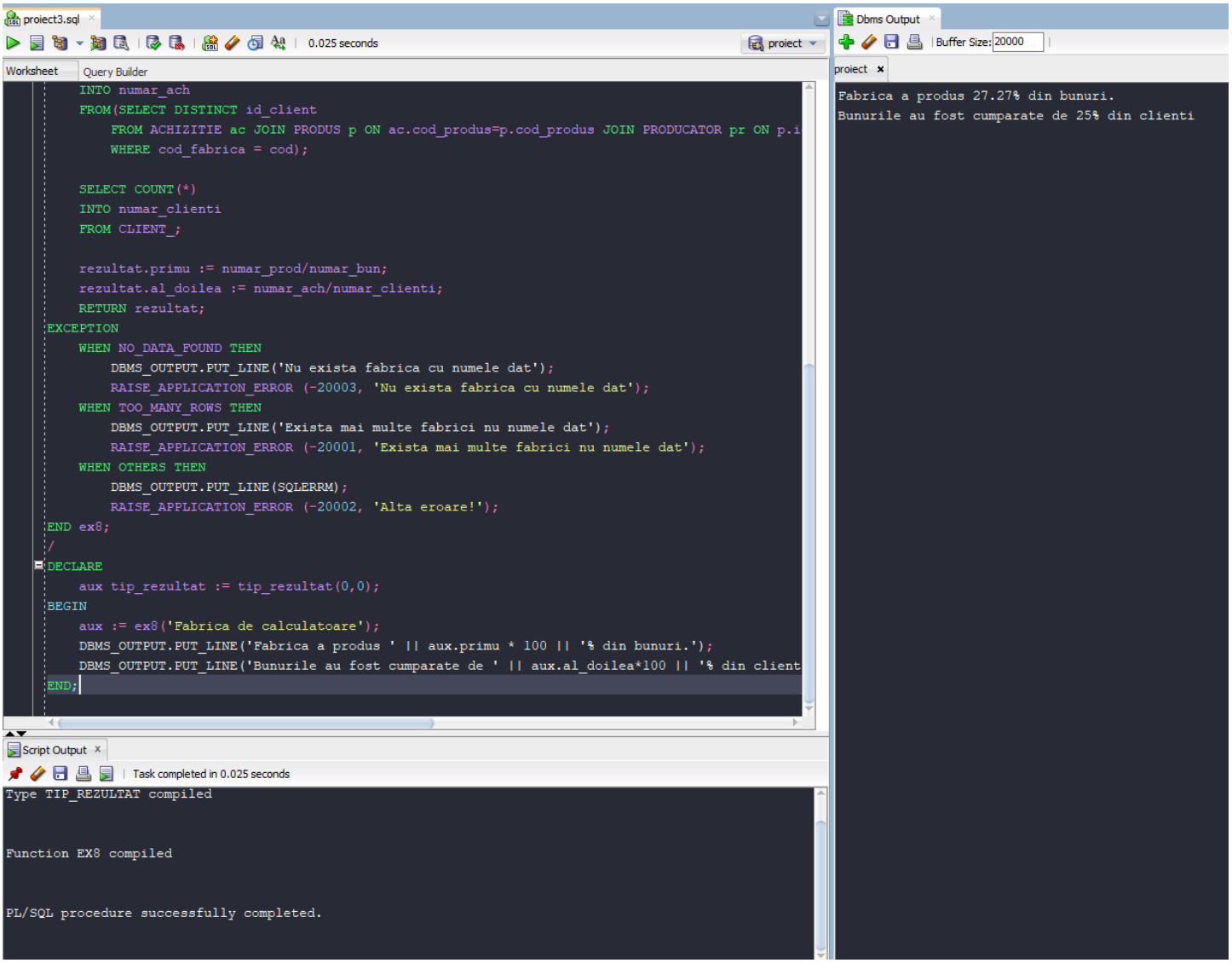
Codul:

```
--valori pentru cazul too many rows
INSERT INTO FABRICA VALUES(7, 'Fabrica de mancare', 'Jiului 19');
--il facem obiect in loc de record pt a putea face functia stocata, nu declarata in bloc
CREATE OR REPLACE TYPE tip_rezultat IS OBJECT(primu numeric(5,4), al_doilea
numeric(5,4));
/
CREATE OR REPLACE FUNCTION ex8(numele fabrica.nume%TYPE DEFAULT 'Fabrica de becuri')
RETURN tip_rezultat
IS
    rezultat tip_rezultat := tip_rezultat(0,0);
    numar_prod NUMBER;
    numar_bun NUMBER;
    numar_ach NUMBER;
    numar_clienti NUMBER;
    cod fabrica.cod_fabrica%TYPE;
BEGIN
    SELECT cod_fabrica INTO cod FROM fabrica WHERE nume = numele;

    SELECT COUNT(*)
    INTO numar_prod
    FROM PRODUS p JOIN PRODUCATOR pr ON p.id_producator = pr.id_producator JOIN
PARTENERIAT pa ON pa.id_producator = pr.id_producator
    WHERE cod_fabrica = cod;

    SELECT COUNT(*)
```

```
    INTO numar_bun
    FROM PRODUS
    WHERE tip_produs = 'bun';

    SELECT COUNT(*)
    INTO numar_ach
    FROM(SELECT DISTINCT id_client
        FROM ACHIZITIE ac JOIN PRODUS p ON ac.cod_produs=p.cod_produs JOIN PRODUCATOR pr
ON p.id_producator = pr.id_producator JOIN PARTENERIAT pa ON pa.id_producator =
pr.id_producator
        WHERE cod_fabrica = cod);

    SELECT COUNT(*)
    INTO numar_clienti
    FROM CLIENT_;

    rezultat.primu := numar_prod/numar_bun;
    rezultat.al_doilea := numar_ach/numar_clienti;
    RETURN rezultat;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista fabrica cu numele dat');
        RAISE_APPLICATION_ERROR (-20003, 'Nu exista fabrica cu numele dat');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Exista mai multe fabrici nu numele dat');
        RAISE_APPLICATION_ERROR (-20001, 'Exista mai multe fabrici nu numele dat');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
        RAISE_APPLICATION_ERROR (-20002, 'Alta eroare!');
END ex8;
/
DECLARE
    aux tip_rezultat := tip_rezultat(0,0);
BEGIN
    aux := ex8('Fabrica de scaune');
```

```
    DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');

    DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din
clienti');

END;
```

Cazul fara erori:



Cazul cu NO DATA FOUND:

proiect3.sql

```
        INTO numar_ach
        FROM(SELECT DISTINCT id_client
            FROM ACHIZITIE ac JOIN PRODUS p ON ac.cod_produs=p.cod_produs JOIN PRODUCATOR pr ON p.i
            WHERE cod_fabrica = cod);

        SELECT COUNT(*)
        INTO numar_clienti
        FROM CLIENT_;

        rezultat.primu := numar_prod/numar_bun;
        rezultat.al_doilea := numar_ach/numar_clienti;
        RETURN rezultat;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista fabrica cu numele dat');
            RAISE_APPLICATION_ERROR (-20003, 'Nu exista fabrica cu numele dat');
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multe fabrici nu numele dat');
            RAISE_APPLICATION_ERROR (-20001, 'Exista mai multe fabrici nu numele dat');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE(SQLERRM);
            RAISE_APPLICATION_ERROR (-20002, 'Alta eroare!');
    END ex8;
    /
    DECLARE
        aux tip_rezultat := tip_rezultat(0,0);
    BEGIN
        aux := ex8('Fabrica de scaune');
        DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');
        DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din client
    END;
```

Dbms Output

```
Nu exista fabrica cu numele dat
```

Script Output

```
    DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');
    DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din clienti');
END;
Error report -
ORA-20003: Nu exista fabrica cu numele dat
ORA-06512: at "LAUR.EX8", line 39
ORA-06512: at line 4
```

Cazul cu TOO MANY ROWS:

proiect3.sql

```
    INTO numar_ach
    FROM(SELECT DISTINCT id_client
        FROM ACHIZITIE ac JOIN PRODUS p ON ac.cod_produs=p.cod_produs JOIN PRODUCATOR pr ON p.i
        WHERE cod_fabrica = cod);

    SELECT COUNT(*)
    INTO numar_clienti
    FROM CLIENT_;

    rezultat.primu := numar_prod/numar_bun;
    rezultat.al_doilea := numar_ach/numar_clienti;
    RETURN rezultat;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista fabrica cu numele dat');
        RAISE_APPLICATION_ERROR (-20003, 'Nu exista fabrica cu numele dat');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Exista mai multe fabrici nu numele dat');
        RAISE_APPLICATION_ERROR (-20001, 'Exista mai multe fabrici nu numele dat');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
        RAISE_APPLICATION_ERROR (-20002, 'Alta eroare!');
END ex8;
/
DECLARE
    aux tip_rezultat := tip_rezultat(0,0);
BEGIN
    aux := ex8('Fabrica de mancare');
    DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');
    DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din client
END;
```

Script Output

```
    DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');
    DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din clienti');
END;
Error report -
ORA-20001: Exista mai multe fabrici nu numele dat
ORA-06512: at "LAUR.EX8", line 42
ORA-06512: at line 4
```

Dbms Output

```
Exista mai multe fabrici nu numele dat
```

9.

