Gestiunea unei Firme de produse IT

Zabava Claudiu-Alexandru

Grupa 243

Aplicația realizează gestiunea unei firme de produse IT prin intermediul programului Oracle. Am ales această temă întrucât am dorit să îmi combin pasiunea pentru calculatoare cu abilitatile si cunostintele informatice.

1 Punctul 1:

Modelul de date ales va gestiona informaţii legate de o firma ce se ocupa cu asamblarea si vanzarea unor produse IT precum : monitoare, PC-uri si laptop-uri. Pentru a intelege cu exactitate principiul din spatele acestui proiect, putem compara aceasta firma cu compania HP(Hewlett-Packard), care produce obiectele enumerate mai sus folosindu-se de componente de la un producator extern (eg: Intel Core, nVidia, etc..). Firma creata de noi are scopuri asemanatoare cu exemplul dat, diferenta fiind facuta de faptul ca noi ne propunem sa creem produse la comanda, insemnand ca un produs o sa fie predestinat unui cumparator cunoscut.

Astfel, firma noastra se ocupa cu crearea / asamblarea unor produse IT la cererea clientilor. Firma nu isi produce singura componentele folosite in aceste produse , acestea sunt distribuite de catre furnizori de la anumiti producatori externi ce se ocupa in special de crearea unui anumit tip de componenta(eg: procesor grafic, microprocesor, placa de baza, display).

Asamblarea sau crearea propriuzisa a produselor pe care firma urmeaza sa le vanda clientiloreste realizata de salariatii firmei. Fiecare salariat are aptitudini specifice, care il fac sa fie persoana ideala pentru o anumita echipa. Echipele au propriile lor sedii, fiind impartite pe mai multe regiuni si orase. Echipa primeste unul sau mai multe taskuri din procesul de creare al unui produs. Task-urile sunt rezolvate pe rand de catre intraga echipa. Pe langa aceste aptitudini, sunt retinute si informatii de baza despre un angajat, cum ar fi adresa, numele complet, CNP-ul si alte date de contact cat si o dovada a acestor aptitudini (diploma care atesta completarea unor studii superioare in acest domeniu, scrisoare de recomandare de la un alt loc de munca similar). Angajatii au cate un superior numit lider. Liderul unui angajat trebuie sa fie in aceeasi echipa cu acesta deoarece el este lider de echipa.

Un angajat poate sa demisioneze dintr-o anumita echipa si sa intre in alta. In situatii ca acestea, datele respectivului angajat trebuie salvate in arhiva, alaturi de data la care a inceput sa lucreze pentru echipa respectiva, cat si data la care a parasit acea echipa.

Pretul unui produs este fixat astfel incat sa acopere costul componentelor.

Produsele firmei sunt indexate printr-o valoare numerica unica numita *cod-fabricatie* astfel incat, indiferent de locatia in care se afla echipele , nu o sa existe doua produse care impart acelasi id.

Clientii pot achizitiona produsele finale pe care firma le vinde prin plasarea unor comenzi telefonic sau pe website-ul firmei, specificand cantitatea (numarul de produse de tipul x), dar si tipul de livrare :

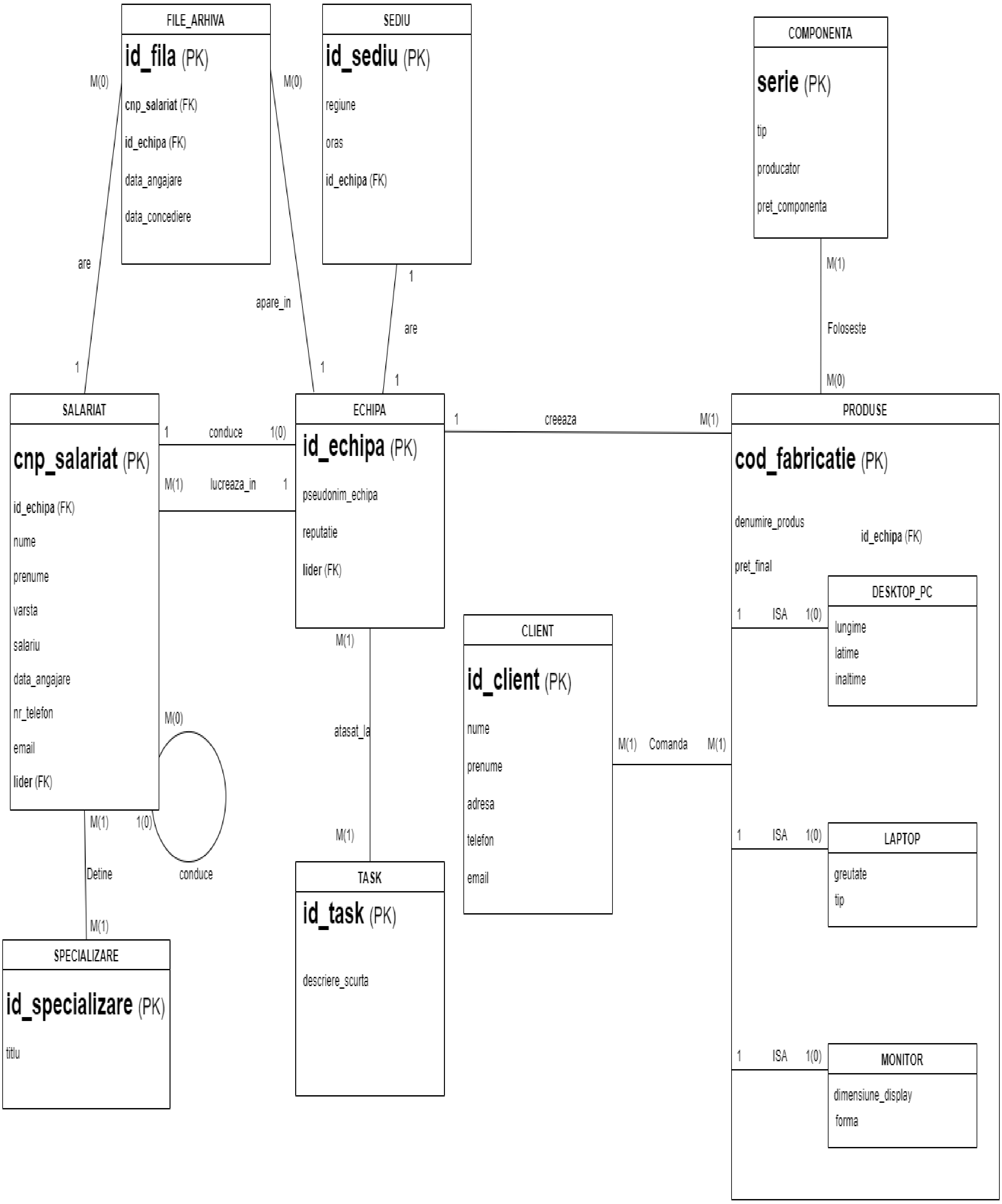
premium– firma se ocupa personal de livrarea produsului.

standard – sunt folosite mijloace normale de livrare a produselor

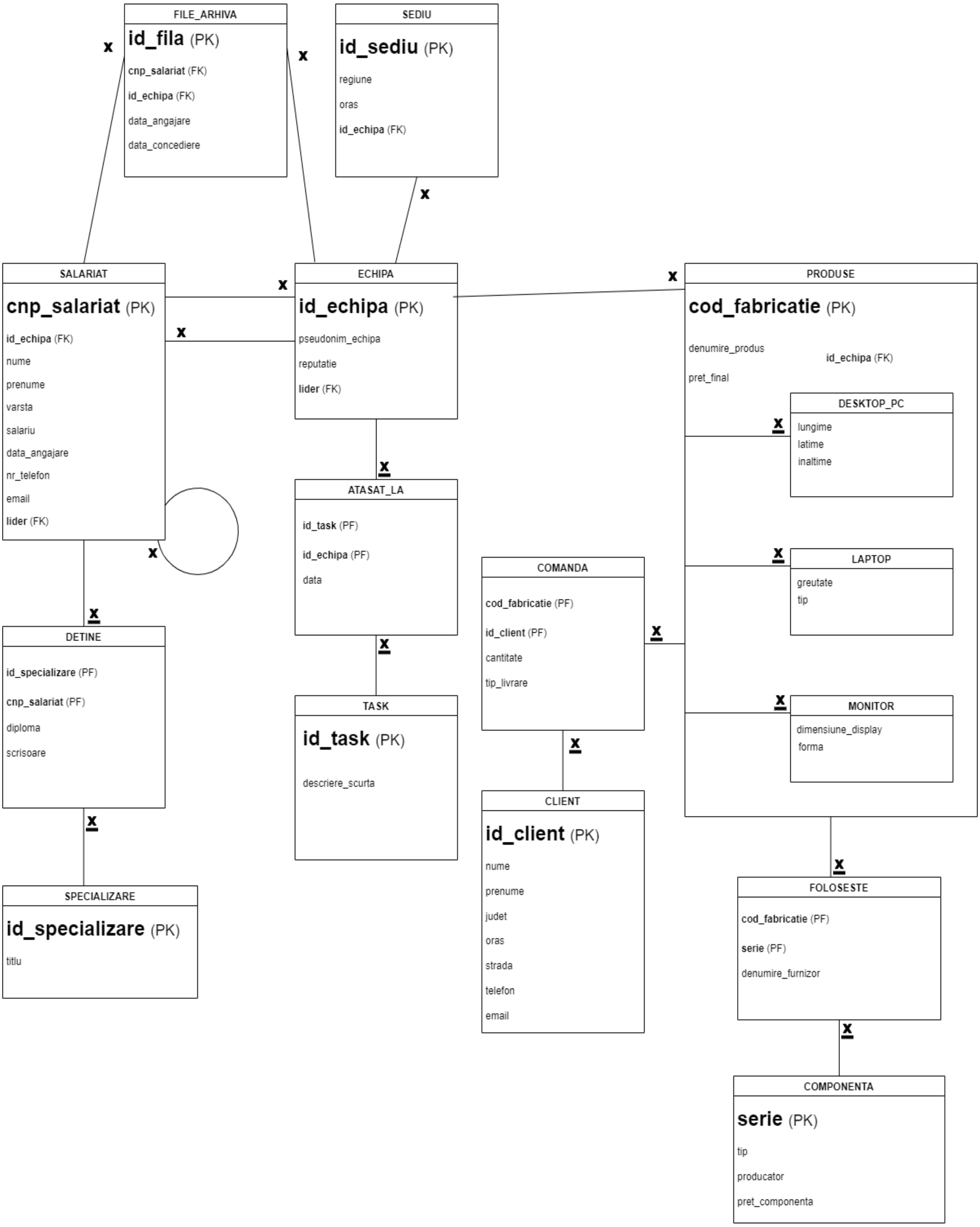
La plasarea unei comenzi sunt retinute informatii de baza despre un client, cum ar fi numele complet, adresa unde el o sa primeasca comanda si cateva date de contact.