Problema:

Afisati folosind o procedura stocata toti curierii care livreaza produse distribuite de distribuitorul cu numele dat ca parametru, magazinul la care lucreaza(numele) si procentul din clienti care au interactionat cu acest curier.

Codul:

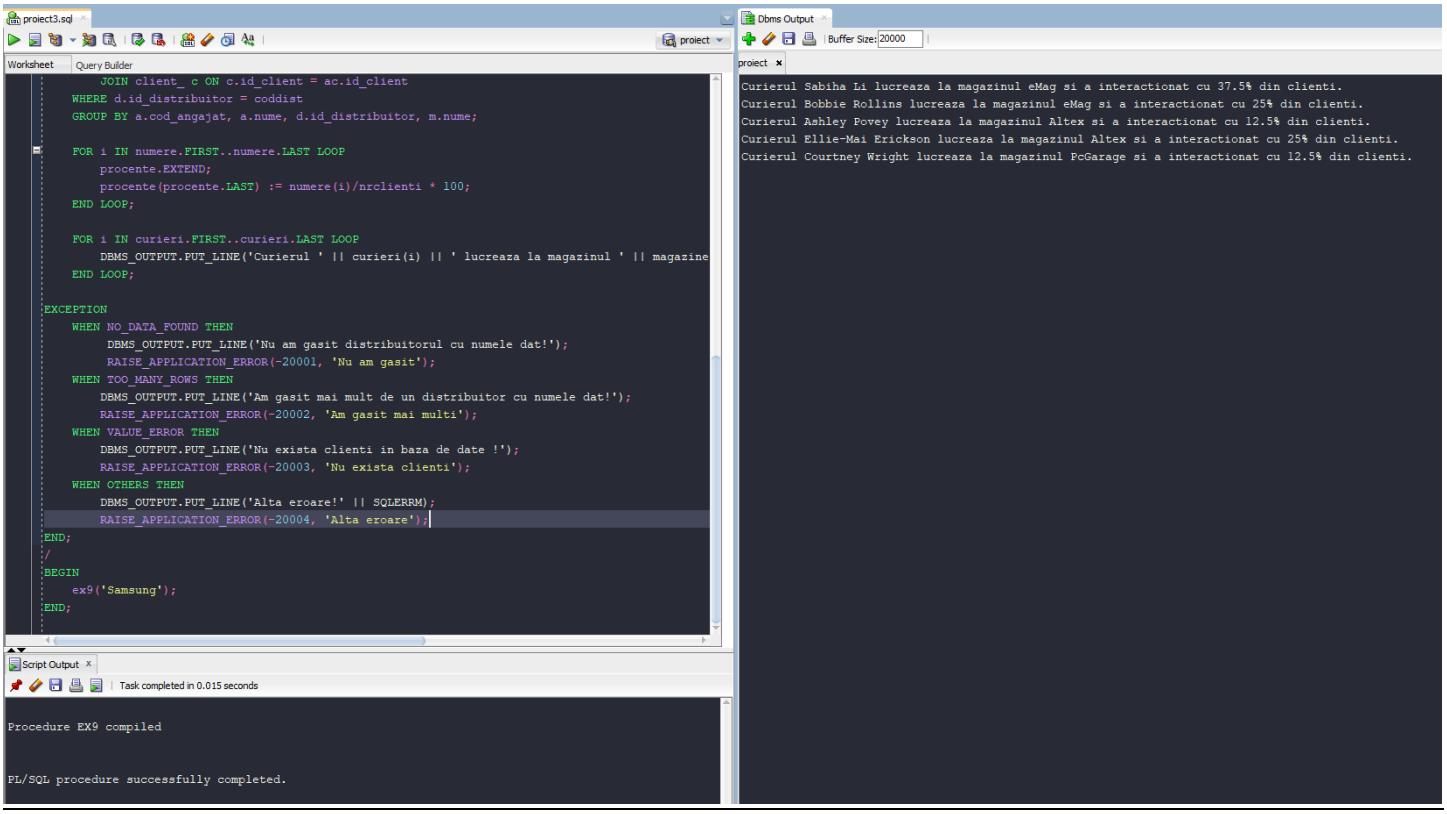
--valori pentru cazul too many rows:

```
INSERT INTO DISTRIBUITOR VALUES (100,'Bosch',TO_DATE('11-01-2001','DD-MM-YYYY'));


CREATE OR REPLACE PROCEDURE ex9 (numele distribuitor.nume%TYPE)
IS
    coddist distribuitor.id_distribuitor%TYPE;
    nrclienti NUMBER;
    TYPE tipnume IS TABLE OF STRING(3000);
    TYPE tipnumere IS TABLE OF NUMBER;
    TYPE tipprocente IS TABLE OF numeric(5,2);
    curieri tipnume := tipnume();
    magazine tipnume := tipnume();
    numere tipnumere := tipnumere();
    procente tipprocente := tipprocente();
BEGIN
    SELECT id_distribuitor INTO coddist
    FROM distribuitor
    WHERE nume = numele;

    SELECT COUNT(*) INTO nrclienti
    FROM CLIENT_;

    SELECT a.nume,m.nume,COUNT(c.id_client)
    BULK COLLECT INTO curieri, magazine, numere
```

```
    FROM angajat a JOIN produs p ON a.cod_angajat = p.cod_curier
        JOIN oferta o ON p.cod_produs = o.cod_produs
        JOIN distribuitor d ON o.id_distribuitor = d.id_distribuitor
        JOIN magazin m ON a.id_magazin = m.id_magazin
        JOIN achizitie ac ON p.cod_produs = ac.cod_produs
        JOIN client_ c ON c.id_client = ac.id_client
    WHERE d.id_distribuitor = coddist
    GROUP BY a.cod_angajat, a.nume, d.id_distribuitor, m.nume;

    FOR i IN numere.FIRST..numere.LAST LOOP
        procente.EXTEND;
        procente(procente.LAST) := numere(i)/nrclienti * 100;
    END LOOP;

    FOR i IN curieri.FIRST..curieri.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Curierul ' || curieri(i) || ' lucreaza la magazinul ' ||
magazine(i) || ' si a interactionat cu ' || procente(i) || '% din clienti.');
    END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Nu am gasit distribuitorul cu numele dat!');
         RAISE_APPLICATION_ERROR(-20001, 'Nu am gasit');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Am gasit mai mult de un distribuitor cu numele dat!');
        RAISE_APPLICATION_ERROR(-20002, 'Am gasit mai multi');
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti in baza de date !');
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista clienti');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!' || SQLERRM);
        RAISE_APPLICATION_ERROR(-20004, 'Alta eroare');
END ex9;
```

```
/
BEGIN
    ex9('Samsung');
END;
```

Cazul fara erori:



Cazul NO DATA FOUND:

proiect3.sql

0.034 seconds

Worksheet | Query Builder

```
        JOIN client_ c ON c.id_client = ac.id_client
    WHERE d.id_distribuitor = coddist
    GROUP BY a.cod_angajat, a.nume, d.id_distribuitor, m.nume;

    FOR i IN numere.FIRST..numere.LAST LOOP
        procente.EXTEND;
        procente(procente.LAST) := numere(i)/nrclienti * 100;
    END LOOP;

    FOR i IN curieri.FIRST..curieri.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Curierul ' || curieri(i) || ' lucreaza la magazinul ' || magazine
    END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Nu am gasit distribuitorul cu numele dat!');
         RAISE_APPLICATION_ERROR(-20001, 'Nu am gasit');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Am gasit mai mult de un distribuitor cu numele dat!');
        RAISE_APPLICATION_ERROR(-20002, 'Am gasit mai multi');
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti in baza de date !');
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista clienti');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!' || SQLERRM);
        RAISE_APPLICATION_ERROR(-20004, 'Alta eroare');
END;
/
BEGIN
    ex9('Cineva');
END;
```