Observatie: Intrucat noi lucram pe conexiunea SQL:GR243, comuna tuturor membrilor grupei, pentru a evita eventualele conflicte, la implementarea in SQL numele tabelelor prezentate in diagrama conceptuala sunt insotite si de sufixul ”-zxc”.

2 Punctul 2: Diagrama ER



3 Punctul 3: Diagrama Conceptuala



4 Punctul 4: Implementare in SQL

```
CREATE TABLE ECHIPA_ZXC
(
 id_echipa number(5,0) NOT NULL PRIMARY KEY,
 pseudonim_echipa varchar2(32) NOT NULL UNIQUE,
 reputatie number(5,2),
 CONSTRAINT reputatie_Ck CHECK (reputatie >= 0),
 lider number(10,0)
 );

CREATE TABLE SALARIAT_ZXC
(
 cnp_salariat number(10,0) NOT NULL PRIMARY KEY,
 id_echipa number(5,0) NOT NULL,
 CONSTRAINT fk_id_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA_ZXC(id_echipa),
 nume varchar2(32) NOT NULL,
 prenume varchar2(32) NOT NULL,
 varsta number(3,0) NOT NULL,
 CONSTRAINT varsta_Ck CHECK (varsta >= 18),
 salariu number(10,2) NOT NULL,
 CONSTRAINT salariu_check CHECK (salariu >0),
 data_angajare date,
 nr_telefon varchar2(32) NOT NULL UNIQUE,
 email varchar2(32) UNIQUE,
 lider number(10,0),
 CONSTRAINT fk_lider FOREIGN KEY (lider) REFERENCES SALARIAT_ZXC(cnp_salariat)
);

ALTER TABLE ECHIPA_ZXC
ADD CONSTRAINT fk_echipa_lider FOREIGN KEY (lider) REFERENCES SALARIAT_ZXC(cnp_salariat);

CREATE TABLE SPECIALIZARE_ZXC
(
 id_specializare number(5,0) NOT NULL PRIMARY KEY,
 titlu varchar2(32) NOT NULL
);


CREATE TABLE DETINE_ZXC
(
 id_specializare number(5,0) NOT NULL,
 cnp_salariat number(10,0) NOT NULL,
 diploma varchar2(128) NOT NULL,
 scrisoare varchar2(128),
 CONSTRAINT fk_detine_id_specializare FOREIGN KEY (id_specializare) REFERENCES SPECIALIZARE_ZXC(
     ↪ id_specializare),
 CONSTRAINT fk_detine_cnp_salariat FOREIGN KEY (cnp_salariat) REFERENCES SALARIAT_ZXC(cnp_salariat)
     ↪ ,
 CONSTRAINT pk_detine PRIMARY KEY (id_specializare, cnp_salariat)
);

CREATE TABLE SEDIU_ZXC
(
 id_sediu number(5,0) NOT NULL PRIMARY KEY,
 regiune varchar2(32) NOT NULL,
 oras varchar2(32) NOT NULL,
 id_echipa number(5,0) NOT NULL UNIQUE,
 CONSTRAINT fk_sediu_id_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA_ZXC(id_echipa)
);

CREATE TABLE FILE_ARHIVA_ZXC
(
 id_fila number(5,0) NOT NULL PRIMARY KEY,
 id_echipa number(5,0),
 cnp_salariat number(10,0),
 data_angajare date NOT NULL,
 data_concediere date NOT NULL,
 CONSTRAINT data_check CHECK ( data_angajare <= data_concediere),
 CONSTRAINT fk_arhiva_id_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA_ZXC(id_echipa),
 CONSTRAINT fk_arhiva_salariat FOREIGN KEY (cnp_salariat) REFERENCES SALARIAT_ZXC(cnp_salariat)
);

CREATE TABLE TASK_ZXC
(
 id_task number(5,0) NOT NULL PRIMARY KEY,
 descriere_scurta varchar2(128) NOT NULL UNIQUE
);
```

```
CREATE TABLE ATASAT_LA_ZXC
(
 id_echipa number(5,0) NOT NULL,
 id_task number(5,0) NOT NULL,
 dataa date not NULL,
 CONSTRAINT fk_atasat_la_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA_ZXC(id_echipa),
 CONSTRAINT fk_atasat_la_task FOREIGN KEY (id_task) REFERENCES TASK_ZXC(id_task),
 CONSTRAINT pk_atasat_la PRIMARY KEY (id_echipa, id_task)
);


CREATE TABLE CLIENT_ZXC
(
 id_client number(5,0) NOT NULL PRIMARY KEY,
 nume varchar2(32) NOT NULL,
 prenume varchar2(32) NOT NULL,
 nr_telefon varchar2(32) NOT NULL UNIQUE,
 email varchar2(32) UNIQUE,
 judet varchar2(32) NOT NULL,
 oras varchar2(32) NOT NULL,
 strada varchar2(32) NOT NULL
);


CREATE TABLE COMPONENTA_ZXC
(
 serie number(5,0) NOT NULL PRIMARY KEY,
 tip varchar2(32) NOT NULL,
 producator varchar2(32) NOT NULL,
 pret_componenta number(10,2) NOT NULL,
 CONSTRAINT check_pret_c CHECK ( pret_componenta > 0) -- nu se cumpara componente pe gratis --
);


CREATE TABLE PRODUS_ZXC
(
 cod_fabricatie number(5,0) NOT NULL PRIMARY KEY,
 denumire_produs varchar2(32) NOT NULL UNIQUE,
 pret_produs number(10,2) NOT NULL,
 CONSTRAINT check_pret_p CHECK (pret_produs > 0),
 id_echipa number(5,0) NOT NULL,
 tip_produs varchar2(32) NOT NULL,
 CONSTRAINT produs_check CHECK (upper(tip_produs)='LAPTOP' or upper(tip_produs)='MONITOR' or
     ↪ upper(tip_produs)='DESKTOP_PC'),
 dimensiune_display number(10,2),
 CONSTRAINT dimensiune_check CHECK (dimensiune_display > 0),
 formaa varchar2(32),
 CONSTRAINT forma_monitor_check CHECK( upper(formaa)='NORMAL' or upper(formaa)='CURBAT'),
 greutate number(10,2),
 CONSTRAINT greutate_check CHECK (greutate > 0),
 tip varchar2(32),
 CONSTRAINT tip_laptop_check CHECK( upper(tip)='NOTEBOOK' or upper(tip)='GAMING'),
 lungime number(10,2),
 CONSTRAINT lungime_check CHECK (lungime > 0),
 latime number(10,2),
 CONSTRAINT latime_check CHECK (latime > 0),
 inaltime number(10,2),
 CONSTRAINT inaltime_check CHECK (inaltime > 0),
 CONSTRAINT fk_produs_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA_ZXC(id_echipa)
);

CREATE TABLE COMANDA_ZXC
(
 id_client number(5,0) NOT NULL,
 cod_fabricatie number(5,0) NOT NULL,
 cantitate number(10,0) NOT NULL,
 CONSTRAINT cantitate_check CHECK (cantitate > 0),
 tip_livrare varchar2(32) NOT NULL,
 CONSTRAINT livrare_check CHECK (upper(tip_livrare)='STANDARD' or upper(tip_livrare)='PREMIUM'),
 CONSTRAINT fk_comanda_client FOREIGN KEY (id_client) REFERENCES CLIENT_ZXC(id_client),
 CONSTRAINT fk_comanda_produs FOREIGN KEY (cod_fabricatie) REFERENCES PRODUS_ZXC(cod_fabricatie),
 CONSTRAINT pk_comanda PRIMARY KEY (cod_fabricatie,id_client)
);


CREATE TABLE FOLOSESTE_ZXC
```

```
(
 cod_fabricatie number(5,0) NOT NULL,
 serie number(5,0),
 denumire_furnizor varchar2(32),
 CONSTRAINT fk_foloseste_produs FOREIGN KEY (cod_fabricatie) REFERENCES PRODUS_ZXC(cod_fabricatie),
 CONSTRAINT fk_foloseste_componenta FOREIGN KEY (serie) REFERENCES COMPONENTA_ZXC(serie),
 CONSTRAINT pk_foloseste PRIMARY KEY (cod_fabricatie,serie)
);
```

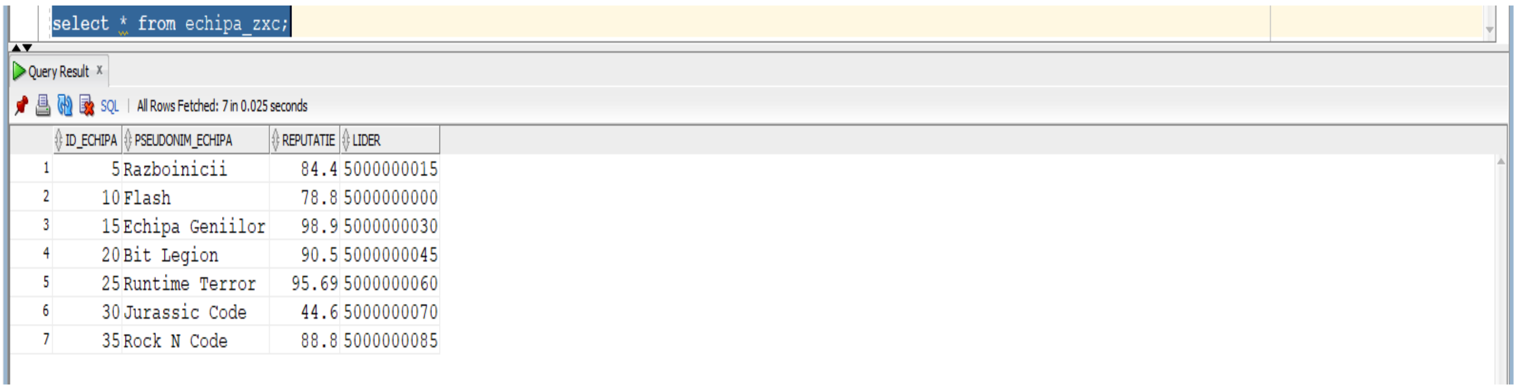
5 Punctul 5: Populare

```
−− Secventa pentru id echipa −−
CREATE SEQUENCE seq_echipa
INCREMENT by 5
START WITH 5
MAXVALUE 10000
NOCYCLE;

−− Secventa pentru id salariat −−
CREATE SEQUENCE seq_salariat
INCREMENT by 5
START WITH 5000000000
MAXVALUE 59999000000
NOCYCLE;
```

Vom avea urmatoarele tabele:

ECHIPA

```

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Razboinicii', 85.9, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Flash', 78.8, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Echipa_Geniilor', 98.9, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Bit_Legion', 90.5, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Runtime_Terror', 95.69, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Jurassic_Code', 44.6, NULL);

INSERT INTO ECHIPA_ZXC
VALUES(seq_echipa.nextval, 'Rock_N_Code', 88.8, NULL);
```



```
select * from echipa_zxc;
```

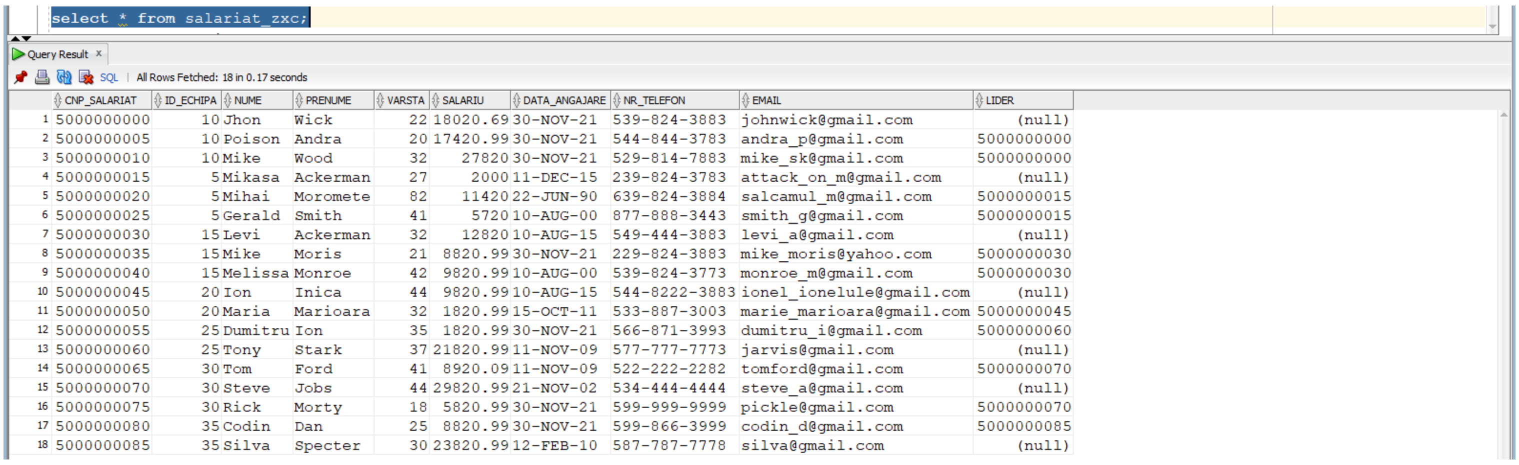
All Rows Fetched: 7 in 0.025 seconds

```
    ID_ECHIPA PSEUDONIM_ECHIPA REPUTATIE LIDER
1           5 Razboinicii           84.4 5000000015
2          10 Flash                 78.8 5000000000
3          15 Echipa Geniilor       98.9 5000000030
4          20 Bit Legion            90.5 5000000045
5          25 Runtime Terror       95.69 5000000060
6          30 Jurassic Code         44.6 5000000070
7          35 Rock N Code           88.8 5000000085
```

SALARIAT

```
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,10, 'Jhon', 'Wick',22,18020.69,sysdate,'539−824−3883','johnwick@gmail.com', NULL);
INSERT INTO SALARIAT_ZXC
```

```
VALUES(seq_salariat.nextval,10, ’Poison’, ’Andra’,20,17420.99,sysdate,’544−844−3783’,’andra_p@gmail.com’,
    ↪ 5000000000);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,10, ’Mike’, ’Wood’,32,27820.00,sysdate,’529−814−7883’,’mike_sk@gmail.com’, 5000000000);

INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,5, ’Mikasa’, ’Ackerman’,27,2000.00,TO_DATE(’11−12−2015’, ’DD−MM−YYYY’),’
    ↪ 239−824−3783’,’attack_on_m@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,5, ’Mihai’, ’Moromete’,82,11420.00,TO_DATE(’22−06−1990’, ’DD−MM−YYYY’),’
    ↪ 639−824−3884’,’salcamul_m@gmail.com’, 5000000015);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,5, ’Gerald’, ’Smith’,41,5720.00,TO_DATE(’10−08−2000’, ’DD−MM−YYYY’),’
    ↪ 877−888−3443’,’smith_g@gmail.com’, 5000000015);


INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,15, ’Levi’, ’Ackerman’,32,12820.00,TO_DATE(’10−08−2015’, ’DD−MM−YYYY’),’
    ↪ 549−444−3883’,’levi_a@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,15, ’Mike’, ’Moris’,21,8820.99,sysdate,’229−824−3883’,’mike_moris@yahoo.com’,5000000030)
    ↪ ;
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,15, ’Melissa’, ’Monroe’,42,9820.99,TO_DATE(’10−08−2000’, ’DD−MM−YYYY’),’
    ↪ 539−824−3773’,’monroe_m@gmail.com’, 5000000030);

INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,20, ’Ion’, ’Inica’,44,9820.99,TO_DATE(’10−08−2015’, ’DD−MM−YYYY’),’544−8222−3883’
    ↪ ,’ionel_ionelule@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,20, ’Maria’, ’Marioara’,32,1820.99,TO_DATE(’15−10−2011’, ’DD−MM−YYYY’),’
    ↪ 533−887−3003’,’marie_marioara@gmail.com’, NULL);

INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,25, ’Dumitru’, ’Ion’,35,1820.99,sysdate,’566−871−3993’,’dumitru_i@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,25, ’Tony’, ’Stark’,37,21820.99,TO_DATE(’11−11−2009’,’DD−MM−YYYY’),’
    ↪ 577−777−7773’,’jarvis@gmail.com’, NULL);

INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,30, ’Tom’, ’Ford’,41,8920.09,TO_DATE(’11−11−2009’,’DD−MM−YYYY’),’522−222−2282’,
    ↪ ’tomford@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,30, ’Steve’, ’Jobs’,44,29820.99,TO_DATE(’21−11−2002’,’DD−MM−YYYY’),’
    ↪ 534−444−4444’,’steve_a@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,30, ’Rick’, ’Morty’,18,5820.99,sysdate,’599−999−9999’,’pickle@gmail.com’, NULL);

INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,35, ’Codin’, ’Dan’,25,8820.99,sysdate,’599−866−3999’,’codin_d@gmail.com’, NULL);
INSERT INTO SALARIAT_ZXC
VALUES(seq_salariat.nextval,35, ’Silva’, ’Specter’,30,23820.99,TO_DATE(’12−02−2010’,’DD−MM−YYYY’),’
    ↪ 587−787−7778’,’silva@gmail.com’, NULL);
```



```
select * from salariat_zxc;
```

	CNP_SALARIAT	ID_ECHIPA	NUME	PRENUME	VARSTA	SALARIU	DATA_ANGAJARE	NR_TELEFON	EMAIL	LIDER
1	5000000000	10	Jhon	Wick	22	18020.69	30-NOV-21	539-824-3883	johnwick@gmail.com	(null)
2	5000000005	10	Poison	Andra	20	17420.99	30-NOV-21	544-844-3783	andra_p@gmail.com	5000000000
3	5000000010	10	Mike	Wood	32	27820	30-NOV-21	529-814-7883	mike_sk@gmail.com	5000000000
4	5000000015	5	Mikasa	Ackerman	27	2000	11-DEC-15	239-824-3783	attack_on_m@gmail.com	(null)
5	5000000020	5	Mihai	Moromete	82	11420	22-JUN-90	639-824-3884	salcamul_m@gmail.com	5000000015
6	5000000025	5	Gerald	Smith	41	5720	10-AUG-00	877-888-3443	smith_g@gmail.com	5000000015
7	5000000030	15	Levi	Ackerman	32	12820	10-AUG-15	549-444-3883	levi_a@gmail.com	(null)
8	5000000035	15	Mike	Moris	21	8820.99	30-NOV-21	229-824-3883	mike_moris@yahoo.com	5000000030
9	5000000040	15	Melissa	Monroe	42	9820.99	10-AUG-00	539-824-3773	monroe_m@gmail.com	5000000030
10	5000000045	20	Ion	Inica	44	9820.99	10-AUG-15	544-8222-3883	ionel_ionelule@gmail.com	(null)
11	5000000050	20	Maria	Marioara	32	1820.99	15-OCT-11	533-887-3003	marie_marioara@gmail.com	5000000045
12	5000000055	25	Dumitru	Ion	35	1820.99	30-NOV-21	566-871-3993	dumitru_i@gmail.com	5000000060
13	5000000060	25	Tony	Stark	37	21820.99	11-NOV-09	577-777-7773	jarvis@gmail.com	(null)
14	5000000065	30	Tom	Ford	41	8920.09	11-NOV-09	522-222-2282	tomford@gmail.com	5000000070
15	5000000070	30	Steve	Jobs	44	29820.99	21-NOV-02	534-444-4444	steve_a@gmail.com	(null)
16	5000000075	30	Rick	Morty	18	5820.99	30-NOV-21	599-999-9999	pickle@gmail.com	5000000070
17	5000000080	35	Codin	Dan	25	8820.99	30-NOV-21	599-866-3999	codin_d@gmail.com	5000000085
18	5000000085	35	Silva	Specter	30	23820.99	12-FEB-10	587-787-7778	silva@gmail.com	(null)

Setare lider pentru angajat si echipa

```
UPDATE ECHIPA_ZXC
SET lider = 5000000000
WHERE id_echipa = 10;

UPDATE ECHIPA_ZXC
SET lider = 5000000015
```