Script Output

Task completed in 0.034 seconds

```
BEGIN
    ex9('Cineva');
END;
Error report -
ORA-20001: Nu am gasit
ORA-06512: at "LAUR.EX9", line 43
ORA-06512: at line 2
```

Dbms Output

Buffer Size: 20000

proiect

```
Nu am gasit distribuitorul cu numele dat!
```

Cazul TOO MANY ROWS:

proiect3.sql

0.018 seconds

proiect

Worksheet | Query Builder

```
        JOIN client_ c ON c.id_client = ac.id_client
    WHERE d.id_distribuitor = coddist
    GROUP BY a.cod_angajat, a.nume, d.id_distribuitor, m.nume;

    FOR i IN numere.FIRST..numere.LAST LOOP
        procente.EXTEND;
        procente(procente.LAST) := numere(i)/nrclienti * 100;
    END LOOP;

    FOR i IN curieri.FIRST..curieri.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Curierul ' || curieri(i) || ' lucreaza la magazinul ' || magazine
    END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Nu am gasit distribuitorul cu numele dat!');
         RAISE_APPLICATION_ERROR(-20001, 'Nu am gasit');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Am gasit mai mult de un distribuitor cu numele dat!');
        RAISE_APPLICATION_ERROR(-20002, 'Am gasit mai multi');
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti in baza de date !');
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista clienti');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!' || SQLERRM);
        RAISE_APPLICATION_ERROR(-20004, 'Alta eroare');
END;
/
BEGIN
    ex9('Bosch');
END;
```

Script Output

Task completed in 0.018 seconds

```
BEGIN
    ex9('Bosch');
END;
Error report -
ORA-20002: Am gasit mai multi
ORA-06512: at "LAUR.EX9", line 46
ORA-06512: at line 2
```

Dbms Output

Buffer Size: 20000

proiect

```
Am gasit mai mult de un distribuitor cu numele dat!
```

Cazul VALUE ERROR: **(in cazul acesta impartire la 0 pentru ca am dat delete tuturor clientilor)**

proiect3.sql

```
        JOIN achizitie ac ON p.cod_produs = ac.cod_produs
        JOIN client_ c ON c.id_client = ac.id_client
    WHERE d.id_distribuitor = coddist
    GROUP BY a.cod_angajat, a.nume, d.id_distribuitor, m.nume;

    FOR i IN numere.FIRST..numere.LAST LOOP
        procente.EXTEND;
        procente(procente.LAST) := numere(i)/nrclienti * 100;
    END LOOP;

    FOR i IN curieri.FIRST..curieri.LAST LOOP
        DBMS_OUTPUT.PUT_LINE('Curierul ' || curieri(i) || ' lucreaza la magazinul ' || magazine
    END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Nu am gasit distribuitorul cu numele dat!');
         RAISE_APPLICATION_ERROR(-20001, 'Nu am gasit');
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Am gasit mai mult de un distribuitor cu numele dat!');
        RAISE_APPLICATION_ERROR(-20002, 'Am gasit mai multi');
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista clienti in baza de date !');
        RAISE_APPLICATION_ERROR(-20003, 'Nu exista clienti');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Alta eroare!' || SQLERRM);
        RAISE_APPLICATION_ERROR(-20004, 'Alta eroare');
END;
/
BEGIN
    ex9('Nvidia');
END;
```

Dbms Output

```
Nu exista clienti in baza de date !
```

Script Output

```
BEGIN
    ex9('Nvidia');
END;
Error report -
ORA-20003: Nu exista clienti
ORA-06512: at "LAUR.EX9", line 49
ORA-06512: at line 2
```

10.

Problema:

Creati un nou tabel 'Accesari' in care se vor introduce automat printr-un trigger actiunea efectuata asupra tabelului PRODUS, userul care a apelat instructiunea, data la care s-a apelat, si inca o coloana in care introducem 'la timp' / 'intarziat' dupa criteriul 'se executa in intervalul orar 12:00 - 20:00?' pentru fiecare dintre comenzile insert, update si delete.

Codul:

```
CREATE TABLE ACCESARI (
    indice NUMBER,
    actiune VARCHAR2(50),
    userul VARCHAR2(50),
    data_apel DATE,
    validat VARCHAR2(50),
    CONSTRAINT pk primary key(indice)
    )
/
CREATE OR REPLACE PACKAGE parametru
AS
    validare VARCHAR2(50) := 'la timp';
    usr VARCHAR2(50);
    actiune VARCHAR2(50);
    contor NUMBER := 1;
END;
/
CREATE OR REPLACE TRIGGER trig_ex10
    AFTER INSERT OR DELETE OR UPDATE ON PRODUS
BEGIN
    SELECT USER
    INTO parametru.usr
    FROM DUAL;
```

```
    IF INSERTING THEN
        parametru.actiune := 'inserare';
    ELSIF DELETING THEN
        parametru.actiune := 'stergere';
    ELSE
        parametru.actiune := 'actualizare';
    END IF;



    IF (TO_CHAR(SYSDATE, 'HH24') NOT BETWEEN 12 AND 20) THEN
        parametru.validare := 'intarziat';
    ELSE
        parametru.validare := 'la timp';
    END IF;

    parametru.contor := parametru.contor + 1;

    INSERT INTO ACCESARI VALUES(parametru.contor, parametru.actiune, parametru.usr,
SYSDATE, parametru.validare);


END;
/
INSERT INTO PRODUS VALUES(17,'bun',100,null,null,'virusi');
/
INSERT INTO PRODUS VALUES(18,'bun',100,null,null,'virusi');
/
DELETE FROM PRODUS WHERE cod_produs = 17;
/
UPDATE PRODUS SET nume = 'virusi 2' WHERE cod_produs = 18;
```

proiect4.sql

0.04 seconds

proiect

Worksheet | Query Builder

```
    IF INSERTING THEN
        parametru.actiune := 'inserare';
    ELSIF DELETING THEN
        parametru.actiune := 'stergere';
    ELSE
        parametru.actiune := 'actualizare';
    END IF;


    IF (TO_CHAR(SYSDATE, 'HH24') NOT BETWEEN 12 AND 20) THEN
        parametru.validare := 'intarziat';
    ELSE
        parametru.validare := 'la timp';
    END IF;

    parametru.contor := parametru.contor + 1;

    INSERT INTO ACCESARI VALUES(parametru.contor, parametru.actiune, parametru.usr, SYSDATE, parametru.validare);

END;
/
INSERT INTO PRODUS VALUES(17,'bun',100,null,null,'virusi');
/
INSERT INTO PRODUS VALUES(18,'bun',100,null,null,'virusi');
/
DELETE FROM PRODUS WHERE cod_produs = 17;
/
UPDATE PRODUS SET nume = 'virusi 2' WHERE cod_produs = 18;
```

Script Output

Task completed in 0.04 seconds

```
1 row inserted.


1 row inserted.


1 row deleted.


1 row updated.
```

Dbms Output | ACCESARI

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependen

Sort... | Filter:

	INDICE	ACTIUNE	USERUL	DATA_APEL	VALIDAT
1	2	inserare	LAUR	30-DEC-21	intarziat
2	3	inserare	LAUR	30-DEC-21	intarziat
3	4	stergere	LAUR	30-DEC-21	intarziat
4	5	actualizare	LAUR	30-DEC-21	intarziat

11.