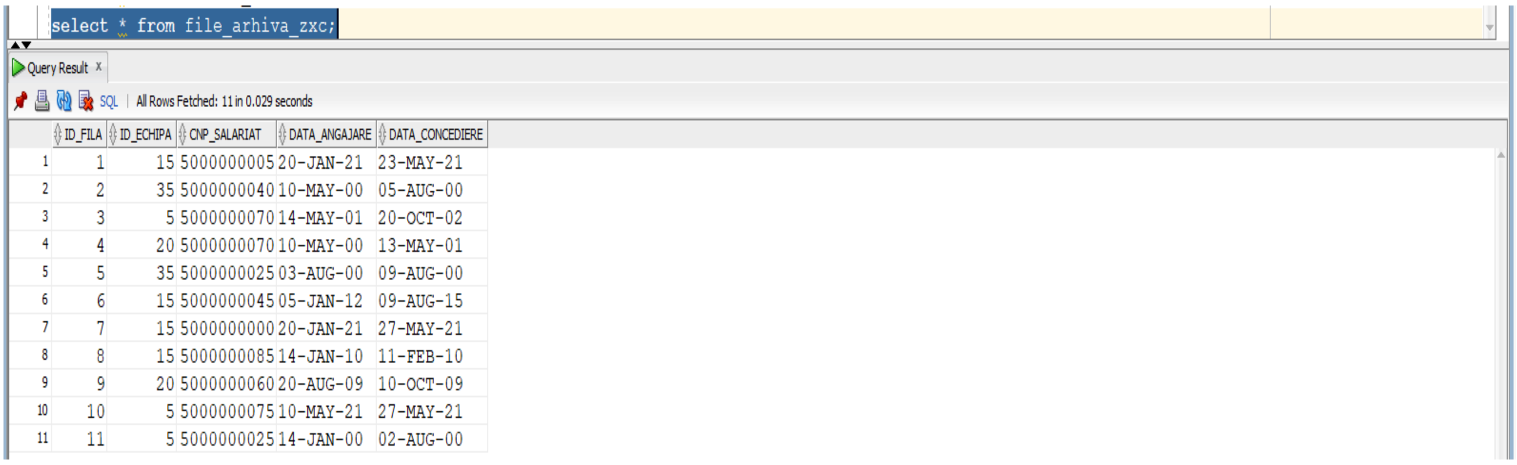
```
WHERE id_echipa = 5;

UPDATE ECHIPA_ZXC
SET lider = 5000000030
WHERE id_echipa = 15;


UPDATE ECHIPA_ZXC
SET lider = 5000000045
WHERE id_echipa = 20;
UPDATE SALARIAT_ZXC
SET lider = (select lider from echipa_zxc where id_echipa = 20)
WHERE salariat_zxc.id_echipa = 20 and cnp_salariat != (select lider from echipa_zxc where id_echipa=20);

UPDATE ECHIPA_ZXC
SET lider = 5000000060
WHERE id_echipa = 25;
UPDATE SALARIAT_ZXC
SET lider = (select lider from echipa_zxc where id_echipa = 25)
WHERE salariat_zxc.id_echipa = 25 and cnp_salariat != (select lider from echipa_zxc where id_echipa=25);

UPDATE ECHIPA_ZXC
SET lider = 5000000070
WHERE id_echipa = 30;
UPDATE SALARIAT_ZXC
SET lider = (select lider from echipa_zxc where id_echipa = 30)
WHERE salariat_zxc.id_echipa = 30 and cnp_salariat != (select lider from echipa_zxc where id_echipa=30);

UPDATE ECHIPA_ZXC
SET lider = 5000000085
WHERE id_echipa = 35;
UPDATE SALARIAT_ZXC
SET lider = (select lider from echipa_zxc where id_echipa = 35)
WHERE salariat_zxc.id_echipa = 35 and cnp_salariat != (select lider from echipa_zxc where id_echipa=35);
```

Modificarile sunt deja vizibile in pozele de mai sus.

FILE-ARHIVA

```
INSERT INTO FILE_ARHIVA_ZXC
VALUES(1,15,5000000005, TO_DATE(’20−01−2021’,’DD−MM−YYYY’), TO_DATE(’23−05−2021’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(2, 35,5000000040, TO_DATE(’10−05−2000’,’DD−MM−YYYY’), TO_DATE(’05−08−2000’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(3, 5,5000000070, TO_DATE(’14−05−2001’,’DD−MM−YYYY’), TO_DATE(’20−10−2002’,’DD−MM−YYYY’))
    ↪ ;

INSERT INTO FILE_ARHIVA_ZXC
VALUES(4, 20,5000000070, TO_DATE(’10−05−2000’,’DD−MM−YYYY’), TO_DATE(’13−05−2001’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(5, 35,5000000025, TO_DATE(’03−08−2000’,’DD−MM−YYYY’), TO_DATE(’09−08−2000’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(6, 15,5000000045, TO_DATE(’05−01−2012’,’DD−MM−YYYY’), TO_DATE(’09−08−2015’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(7, 15,5000000000, TO_DATE(’20−01−2021’,’DD−MM−YYYY’), TO_DATE(’27−05−2021’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(8, 15,5000000085, TO_DATE(’14−01−2010’,’DD−MM−YYYY’), TO_DATE(’11−02−2010’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(9, 20,5000000060, TO_DATE(’20−08−2009’,’DD−MM−YYYY’), TO_DATE(’10−10−2009’,’DD−MM−YYYY’)
    ↪ );

INSERT INTO FILE_ARHIVA_ZXC
VALUES(10, 5,5000000075, TO_DATE(’10−05−2021’,’DD−MM−YYYY’), TO_DATE(’27−05−2021’,’DD−MM−YYYY’)
    ↪ );
```

```
INSERT INTO FILE_ARHIVA_ZXC
VALUES(11, 5,5000000025, TO_DATE('14−01−2000','DD−MM−YYYY'), TO_DATE('02−08−2000','DD−MM−YYYY')
    ↪ );
```



```
select * from file_arhiva_zxc;
```

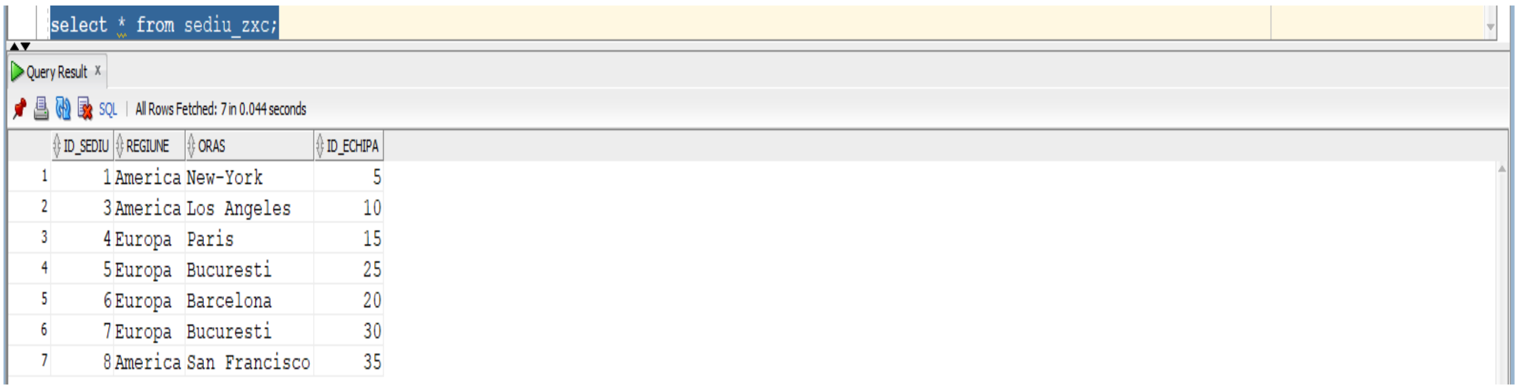
Query Result

SQL | All Rows Fetched: 11 in 0.029 seconds

	ID_FILA	ID_ECHIPA	CNP_SALARIAT	DATA_ANGAJARE	DATA_CONCEDIERE
1	1	15	5000000005	20-JAN-21	23-MAY-21
2	2	35	5000000040	10-MAY-00	05-AUG-00
3	3	5	5000000070	14-MAY-01	20-OCT-02
4	4	20	5000000070	10-MAY-00	13-MAY-01
5	5	35	5000000025	03-AUG-00	09-AUG-00
6	6	15	5000000045	05-JAN-12	09-AUG-15
7	7	15	5000000000	20-JAN-21	27-MAY-21
8	8	15	5000000085	14-JAN-10	11-FEB-10
9	9	20	5000000060	20-AUG-09	10-OCT-09
10	10	5	5000000075	10-MAY-21	27-MAY-21
11	11	5	5000000025	14-JAN-00	02-AUG-00

SEDIU

```
INSERT INTO SEDIU_ZXC
VALUES(1,'America', 'New−York',5);

INSERT INTO SEDIU_ZXC
VALUES(3,'America', 'Los␣Angeles',10);

INSERT INTO SEDIU_ZXC
VALUES(4,'Europa', 'Paris',15);

INSERT INTO SEDIU_ZXC
VALUES(5,'Europa', 'Bucuresti',25);

INSERT INTO SEDIU_ZXC
VALUES(6,'Europa', 'Spania',20);

INSERT INTO SEDIU_ZXC
VALUES(7,'Europa', 'Bucuresti',30);

INSERT INTO SEDIU_ZXC
VALUES(8,'America', 'San␣Francisco',35);
```



```
select * from sediu_zxc;
```

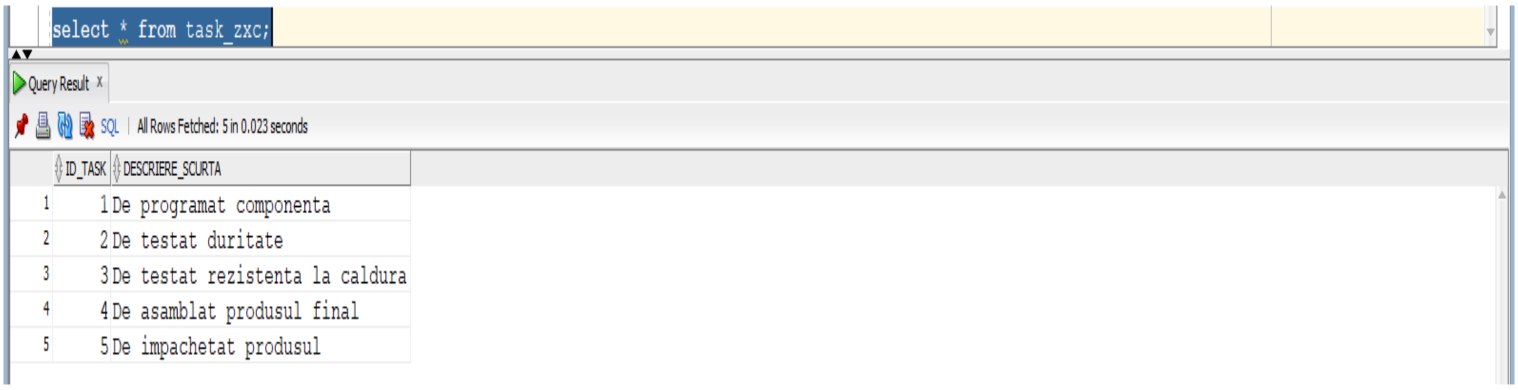
Query Result

SQL | All Rows Fetched: 7 in 0.044 seconds

	ID_SEDIU	REGIUNE	ORAS	ID_ECHIPA
1	1	America	New-York	5
2	3	America	Los Angeles	10
3	4	Europa	Paris	15
4	5	Europa	Bucuresti	25
5	6	Europa	Barcelona	20
6	7	Europa	Bucuresti	30
7	8	America	San Francisco	35