Problema:

Construiti un trigger ce verifica pentru fiecare insert, update (se da update doar dupa cod_angajat) sau delete asupra tabelei ANGAJAT daca instructiunea a fost valida, iar in cazul in care nu este valida sa anuleze comanda (sa repare instructiunea precedenta). O instructiune este valida dupa cum urmeaza

- un delete este valid daca data angajarii nu este data curenta (ziua si luna)
- un insert sau un update este valid daca respecta regula 'doar paznicii au id_sediu diferit de null'

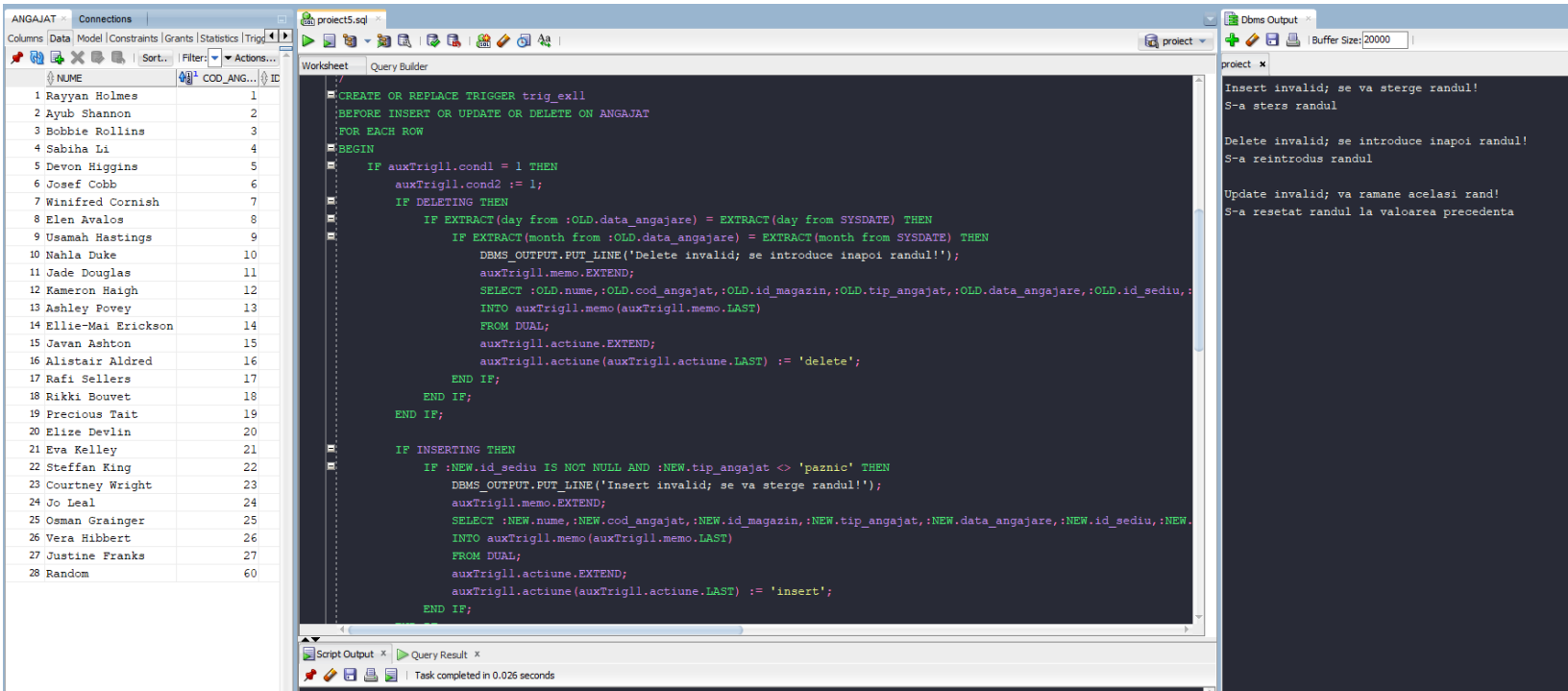
Codul:

```
CREATE OR REPLACE PACKAGE auxTrig11
AS
    TYPE tip IS TABLE OF angajat%ROWTYPE;
    TYPE act IS TABLE OF VARCHAR2(50);
    memo tip := tip();
    actiune act := act();
    cond1 NUMBER := 1;
    cond2 NUMBER := 1;
END;
/
BEGIN
    auxTrig11.cond1 := 1;
    auxTrig11.cond2 := 1;
END;
/
CREATE OR REPLACE TRIGGER trig_ex11
BEFORE INSERT OR UPDATE OR DELETE ON ANGAJAT
FOR EACH ROW
BEGIN
    IF auxTrig11.cond1 = 1 THEN
        auxTrig11.cond2 := 1;
        IF DELETING THEN
```

```
            IF EXTRACT(day from :OLD.data_angajare) = EXTRACT(day from SYSDATE) THEN
                IF EXTRACT(month from :OLD.data_angajare) = EXTRACT(month from SYSDATE)
THEN
                    DBMS_OUTPUT.PUT_LINE('Delete invalid; se introduce inapoi randul!');
                    auxTrig11.memo.EXTEND;
                    SELECT
:OLD.nume,:OLD.cod_angajat,:OLD.id_magazin,:OLD.tip_angajat,:OLD.data_angajare,:OLD.id_se
diu,:OLD.salariu
                    INTO auxTrig11.memo(auxTrig11.memo.LAST)
                    FROM DUAL;
                    auxTrig11.actiune.EXTEND;
                    auxTrig11.actiune(auxTrig11.actiune.LAST) := 'delete';
                END IF;
            END IF;
        END IF;

        IF INSERTING THEN
            IF :NEW.id_sediu IS NOT NULL AND :NEW.tip_angajat <> 'paznic' THEN
                DBMS_OUTPUT.PUT_LINE('Insert invalid; se va sterge randul!');
                auxTrig11.memo.EXTEND;
                SELECT
:NEW.nume,:NEW.cod_angajat,:NEW.id_magazin,:NEW.tip_angajat,:NEW.data_angajare,:NEW.id_se
diu,:NEW.salariu
                INTO auxTrig11.memo(auxTrig11.memo.LAST)
                FROM DUAL;
                auxTrig11.actiune.EXTEND;
                auxTrig11.actiune(auxTrig11.actiune.LAST) := 'insert';
            END IF;
        END IF;

        IF UPDATING THEN
            IF :NEW.id_sediu IS NOT NULL AND :NEW.tip_angajat <> 'paznic' THEN
                DBMS_OUTPUT.PUT_LINE('Update invalid; va ramane acelasi rand!');
                auxTrig11.memo.EXTEND;
```

```
                SELECT
:OLD.nume,:OLD.cod_angajat,:OLD.id_magazin,:OLD.tip_angajat,:OLD.data_angajare,:OLD.id_se
diu,:OLD.salariu
                INTO auxTrig11.memo(auxTrig11.memo.LAST)
                FROM DUAL;
                auxTrig11.actiune.EXTEND;
                auxTrig11.actiune(auxTrig11.actiune.LAST) := 'update';
            END IF;
        END IF;
    END IF;
END;
/
CREATE OR REPLACE TRIGGER trig_ex11after
AFTER INSERT OR UPDATE OR DELETE ON ANGAJAT
DECLARE
BEGIN

    IF auxTrig11.cond2 = 1 THEN
        auxTrig11.cond1 := 0;
        auxTrig11.cond2 := 0;
        FOR i in auxTrig11.actiune.FIRST..auxTrig11.actiune.LAST LOOP
            IF auxTrig11.actiune(i)='delete' THEN
                INSERT INTO ANGAJAT VALUES auxTrig11.memo(i);
                DBMS_OUTPUT.PUT_LINE('S-a reintrodus randul');
            END IF;
            IF auxTrig11.actiune(i)='insert' THEN
                DELETE FROM ANGAJAT WHERE cod_angajat = auxTrig11.memo(i).cod_angajat;
                DBMS_OUTPUT.PUT_LINE('S-a sters randul');
            END IF;
            IF auxTrig11.actiune(i)='update' THEN
                UPDATE ANGAJAT
                SET nume = auxTrig11.memo(i).nume,id_magazin =
auxTrig11.memo(i).id_magazin,tip_angajat = auxTrig11.memo(i).tip_angajat,data_angajare =
auxTrig11.memo(i).data_angajare,id_sediu =
auxTrig11.memo(i).id_sediu,salariu=auxTrig11.memo(i).salariu
```

```
                WHERE cod_angajat = auxTrig11.memo(i).cod_angajat;
                DBMS_OUTPUT.PUT_LINE('S-a resetat randul la valoarea precedenta');
            END IF;
        END LOOP;

        auxTrig11.actiune.DELETE;
        auxTrig11.memo.DELETE;
    END IF;
    auxTrig11.cond1 := 1;
    auxTrig11.cond2 := 1;
END;
/
INSERT INTO ANGAJAT VALUES('Random',60,1,'curier',SYSDATE,8,3500);
/
INSERT INTO ANGAJAT VALUES('Random',60,1,'curier',SYSDATE,null,3500);
/
DELETE FROM ANGAJAT WHERE cod_angajat = 60;
/
UPDATE ANGAJAT SET id_sediu = 9 WHERE cod_angajat = 60;
/
SELECT cod_angajat, id_sediu FROM ANGAJAT WHERE cod_angajat = 60; --este null, nu s-a
modificat
```



12.