TASK

```
INSERT INTO TASK_ZXC
VALUES(1,'De␣programat␣componenta');

INSERT INTO TASK_ZXC
VALUES(2,'De␣testat␣duritate');

INSERT INTO TASK_ZXC
VALUES(3,'De␣testat␣rezistenta␣la␣caldura');

INSERT INTO TASK_ZXC
VALUES(4,'De␣asamblat␣produsul␣final');

INSERT INTO TASK_ZXC
VALUES(5,'De␣impachetat␣produsul');
```



```
select * from task_zxc;
```

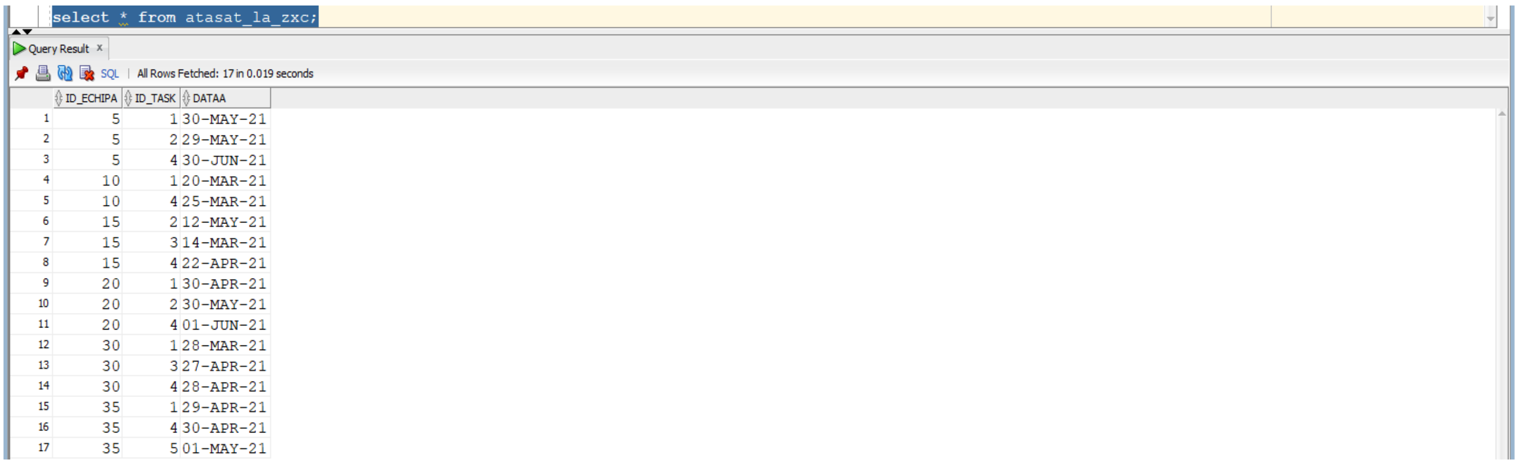
Query Result

All Rows Fetched: 5 in 0.023 seconds

	ID_TASK	DESCRIERE_SCURTA
1	1	De programat componenta
2	2	De testat duritate
3	3	De testat rezistenta la caldura
4	4	De asamblat produsul final
5	5	De impachetat produsul

ATASAT-LA

```
INSERT INTO ATASAT_LA_ZXC
VALUES(5,1,TO_DATE('30–05–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(5,2,TO_DATE('29–05–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(5,4,TO_DATE('30–06–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(10,1,TO_DATE('20–03–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(10,4,TO_DATE('25–03–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(15,2,TO_DATE('12–05–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(15,3,TO_DATE('14–03–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(15,4,TO_DATE('22–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(20,1,TO_DATE('30–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(20,2,TO_DATE('30–05–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(20,4,TO_DATE('1–06–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(30,1,TO_DATE('28–03–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(30,3,TO_DATE('27–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(30,4,TO_DATE('28–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(35,1,TO_DATE('29–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(35,4,TO_DATE('30–04–2021','DD–MM–YYYY'));

INSERT INTO ATASAT_LA_ZXC
VALUES(35,5,TO_DATE('01–05–2021','DD–MM–YYYY'));
```



```
select * from atasat_la_zxc;
```

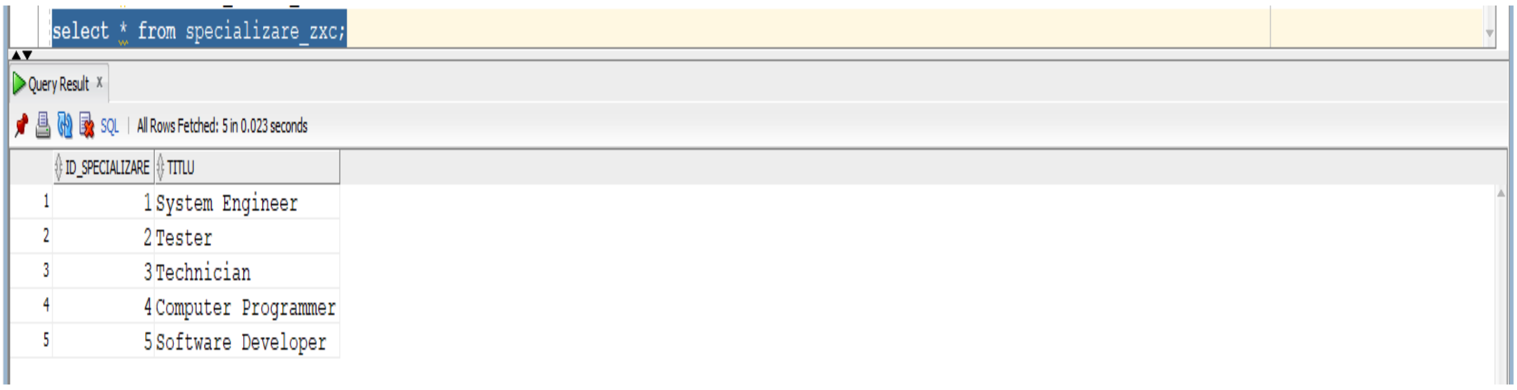
Query Result

SQL | All Rows Fetched: 17 in 0.019 seconds

	ID_ECHIPA	ID_TASK	DATAA
1	5	1	30-MAY-21
2	5	2	29-MAY-21
3	5	4	30-JUN-21
4	10	1	20-MAR-21
5	10	4	25-MAR-21
6	15	2	12-MAY-21
7	15	3	14-MAR-21
8	15	4	22-APR-21
9	20	1	30-APR-21
10	20	2	30-MAY-21
11	20	4	01-JUN-21
12	30	1	28-MAR-21
13	30	3	27-APR-21
14	30	4	28-APR-21
15	35	1	29-APR-21
16	35	4	30-APR-21
17	35	5	01-MAY-21

SPECIALIZARE

```
INSERT INTO SPECIALIZARE_ZXC
VALUES(1,’System_Engineer’);

INSERT INTO SPECIALIZARE_ZXC
VALUES(2,’Tester’);

INSERT INTO SPECIALIZARE_ZXC
VALUES(3,’Technician’);

INSERT INTO SPECIALIZARE_ZXC
VALUES(4,’Computer_Programmer’);

INSERT INTO SPECIALIZARE_ZXC
VALUES(5,’Software_Developer’);
```



```
select * from specializare_zxc;
```

Query Result

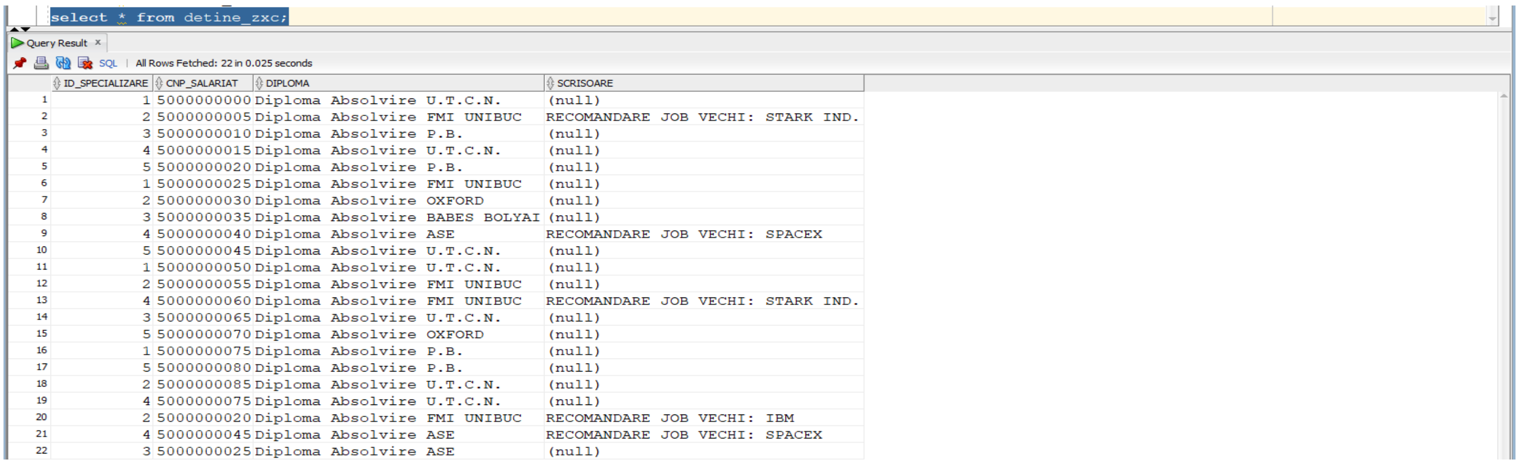
SQL | All Rows Fetched: 5 in 0.023 seconds

	ID_SPECIALIZARE	TITLU
1	1	System Engineer
2	2	Tester
3	3	Technician
4	4	Computer Programmer
5	5	Software Developer

DETINE

```
INSERT INTO DETINE_ZXC
VALUES(1,5000000000, ’Diploma_Absolvire_U.T.C.N.’,NULL);

INSERT INTO DETINE_ZXC
VALUES(2,5000000005, ’Diploma_Absolvire_FMI_UNIBUC’,’RECOMANDARE_JOB_VECHI:_STARK_IND.’);

INSERT INTO DETINE_ZXC
VALUES(3,5000000010, ’Diploma_Absolvire_P.B.’,NULL);

INSERT INTO DETINE_ZXC
VALUES(4,5000000015, ’Diploma_Absolvire_U.T.C.N.’,NULL);

INSERT INTO DETINE_ZXC
VALUES(5,5000000020, ’Diploma_Absolvire_P.B.’,NULL);

INSERT INTO DETINE_ZXC
VALUES(1,5000000025, ’Diploma_Absolvire_FMI_UNIBUC’,NULL);

INSERT INTO DETINE_ZXC
VALUES(2,5000000030, ’Diploma_Absolvire_OXFORD’,NULL);

INSERT INTO DETINE_ZXC
VALUES(3,5000000035, ’Diploma_Absolvire_BABES_BOLYAI’,NULL);

INSERT INTO DETINE_ZXC
VALUES(4,5000000040, ’Diploma_Absolvire_ASE’,’RECOMANDARE_JOB_VECHI:_SPACEX’);

INSERT INTO DETINE_ZXC
VALUES(5,5000000045, ’Diploma_Absolvire_U.T.C.N.’,NULL);
```

```
INSERT INTO DETINE_ZXC
VALUES(1,5000000050, 'Diploma␣Absolvire␣U.T.C.N.',NULL);

INSERT INTO DETINE_ZXC
VALUES(2,5000000055, 'Diploma␣Absolvire␣FMI␣UNIBUC',NULL);

INSERT INTO DETINE_ZXC
VALUES(4,5000000060, 'Diploma␣Absolvire␣FMI␣UNIBUC','RECOMANDARE␣JOB␣VECHI:␣STARK␣IND.');

INSERT INTO DETINE_ZXC
VALUES(3,5000000065, 'Diploma␣Absolvire␣U.T.C.N.',NULL);

INSERT INTO DETINE_ZXC
VALUES(5,5000000070, 'Diploma␣Absolvire␣OXFORD',NULL);

INSERT INTO DETINE_ZXC
VALUES(1,5000000075, 'Diploma␣Absolvire␣P.B.',NULL);

INSERT INTO DETINE_ZXC
VALUES(5,5000000080, 'Diploma␣Absolvire␣P.B.',NULL);

INSERT INTO DETINE_ZXC
VALUES(2,5000000085, 'Diploma␣Absolvire␣U.T.C.N.',NULL);

INSERT INTO DETINE_ZXC
VALUES(4,5000000075, 'Diploma␣Absolvire␣U.T.C.N.',NULL);

INSERT INTO DETINE_ZXC
VALUES(2,5000000020, 'Diploma␣Absolvire␣FMI␣UNIBUC','RECOMANDARE␣JOB␣VECHI:␣IBM');

INSERT INTO DETINE_ZXC
VALUES(4,5000000045, 'Diploma␣Absolvire␣ASE','RECOMANDARE␣JOB␣VECHI:␣SPACEX');

INSERT INTO DETINE_ZXC
VALUES(3,5000000025, 'Diploma␣Absolvire␣ASE',NULL);
```



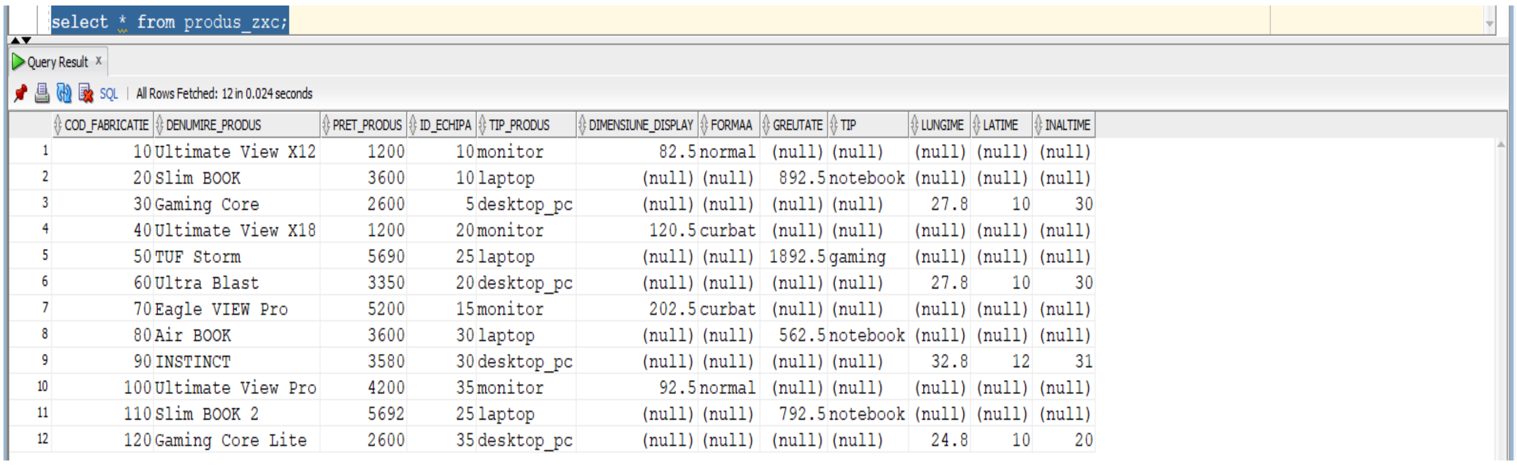
```
select * from detine_zxc;
```

	ID_SPECIALIZARE	CNP_SALARIAT	DIPLOMA	SCRISOARE
1	1	5000000000	Diploma Absolvire U.T.C.N.	(null)
2	2	5000000005	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: STARK IND.
3	3	5000000010	Diploma Absolvire P.B.	(null)
4	4	5000000015	Diploma Absolvire U.T.C.N.	(null)
5	5	5000000020	Diploma Absolvire P.B.	(null)
6	1	5000000025	Diploma Absolvire FMI UNIBUC	(null)
7	2	5000000030	Diploma Absolvire OXFORD	(null)
8	3	5000000035	Diploma Absolvire BABES BOLYAI	(null)
9	4	5000000040	Diploma Absolvire ASE	RECOMANDARE JOB VECHI: SPACEX
10	5	5000000045	Diploma Absolvire U.T.C.N.	(null)
11	1	5000000050	Diploma Absolvire U.T.C.N.	(null)
12	2	5000000055	Diploma Absolvire FMI UNIBUC	(null)
13	4	5000000060	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: STARK IND.
14	3	5000000065	Diploma Absolvire U.T.C.N.	(null)
15	5	5000000070	Diploma Absolvire OXFORD	(null)
16	1	5000000075	Diploma Absolvire P.B.	(null)
17	5	5000000080	Diploma Absolvire P.B.	(null)
18	2	5000000085	Diploma Absolvire U.T.C.N.	(null)
19	4	5000000075	Diploma Absolvire U.T.C.N.	(null)
20	2	5000000020	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: IBM
21	4	5000000045	Diploma Absolvire ASE	RECOMANDARE JOB VECHI: SPACEX
22	3	5000000025	Diploma Absolvire ASE	(null)

PRODUS

```
--Secventa pentru id produs --
CREATE SEQUENCE seq_produs_zxc
INCREMENT by 10
START WITH 10
MAXVALUE 10000
NOCYCLE;


INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Ultimate␣View␣X12',1200.00,10,'monitor',82.5,'normal',NULL,NULL,NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Slim␣BOOK',3600.00,10,'laptop',NULL,NULL,892.5,'notebook',NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Gaming␣Core',2600.00,5,'desktop␣pc',NULL,NULL,NULL,NULL,27.8,10,30);

INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Ultimate␣View␣X18',1200.00,20,'monitor',120.5,'curbat',NULL,NULL,NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'TUF␣Storm',5690.00,25,'laptop',NULL,NULL,1892.5,'gaming',NULL,NULL,NULL)
    ↪ ;
```

```
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Ultra_Blast',3350.00,20,'desktop_pc',NULL,NULL,NULL,NULL,27.8,10,30);

INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Eagle_VIEW_Pro',5200.00,15,'monitor',202.5,'curbat',NULL,NULL,NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Air_BOOK',3600.00,30,'laptop',NULL,NULL,562.5,'notebook',NULL,NULL,NULL)
    ↪ ;
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'INSTINCT',3580.00,30,'desktop_pc',NULL,NULL,NULL,NULL,32.8,12,31);

INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Ultimate_View_Pro',4200.00,35,'monitor',92.5,'normal',NULL,NULL,NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Slim_BOOK_2',5692.00,25,'laptop',NULL,NULL,792.5,'notebook',NULL,NULL,
    ↪ NULL);
INSERT INTO PRODUS_ZXC
VALUES(seq_produs_zxc.nextval,'Gaming_Core_Lite',2600.00,35,'desktop_pc',NULL,NULL,NULL,NULL,24.8,10,20);
```