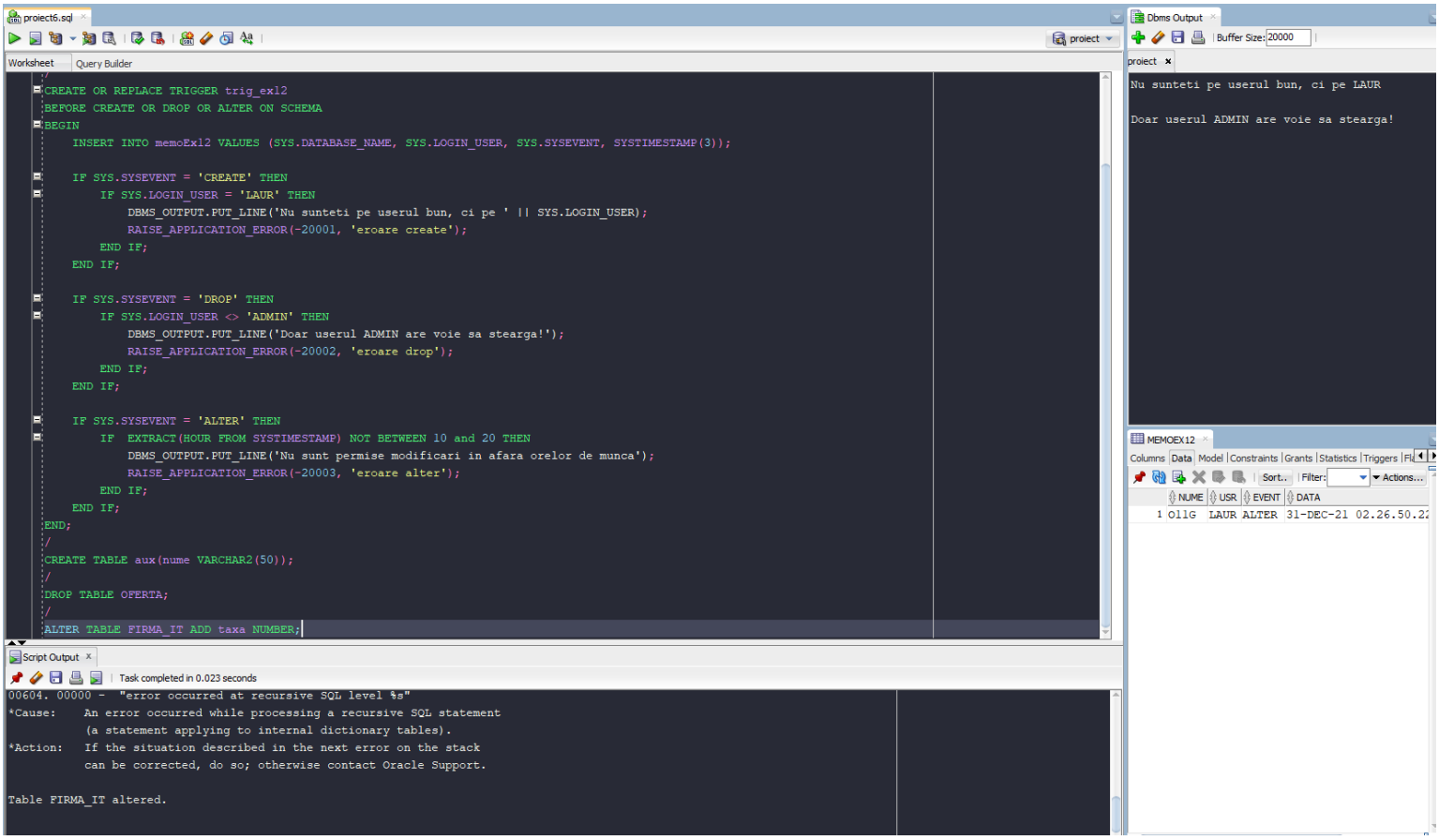
Problema:

Creati un nou trigger ce se declanseaza inainte de o comanda LDD ce verifica urmatoarele : Toti userii mai putin userul ‘LAUR’ au voie sa creeze, doar userul ‘ADMIN’ are voie sa stearga iar alter se poate da doar in intervalul orelor de munca 10 – 20. Memorati intr-un tabel date despre instructiunile care au rulat fara eroare.

Codul:

```
CREATE TABLE memoEx12
(nume VARCHAR2(50),
usr VARCHAR2(50),
event VARCHAR2(50),
data VARCHAR2(50));
/
CREATE OR REPLACE TRIGGER trig_ex12
BEFORE CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
    INSERT INTO memoEx12 VALUES (SYS.DATABASE_NAME, SYS.LOGIN_USER, SYS.SYSEVENT,
SYSTIMESTAMP(3));

    IF SYS.SYSEVENT = 'CREATE' THEN
        IF SYS.LOGIN_USER = 'LAUR' THEN
            DBMS_OUTPUT.PUT_LINE('Nu sunteti pe userul bun, ci pe ' || SYS.LOGIN_USER);
            RAISE_APPLICATION_ERROR(-20001, 'eroare create');
        END IF;
    END IF;

    IF SYS.SYSEVENT = 'DROP' THEN
        IF SYS.LOGIN_USER <> 'ADMIN' THEN
            DBMS_OUTPUT.PUT_LINE('Doar userul ADMIN are voie sa stearga!');
            RAISE_APPLICATION_ERROR(-20002, 'eroare drop');
        END IF;
```

```
    END IF;

    IF SYS.SYSEVENT = 'ALTER' THEN
        IF  EXTRACT(HOUR FROM SYSTIMESTAMP) NOT BETWEEN 10 and 20 THEN
            DBMS_OUTPUT.PUT_LINE('Nu sunt permise modificari in afara orelor de munca');
            RAISE_APPLICATION_ERROR(-20003, 'eroare alter');
        END IF;
    END IF;
END;
/
CREATE TABLE aux(nume VARCHAR2(50));
/
DROP TABLE OFERTA;
/
ALTER TABLE FIRMA_IT ADD taxa NUMBER;
```



13.

```
CREATE OR REPLACE TYPE tip_rezultat_ex8 IS OBJECT(primu numeric(5,4), al_doilea
numeric(5,4));

--NU se pot declara OBIECTE in pachet.

/

CREATE OR REPLACE PACKAGE pachet_proiect AS

    TYPE tablou_numere IS TABLE OF NUMBER;

    TYPE tablou_indexat_string IS TABLE OF tablou_numere INDEX BY STRING(3000);

    TYPE tablou_string IS TABLE OF STRING(3000);

    TYPE afisare IS REF CURSOR RETURN magazin%ROWTYPE;

    TYPE tipprocente IS TABLE OF numeric(5,2);


    PROCEDURE ex6;

    PROCEDURE ex7;

    FUNCTION ex8(numele fabrica.nume%TYPE DEFAULT 'Fabrica de becuri') RETURN
tip_rezultat_ex8;

    PROCEDURE ex9(numele distribuitor.nume%TYPE);

END pachet_proiect;

/

CREATE OR REPLACE PACKAGE BODY pachet_proiect AS


    PROCEDURE ex6

    IS

        programatori_dupa_limbaj tablou_indexat_string;

        limbaje tablou_string := tablou_string();

        aux_limbaje tablou_string := tablou_string();

        aux_lim VARCHAR(50);

        nr NUMBER := 1;

        firma NUMBER;

    BEGIN
```

```
        FOR rand IN (SELECT * FROM PROGRAMATOR) LOOP
             SELECT trim(regexp_substr(rand.limbaje_cunoscute, '[^ ]+', 1, LEVEL))
             BULK COLLECT INTO aux_limbaje
             FROM DUAL
             CONNECT BY regexp_substr(rand.limbaje_cunoscute , '[^ ]+', 1, LEVEL) IS NOT
NULL;



             FOR i in aux_limbaje.FIRST..aux_limbaje.LAST LOOP
                IF aux_limbaje(i) member OF limbaje THEN
                    NULL;
                ELSE
                    limbaje.EXTEND;
                    limbaje(limbaje.LAST) := aux_limbaje(i);
                END IF;
             END LOOP;

             aux_limbaje.DELETE;
        END LOOP;

        FOR i in limbaje.first..limbaje.last LOOP
            SELECT cod_programator
            BULK COLLECT INTO programatori_dupa_limbaj(limbaje(i))
            FROM programator
            WHERE INSTR(limbaje_cunoscute, limbaje(i)) != 0;
        END LOOP;



        aux_lim := programatori_dupa_limbaj.FIRST;
        WHILE aux_lim IS NOT NULL LOOP
            DBMS_OUTPUT.PUT_LINE(aux_lim);
            DBMS_OUTPUT.PUT_LINE('~~~~~~~~~~');
            FOR j in
programatori_dupa_limbaj(aux_lim).FIRST..programatori_dupa_limbaj(aux_lim).LAST LOOP
```

```
                SELECT id_firma INTO firma FROM programator WHERE cod_programator =
programatori_dupa_limbaj(aux_lim)(j);
                DBMS_OUTPUT.PUT_LINE(nr || ') Programatorul cu codul ' ||
programatori_dupa_limbaj(aux_lim)(j) || ' de la firma cu codul ' || firma);
                nr := nr+1;
            END LOOP;
            nr := 1;
            aux_lim := programatori_dupa_limbaj.NEXT(aux_lim);
            DBMS_OUTPUT.PUT_LINE('');
        END LOOP;
    END ex6;
```