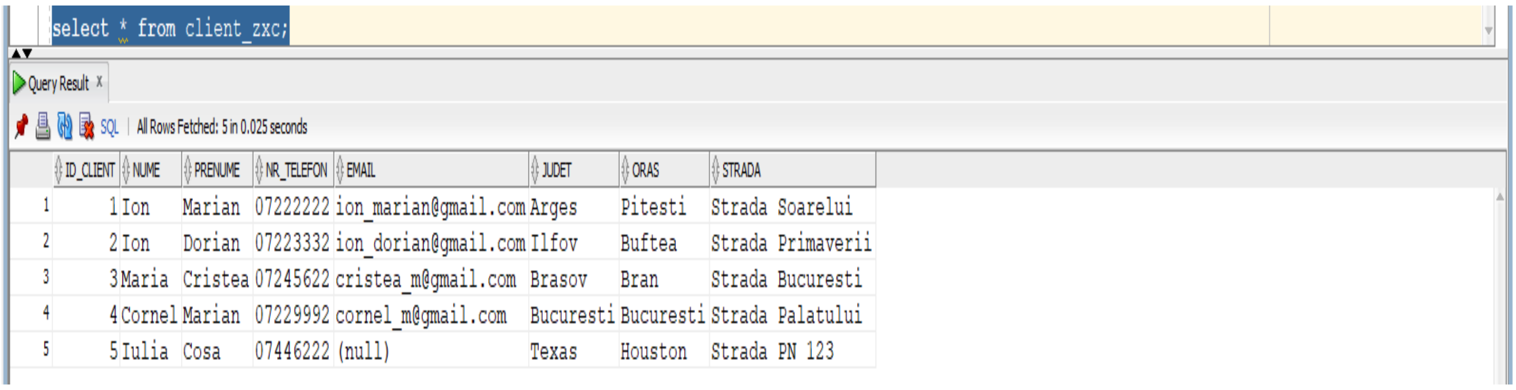


```
select * from produs_zxc;
```

	COD_FABRICATIE	DENUMIRE_PRODUS	PRET_PRODUS	ID_ECHIPA	TIP_PRODUS	DIMENSIUNE_DISPLAY	FORMAA	GREUTATE	TIP	LUNGIME	LATIME	INALTIME
1	10	Ultimate View X12	1200	10	monitor	82.5	normal	(null)	(null)	(null)	(null)	(null)
2	20	Slim BOOK	3600	10	laptop	(null)	(null)	892.5	notebook	(null)	(null)	(null)
3	30	Gaming Core	2600	5	desktop_pc	(null)	(null)	(null)	(null)	27.8	10	30
4	40	Ultimate View X18	1200	20	monitor	120.5	curbat	(null)	(null)	(null)	(null)	(null)
5	50	TUF Storm	5690	25	laptop	(null)	(null)	1892.5	gaming	(null)	(null)	(null)
6	60	Ultra Blast	3350	20	desktop_pc	(null)	(null)	(null)	(null)	27.8	10	30
7	70	Eagle VIEW Pro	5200	15	monitor	202.5	curbat	(null)	(null)	(null)	(null)	(null)
8	80	Air BOOK	3600	30	laptop	(null)	(null)	562.5	notebook	(null)	(null)	(null)
9	90	INSTINCT	3580	30	desktop_pc	(null)	(null)	(null)	(null)	32.8	12	31
10	100	Ultimate View Pro	4200	35	monitor	92.5	normal	(null)	(null)	(null)	(null)	(null)
11	110	Slim BOOK 2	5692	25	laptop	(null)	(null)	792.5	notebook	(null)	(null)	(null)
12	120	Gaming Core Lite	2600	35	desktop_pc	(null)	(null)	(null)	(null)	24.8	10	20

CLIENT

```
INSERT INTO CLIENT_ZXC
VALUES(1, 'Ion','Marian', '07222222', 'ion_marian@gmail.com','Arges', 'Pitesti', 'Strada_Soarelui');

INSERT INTO CLIENT_ZXC
VALUES(2, 'Ion','Dorian', '07223332', 'ion_dorian@gmail.com','Ilfov', 'Buftea', 'Strada_Primaverii');

INSERT INTO CLIENT_ZXC
VALUES(3, 'Maria','Cristea', '07245622', 'cristea_m@gmail.com','Brasov', 'Bran', 'Strada_Bucuresti');

INSERT INTO CLIENT_ZXC
VALUES(4, 'Cornel','Marian', '07229992', 'cornel_m@gmail.com','Bucuresti', 'Bucuresti', 'Strada_Palatului');

INSERT INTO CLIENT_ZXC
VALUES(5, 'Iulia','Cosa', '07446222', NULL,'Texas', 'Houston', 'Strada_PN_123');
```



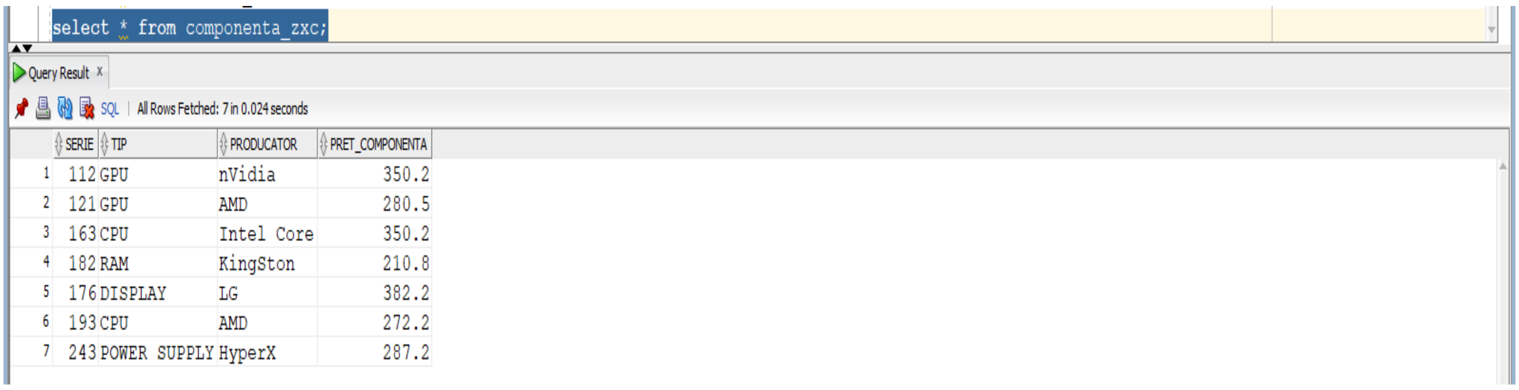
```
select * from client_zxc;
```

	ID_CLIENT	NUME	PRENUME	NR_TELEFON	EMAIL	JUDET	ORAS	STRADA
1	1	Ion	Marian	07222222	ion_marian@gmail.com	Arges	Pitesti	Strada Soarelui
2	2	Ion	Dorian	07223332	ion_dorian@gmail.com	Ilfov	Buftea	Strada Primaverii
3	3	Maria	Cristea	07245622	cristea_m@gmail.com	Brasov	Bran	Strada Bucuresti
4	4	Cornel	Marian	07229992	cornel_m@gmail.com	Bucuresti	Bucuresti	Strada Palatului
5	5	Iulia	Cosa	07446222	(null)	Texas	Houston	Strada PN 123

COMPONENTA

```
INSERT INTO COMPONENTA_ZXC
VALUES(112,'GPU', 'nVidia', 350.20);

INSERT INTO COMPONENTA_ZXC
VALUES(121,'GPU', 'AMD', 280.50);

INSERT INTO COMPONENTA_ZXC
VALUES(163,'CPU', 'Intel_Core', 350.20);
```

```
INSERT INTO COMPONENTA_ZXC
VALUES(182,'RAM', 'KingSton', 210.80);

INSERT INTO COMPONENTA_ZXC
VALUES(176,'DISPLAY', 'LG', 382.20);

INSERT INTO COMPONENTA_ZXC
VALUES(193,'CPU', 'AMD', 272.20);

INSERT INTO COMPONENTA_ZXC
VALUES(243,'POWER_SUPPLY', 'HyperX', 287.20);
```



```
select * from componenta_zxc;
```

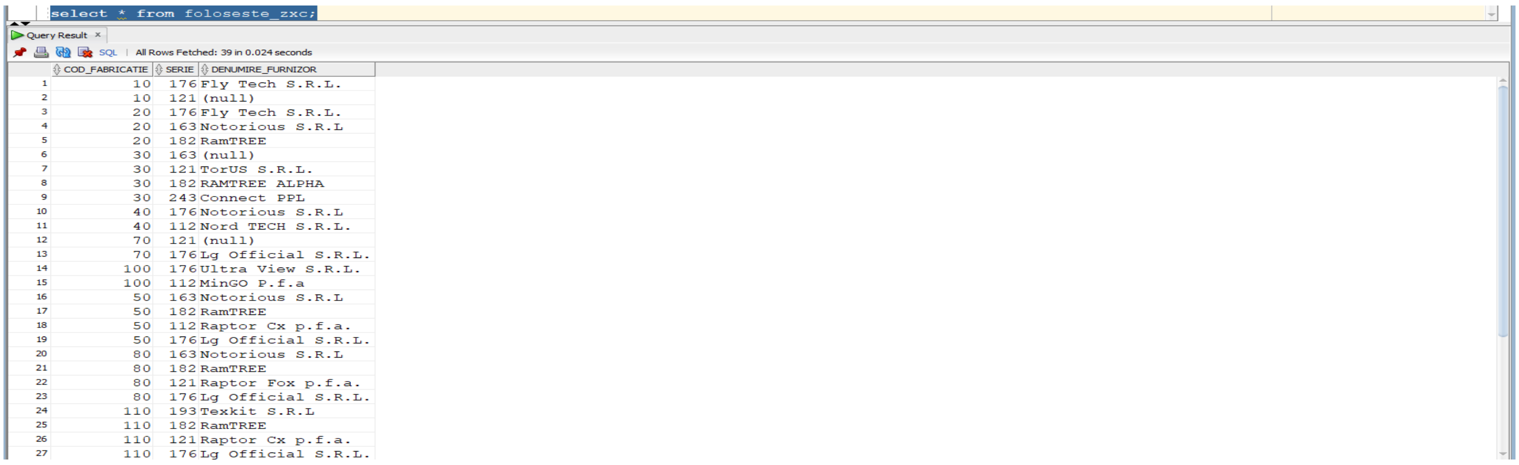
All Rows Fetched: 7 in 0.024 seconds

	SERIE	TIP	PRODUCATOR	PRET_COMPONENTA
1	112	GPU	nVidia	350.2
2	121	GPU	AMD	280.5
3	163	CPU	Intel Core	350.2
4	182	RAM	KingSton	210.8
5	176	DISPLAY	LG	382.2
6	193	CPU	AMD	272.2
7	243	POWER SUPPLY	HyperX	287.2