PROCEDURE ex7

```
    IS
        CURSOR mag IS
           ( SELECT DISTINCT id_magazin
            FROM VANZARE
            WHERE id_distribuitor IN
                (SELECT oferta.id_distribuitor
                 FROM OFERTA
                 WHERE cod_produs IN (SELECT cod_produs FROM produs WHERE pret >= 2000) )

            INTERSECT

            SELECT DISTINCT id_magazin
            FROM promovare
            WHERE url_reclama IN
                (SELECT url_reclama
                 FROM reclama
                 WHERE tip_reclama = 'imagine')
            );

        CURSOR pretmax (cod magazin.id_magazin%TYPE) IS
            (SELECT MAX(pret)
```

```
         FROM produs
         WHERE cod_produs IN (SELECT cod_produs
                              FROM oferta
                              WHERE id_distribuitor IN (SELECT id_distribuitor
                                                        FROM vanzare
                                                        WHERE id_magazin = cod))
        );
    CURSOR pretmin (cod magazin.id_magazin%TYPE) IS
        (SELECT MIN(pret)
         FROM produs
         WHERE cod_produs IN (SELECT cod_produs
                              FROM oferta
                              WHERE id_distribuitor IN (SELECT id_distribuitor
                                                        FROM vanzare
                                                        WHERE id_magazin = cod))
        );
    suma produs.pret%TYPE;
    afis afisare;
    aux magazin%ROWTYPE;
    nume magazin.nume%TYPE;
    taxa magazin.taxa%TYPE;
    cd magazin.id_magazin%TYPE;
BEGIN

    FOR m IN mag LOOP
        OPEN pretmax(m.id_magazin);
        FETCH pretmax INTO suma;
        UPDATE magazin
        SET taxa = suma*0.1
        WHERE id_magazin = m.id_magazin;
        CLOSE pretmax;
    END LOOP;
```

```
        FOR m IN  ( SELECT id_magazin FROM Magazin
                   WHERE id_magazin NOT IN (SELECT DISTINCT id_magazin
                    FROM VANZARE
                    WHERE id_distribuitor IN
                        (SELECT oferta.id_distribuitor
                         FROM OFERTA
                         WHERE cod_produs IN (SELECT cod_produs FROM produs WHERE pret >=
2000) )

                    INTERSECT

                    SELECT DISTINCT id_magazin
                    FROM promovare
                    WHERE url_reclama IN
                        (SELECT url_reclama
                         FROM reclama
                         WHERE tip_reclama = 'imagine') ) ) LOOP

            OPEN pretmin(m.id_magazin);
            FETCH pretmin INTO suma;
            UPDATE magazin
            SET taxa = suma*0.2
            WHERE id_magazin = m.id_magazin;
            CLOSE pretmin;
        END LOOP;

        DBMS_OUTPUT.PUT_LINE('TAXA MARE:');
        DBMS_OUTPUT.PUT_LINE('-------------');
        OPEN afis FOR SELECT * FROM magazin WHERE taxa>199;
        LOOP
            FETCH afis into aux;
            EXIT WHEN afis%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa mare: ' ||
aux.taxa);
```

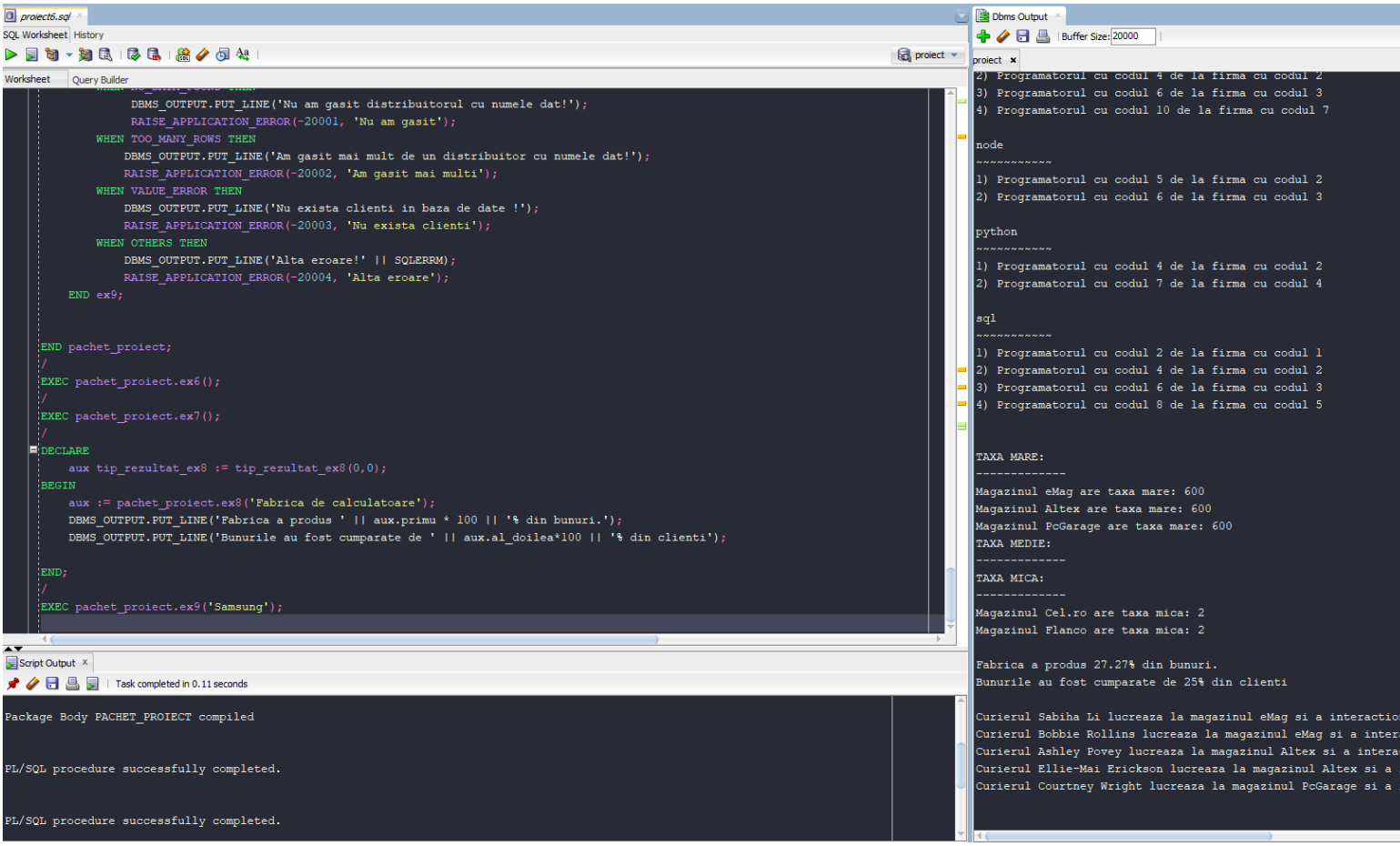
```
        END LOOP;
        CLOSE afis;
        DBMS_OUTPUT.PUT_LINE('TAXA MEDIE:');
        DBMS_OUTPUT.PUT_LINE('-------------');
        OPEN afis FOR SELECT * FROM magazin WHERE taxa BETWEEN 30 and 199;
        LOOP
            FETCH afis into aux;
            EXIT WHEN afis%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa medie: ' ||
aux.taxa);
        END LOOP;
        CLOSE afis;
        DBMS_OUTPUT.PUT_LINE('TAXA MICA:');
        DBMS_OUTPUT.PUT_LINE('-------------');
        OPEN afis FOR SELECT * FROM magazin WHERE taxa< 30;
        LOOP
            FETCH afis into aux;
            EXIT WHEN afis%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE('Magazinul ' || aux.nume || ' are taxa mica: ' ||
aux.taxa);
        END LOOP;
        CLOSE afis;

    END ex7;

    FUNCTION ex8(numele fabrica.nume%TYPE DEFAULT 'Fabrica de becuri')
    RETURN tip_rezultat_ex8
    IS
        rezultat tip_rezultat_ex8 := tip_rezultat_ex8(0,0);
        numar_prod NUMBER;
        numar_bun NUMBER;
        numar_ach NUMBER;
        numar_clienti NUMBER;
        cod fabrica.cod_fabrica%TYPE;
```

```
    BEGIN
        SELECT cod_fabrica INTO cod FROM fabrica WHERE nume = numele;


        SELECT COUNT(*)
        INTO numar_prod
        FROM PRODUS p JOIN PRODUCATOR pr ON p.id_producator = pr.id_producator JOIN
PARTENERIAT pa ON pa.id_producator = pr.id_producator
        WHERE cod_fabrica = cod;


        SELECT COUNT(*)
        INTO numar_bun
        FROM PRODUS
        WHERE tip_produs = 'bun';


        SELECT COUNT(*)
        INTO numar_ach
        FROM(SELECT DISTINCT id_client
            FROM ACHIZITIE ac JOIN PRODUS p ON ac.cod_produs=p.cod_produs JOIN PRODUCATOR
pr ON p.id_producator = pr.id_producator JOIN PARTENERIAT pa ON pa.id_producator =
pr.id_producator
            WHERE cod_fabrica = cod);


        SELECT COUNT(*)
        INTO numar_clienti
        FROM CLIENT_;


        rezultat.primu := numar_prod/numar_bun;
        rezultat.al_doilea := numar_ach/numar_clienti;
        RETURN rezultat;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista fabrica cu numele dat');
            RAISE_APPLICATION_ERROR (-20003, 'Nu exista fabrica cu numele dat');
        WHEN TOO_MANY_ROWS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Exista mai multe fabrici nu numele dat');
        RAISE_APPLICATION_ERROR (-20001, 'Exista mai multe fabrici nu numele dat');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
        RAISE_APPLICATION_ERROR (-20002, 'Alta eroare!');
END ex8;

PROCEDURE ex9 (numele distribuitor.nume%TYPE)
IS
    coddist distribuitor.id_distribuitor%TYPE;
    nrclienti NUMBER;
    curieri tablou_string := tablou_string();
    magazine tablou_string := tablou_string();
    numere tablou_numere := tablou_numere();
    procente tipprocente := tipprocente();
BEGIN
    SELECT id_distribuitor INTO coddist
    FROM distribuitor
    WHERE nume = numele;

    SELECT COUNT(*) INTO nrclienti
    FROM CLIENT_;

    SELECT a.nume,m.nume,COUNT(c.id_client)
    BULK COLLECT INTO curieri, magazine, numere
    FROM angajat a JOIN produs p ON a.cod_angajat = p.cod_curier
        JOIN oferta o ON p.cod_produs = o.cod_produs
        JOIN distribuitor d ON o.id_distribuitor = d.id_distribuitor
        JOIN magazin m ON a.id_magazin = m.id_magazin
        JOIN achizitie ac ON p.cod_produs = ac.cod_produs
        JOIN client_ c ON c.id_client = ac.id_client
    WHERE d.id_distribuitor = coddist
    GROUP BY a.cod_angajat, a.nume, d.id_distribuitor, m.nume;
```

```
        FOR i IN numere.FIRST..numere.LAST LOOP
            procente.EXTEND;
            procente(procente.LAST) := numere(i)/nrclienti * 100;
        END LOOP;

        FOR i IN curieri.FIRST..curieri.LAST LOOP
            DBMS_OUTPUT.PUT_LINE('Curierul ' || curieri(i) || ' lucreaza la magazinul '
|| magazine(i) || ' si a interactionat cu ' || procente(i) || '% din clienti.');
        END LOOP;

    EXCEPTION
        WHEN NO_DATA_FOUND THEN
             DBMS_OUTPUT.PUT_LINE('Nu am gasit distribuitorul cu numele dat!');
             RAISE_APPLICATION_ERROR(-20001, 'Nu am gasit');
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Am gasit mai mult de un distribuitor cu numele dat!');
            RAISE_APPLICATION_ERROR(-20002, 'Am gasit mai multi');
        WHEN VALUE_ERROR THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista clienti in baza de date !');
            RAISE_APPLICATION_ERROR(-20003, 'Nu exista clienti');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Alta eroare!' || SQLERRM);
            RAISE_APPLICATION_ERROR(-20004, 'Alta eroare');
    END ex9;



END pachet_proiect;
/
EXEC pachet_proiect.ex6();
/
EXEC pachet_proiect.ex7();
/
```

```
DECLARE
    aux tip_rezultat_ex8 := tip_rezultat_ex8(0,0);
BEGIN
    aux := pachet_proiect.ex8('Fabrica de calculatoare');
    DBMS_OUTPUT.PUT_LINE('Fabrica a produs ' || aux.primu * 100 || '% din bunuri.');
    DBMS_OUTPUT.PUT_LINE('Bunurile au fost cumparate de ' || aux.al_doilea*100 || '% din clienti');


END;
/
EXEC pachet_proiect.ex9('Samsung');
```



14.

Continutul pachetului:

Functia 1: Intoarceti un tabel cu toate firmele IT implicate(atat sa hosteze serverul paginii si sa le puna la dispozitie programatori) in realizarea paginilor web prin intermediul carora isi vinde produsele magazinul cu numele dat ca parametru.

Functia 2: Intoarceti toti programatorii de la toate firmele IT intoarse de *Functia 1* care cunosc un limbaj de programare dat ca parametru. (are 2 parametri: numele dat ca parametru Functiei 1 si limbajul parametru pentru Functia 2).

Procedura 1: Afisati numele, magazinul la care lucreaza, data angajarii si vechimea(in zile) in firma si ziua saptamanii (in litere) in care au fost angajati a tuturor angajatilor ce au inceput serviciul (au fost angajati) intr-o zi de miercuri si lucreaza intr-un magazin ce-si face reclama prin cel putin o reclama pe youtube sau care au fost angajati intr-o zi de vineri si lucreaza intr-un magazin ce-si face reclama prin putin o reclama pe siteul www.reclama.

Procedura 2: Afisati date despre toate magazinele in ordine descrescatoare dupa numarul de produse vandut. Utilizati un tip de cursor studiat.

Codul:

```
CREATE OR REPLACE PACKAGE pachet_14 AS

    TYPE tip_procedura IS RECORD(nume varchar2(50), magazin varchar2(50), data_angajare
date, vechime NUMBER, zi varchar2(50));

    TYPE tabel_firme IS TABLE OF firma_it%ROWTYPE;

    TYPE tabel_programatori IS TABLE OF programator%ROWTYPE;

    TYPE tabel_procedura IS TABLE OF tip_procedura;


    FUNCTION func_firme(numele magazin.nume%TYPE) RETURN tabel_firme;

    FUNCTION func_programatori(numele magazin.nume%TYPE, limbaj
programator.limbaje_cunoscute%TYPE) RETURN tabel_programatori;

    PROCEDURE proc_angajati;

    PROCEDURE proc_nrproduse;

END pachet_14;

/

CREATE OR REPLACE PACKAGE BODY pachet_14 AS


    FUNCTION func_firme(numele magazin.nume%TYPE) RETURN tabel_firme
```

```
    IS
        rez tabel_firme := tabel_firme();
        cod magazin.id_magazin%TYPE;
    BEGIN
        SELECT id_magazin INTO cod
        FROM magazin WHERE nume = numele;

        WITH pagini as (SELECT v.link_pagina aux
                        FROM VANZARE v
                        WHERE v.id_magazin = cod)
        SELECT *
        BULK COLLECT INTO rez
        FROM FIRMA_IT
        WHERE id_firma in (SELECT id_firma
                           FROM SERVER
                           WHERE id_server in (SELECT id_server
                                               FROM PAGINA_WEB
                                               WHERE link_pagina in (SELECT aux
                                                                     FROM pagini)))
            OR id_firma in(SELECT id_firma
                           FROM PROGRAMATOR
                           WHERE cod_programator in (SELECT cod_programator
                                                     FROM PROGRAM_
                                                     WHERE link_pagina in (SELECT aux
                                                                           FROM
pagini)
                           ));
        RETURN rez;
    EXCEPTION
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multe cu numele acesta!');
            RAISE_APPLICATION_ERROR(-20001,'prea multe');
        WHEN NO_DATA_FOUND THEN
```

```
            DBMS_OUTPUT.PUT_LINE('Nu exista magazin cu numele acesta!');
            RAISE_APPLICATION_ERROR(-20002,'niciunul');
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multe cu numele asta!');
            RAISE_APPLICATION_ERROR(-20003,'alta');
    END func_firme;



    PROCEDURE proc_angajati IS
        rez tabel_procedura := tabel_procedura();
    BEGIN
        SELECT nume,(SELECT nume
            FROM MAGAZIN mag
            WHERE mag.id_magazin = a.id_magazin)as magazin, data_angajare, ROUND(SYSDATE
- a.data_angajare ), DECODE(TO_CHAR(a.data_angajare,'DY'),
'WED','Miercuri','FRI','Vineri')
        BULK COLLECT INTO rez
        FROM ANGAJAT a
        WHERE ( NVL(TO_CHAR(a.data_angajare,'DY'),'A')='WED'
            AND a.id_magazin in (SELECT id_magazin
                                 FROM MAGAZIN
                                 WHERE id_magazin in (SELECT id_magazin
                                                      FROM PROMOVARE
                                                      WHERE url_reclama in (SELECT
url_reclama
                                                                            FROM RECLAMA rec
                                                                            WHERE
INSTR(LOWER(rec.url_reclama),'yt/') != 0)
                                                                            )
                                                                            ))
            OR
            (NVL(TO_CHAR(a.data_angajare,'DY'),'A')='FRI'
            AND a.id_magazin in (SELECT id_magazin
                                 FROM MAGAZIN
```

```
                                WHERE id_magazin in (SELECT id_magazin
                                                     FROM PROMOVARE
                                                     WHERE url_reclama in (SELECT
url_reclama
                                                                           FROM RECLAMA rec2
                                                                           WHERE
INSTR(LOWER(rec2.url_reclama),'www.reclama') != 0)
                                                                           )
                                                                           ));


        DBMS_OUTPUT.PUT_LINE('Nume / Magazin / Data angajare / Vechime / Zi angajare');

DBMS_OUTPUT.PUT_LINE('==========================================================');
        FOR i IN rez.FIRST..rez.LAST LOOP
            DBMS_OUTPUT.PUT_LINE(rez(i).nume || ' / ' || rez(i).magazin || ' / ' ||
rez(i).data_angajare || ' / ' || rez(i).vechime || ' / '  || rez(i).zi);
        END LOOP;
    END proc_angajati;



     FUNCTION func_programatori(numele magazin.nume%TYPE, limbaj
programator.limbaje_cunoscute%TYPE) RETURN tabel_programatori
     IS
        rez tabel_programatori := tabel_programatori();
        intermediar tabel_programatori := tabel_programatori();
        aux tabel_firme := tabel_firme();
        idaux firma_it.id_firma%TYPE;
        nrprog NUMBER;
     BEGIN
        aux := func_firme(numele);

        FOR i IN aux.FIRST..aux.LAST LOOP
            idaux := aux(i).id_firma;

            SELECT COUNT(*)
```

```
            INTO nrprog
            FROM (SELECT * FROM PROGRAMATOR WHERE INSTR(limbaje_cunoscute, limbaj) != 0 )
            WHERE cod_programator = idaux;
            IF nrprog > 0 THEN
                SELECT *
                BULK COLLECT INTO intermediar
                FROM (SELECT * FROM PROGRAMATOR WHERE INSTR(limbaje_cunoscute, limbaj) !=
0 )
                WHERE cod_programator = idaux;

                FOR j in intermediar.FIRST..intermediar.LAST LOOP
                    rez.EXTEND;
                    rez(rez.LAST) := intermediar(j);
                END LOOP;
            END IF;

            intermediar.DELETE;
        END LOOP;

        return rez;
     END func_programatori;



     PROCEDURE proc_nrproduse
     IS
     BEGIN
        FOR linie IN (SELECT mag.*,(SELECT COUNT(*) --ciclu cursor cu subcereri
                        FROM MAGAZIN m JOIN VANZARE v ON(m.id_magazin = v.id_magazin)
                                JOIN DISTRIBUITOR d ON(v.id_distribuitor =
d.id_distribuitor)
                                JOIN OFERTA o ON(d.id_distribuitor = o.id_distribuitor)
                                JOIN PRODUS p ON(o.cod_produs = p.cod_produs)
                        WHERE m.id_magazin = mag.id_magazin) nr
                        FROM MAGAZIN mag
```

```
                        ORDER BY nr desc) LOOP

            DBMS_OUTPUT.PUT_LINE(linie.nume || ' vinde ' || linie.nr || ' produse');
        END LOOP;
     END proc_nrproduse;

END pachet_14;
/
DECLARE
    aux pachet_14.tabel_firme := pachet_14.tabel_firme();
BEGIN
    aux := pachet_14.func_firme('Cel.ro');
    FOR i IN aux.FIRST..aux.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(aux(i).id_firma || ' ' || aux(i).nr_angajati);
    END LOOP;
END;
/
EXEC pachet_14.proc_angajati();
/
DECLARE
    aux pachet_14.tabel_programatori := pachet_14.tabel_programatori();
BEGIN
    aux := pachet_14.func_programatori('Cel.ro','cpp');
    FOR i IN aux.FIRST..aux.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(aux(i).cod_programator || ' ' || aux(i).limbaje_cunoscute ||
' ' || aux(i).id_firma);
    END LOOP;
END;
/
EXEC pachet_14.proc_nrproduse();
```

proiect6.sql | PROGRAMATOR

SQL Worksheet | History

0.064 seconds

proiect

Worksheet | Query Builder

```
                        WHERE m.id_magazin = mag.id_magazin) nr
                        FROM MAGAZIN mag
                        ORDER BY nr desc) LOOP

            DBMS_OUTPUT.PUT_LINE(linie.nume || ' vinde ' || linie.nr || ' produse');
        END LOOP;
    END proc_nrproduse;

END pachet_14;
/
DECLARE
    aux pachet_14.tabel_firme := pachet_14.tabel_firme();
BEGIN
    aux := pachet_14.func_firme('Cel.ro');
    FOR i IN aux.FIRST..aux.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(aux(i).id_firma || ' ' || aux(i).nr_angajati);
    END LOOP;
END;
/
EXEC pachet_14.proc_angajati();
/
DECLARE
    aux pachet_14.tabel_programatori := pachet_14.tabel_programatori();
BEGIN
    aux := pachet_14.func_programatori('Cel.ro','cpp');
    FOR i IN aux.FIRST..aux.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(aux(i).cod_programator || ' ' || aux(i).limbaje_cunoscute || ' ' || aux(i).id_firma);
    END LOOP;
END;
/
EXEC pachet_14.proc_nrproduse();
```

Script Output | Query Result

Task completed in 0.064 seconds

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.
```

Dbms Output

Buffer Size: 20000

proiect

```
1 100
2 10
4 2000

Nume / Magazin / Data angajare / Vechime / Zi angajare
=======================================================
Devon Higgins / eMag / 07-MAR-14 / 2858 / Vineri
Nahla Duke / Cel.ro / 27-APR-05 / 6094 / Miercuri
Jade Douglas / Cel.ro / 29-APR-20 / 613 / Miercuri
Steffan King / PcGarage / 25-JUL-03 / 6736 / Vineri
Vera Hibbert / PcGarage / 19-SEP-18 / 1201 / Miercuri
Justine Franks / PcGarage / 25-JAN-06 / 5821 / Miercuri
Random / eMag / 31-DEC-21 / 2 / Vineri

1 cpp 1
4 python javascript cpp sql 2

eMag vinde 26 produse
Altex vinde 22 produse
Flanco vinde 13 produse
Cel.ro vinde 13 produse
PcGarage vinde 9 produse
```