FOLOSESTE

```
INSERT INTO FOLOSESTE_ZXC
VALUES(10,176,'Fly_Tech_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(10,121,NULL);
INSERT INTO FOLOSESTE_ZXC
VALUES(20,176,'Fly_Tech_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(20,163,'Notorious_S.R.L');
INSERT INTO FOLOSESTE_ZXC
VALUES(20,182,'RamTREE');

INSERT INTO FOLOSESTE_ZXC
VALUES(30,163,NULL);
INSERT INTO FOLOSESTE_ZXC
VALUES(30,121,'TorUS_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(30,182,'RAMTREE_ALPHA');
INSERT INTO FOLOSESTE_ZXC
VALUES(30,243,'Connect_PPL');

INSERT INTO FOLOSESTE_ZXC
VALUES(40,176,'Notorious_S.R.L');
INSERT INTO FOLOSESTE_ZXC
VALUES(40,112,'Nord_TECH_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(70,121,NULL);
INSERT INTO FOLOSESTE_ZXC
VALUES(70,176,'Lg_Official_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(100,176,'Ultra_View_S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(100,112,'MinGO_P.f.a');

INSERT INTO FOLOSESTE_ZXC
VALUES(50,163,'Notorious_S.R.L');
INSERT INTO FOLOSESTE_ZXC
VALUES(50,182,'RamTREE');
INSERT INTO FOLOSESTE_ZXC
VALUES(50,112,'Raptor_Cx_p.f.a.');
INSERT INTO FOLOSESTE_ZXC
VALUES(50,176,'Lg_Official_S.R.L.');

INSERT INTO FOLOSESTE_ZXC
VALUES(80,163,'Notorious_S.R.L');
INSERT INTO FOLOSESTE_ZXC
VALUES(80,182,'RamTREE');
INSERT INTO FOLOSESTE_ZXC
VALUES(80,121,'Raptor_Fox_p.f.a.');
INSERT INTO FOLOSESTE_ZXC
```

```
VALUES(80,176,'Lg␣Official␣S.R.L.');

INSERT INTO FOLOSESTE_ZXC
VALUES(110,193,'Texkit␣S.R.L');
INSERT INTO FOLOSESTE_ZXC
VALUES(110,182,'RamTREE');
INSERT INTO FOLOSESTE_ZXC
VALUES(110,121,'Raptor␣Cx␣p.f.a.');
INSERT INTO FOLOSESTE_ZXC
VALUES(110,176,'Lg␣Official␣S.R.L.');


INSERT INTO FOLOSESTE_ZXC
VALUES(60,182,'RamTREE');
INSERT INTO FOLOSESTE_ZXC
VALUES(60,163,'Bionic␣Soft␣Inc.');
INSERT INTO FOLOSESTE_ZXC
VALUES(60,121,'Lig␣X␣Co.');
INSERT INTO FOLOSESTE_ZXC
VALUES(60,243,'Connect␣PPL');

INSERT INTO FOLOSESTE_ZXC
VALUES(90,193,'Ultra␣Soft␣Co.');
INSERT INTO FOLOSESTE_ZXC
VALUES(90,121,'TorUS␣S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(90,182,'RAMTREE␣ALPHA');
INSERT INTO FOLOSESTE_ZXC
VALUES(90,243,'Connect␣PPL');

INSERT INTO FOLOSESTE_ZXC
VALUES(120,163,'Bionic␣Soft␣Inc.');
INSERT INTO FOLOSESTE_ZXC
VALUES(120,121,'TorUS␣S.R.L.');
INSERT INTO FOLOSESTE_ZXC
VALUES(120,182,'Ram␣TERRA');
INSERT INTO FOLOSESTE_ZXC
VALUES(120,243,'Connect␣PPL');
```



```
select * from foloseste_zxc;
```

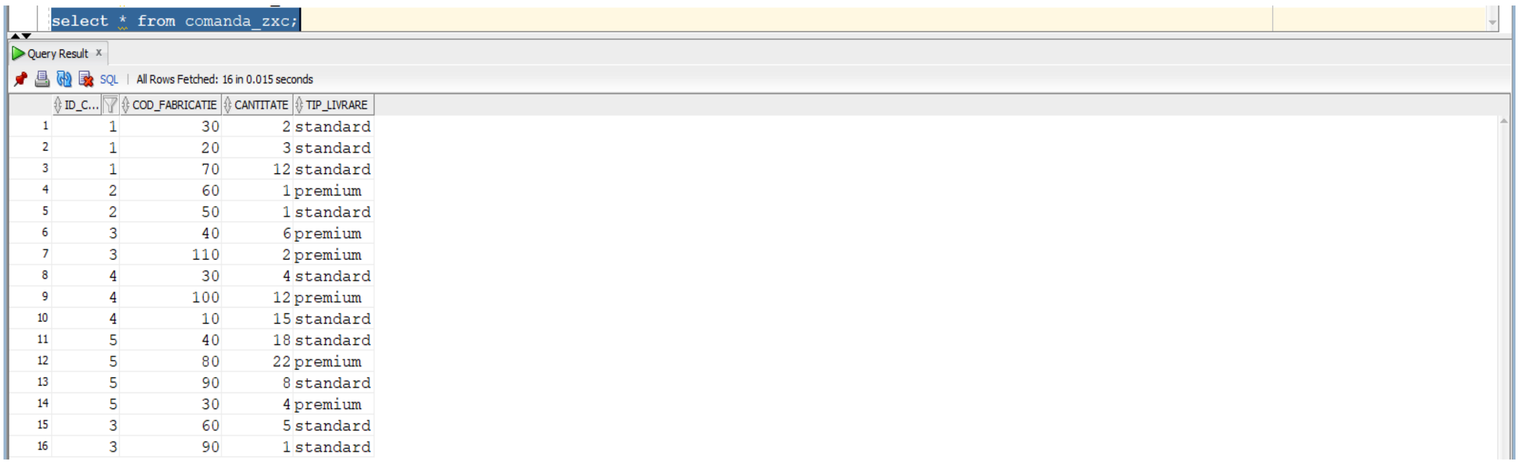
All Rows Fetched: 39 in 0.024 seconds

	COD_FABRICATIE	SERIE	DENUMIRE_FURNIZOR
1	10	176	Fly Tech S.R.L.
2	10	121	(null)
3	20	176	Fly Tech S.R.L.
4	20	163	Notorious S.R.L
5	20	182	RamTREE
6	30	163	(null)
7	30	121	TorUS S.R.L.
8	30	182	RAMTREE ALPHA
9	30	243	Connect PPL
10	40	176	Notorious S.R.L
11	40	112	Nord TECH S.R.L.
12	70	121	(null)
13	70	176	Lg Official S.R.L.
14	100	176	Ultra View S.R.L.
15	100	112	MinGO P.f.a
16	50	163	Notorious S.R.L
17	50	182	RamTREE
18	50	112	Raptor Cx p.f.a.
19	50	176	Lg Official S.R.L.
20	80	163	Notorious S.R.L
21	80	182	RamTREE
22	80	121	Raptor Fox p.f.a.
23	80	176	Lg Official S.R.L.
24	110	193	Texkit S.R.L
25	110	182	RamTREE
26	110	121	Raptor Cx p.f.a.
27	110	176	Lg Official S.R.L.

COMANDA

```
INSERT INTO COMANDA_ZXC
VALUES(1,30,2,'standard');

INSERT INTO COMANDA_ZXC
VALUES(1,20,3,'standard');

INSERT INTO COMANDA_ZXC
VALUES(1,70,12,'standard');

INSERT INTO COMANDA_ZXC
VALUES(2,60,1,'premium');

INSERT INTO COMANDA_ZXC
VALUES(2,50,1,'standard');

INSERT INTO COMANDA_ZXC
VALUES(3,40,6,'premium');

INSERT INTO COMANDA_ZXC
VALUES(3,110,2,'premium');
```

```
INSERT INTO COMANDA_ZXC
VALUES(4,30,4,’standard’);

INSERT INTO COMANDA_ZXC
VALUES(4,100,12,’premium’);

INSERT INTO COMANDA_ZXC
VALUES(4,10,15,’standard’);

INSERT INTO COMANDA_ZXC
VALUES(5,40,18,’standard’);

INSERT INTO COMANDA_ZXC
VALUES(5,80,22,’premium’);

INSERT INTO COMANDA_ZXC
VALUES(5,90,8,’standard’);

INSERT INTO COMANDA_ZXC
VALUES(5,30,4,’premium’);

INSERT INTO COMANDA_ZXC
VALUES(3,90,1,’standard’);

INSERT INTO COMANDA_ZXC
VALUES(3,60,5,’standard’);
```



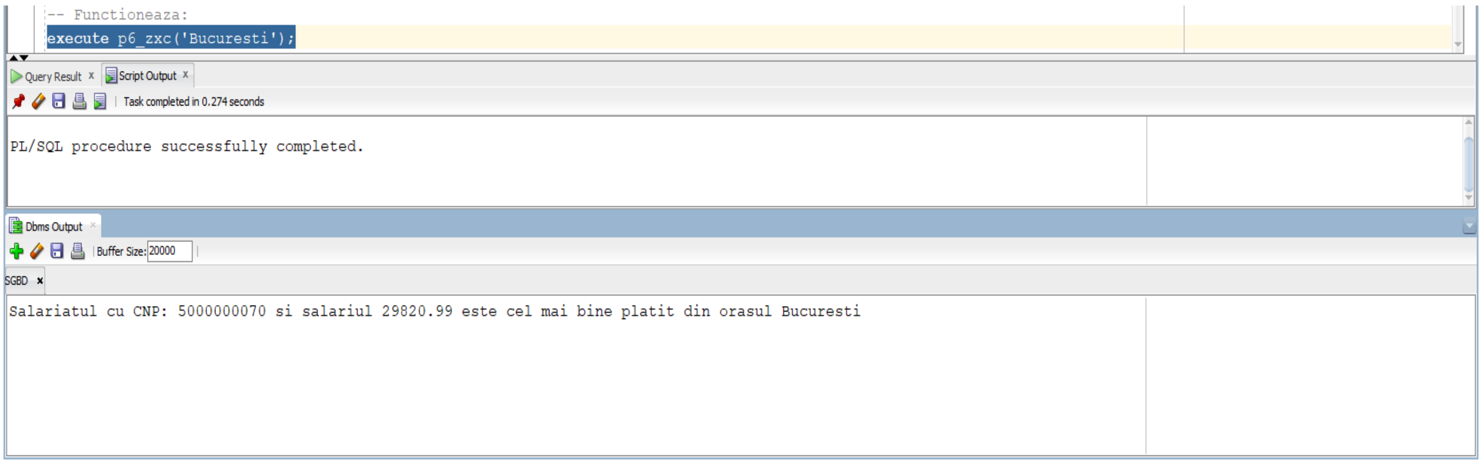
```
select * from comanda_zxc;
```

All Rows Fetched: 16 in 0.015 seconds

	ID_C...	COD_FABRICATIE	CANTITATE	TIP_LIVRARE
1	1	30	2	standard
2	1	20	3	standard
3	1	70	12	standard
4	2	60	1	premium
5	2	50	1	standard
6	3	40	6	premium
7	3	110	2	premium
8	4	30	4	standard
9	4	100	12	premium
10	4	10	15	standard
11	5	40	18	standard
12	5	80	22	premium
13	5	90	8	standard
14	5	30	4	premium
15	3	60	5	standard
16	3	90	1	standard

6 Punctul 6:

```
/*
Formulare: Creati o procedura stocata care determina cel mai bine platit salariat al firmei
care lucreaza intr-una din echipele ce au sediul intr-un oras specificat ca parametru..
Daca sunt mai multi angajati cu acelasi salariu si care lucreaza in aceeasi locatie,
se va selecta cel mai tanar dintre acestia.
Folositi 2 tipuri de colectii studiate.
*/
CREATE OR REPLACE PROCEDURE p6_zxc(v_oras sediu_zxc.oras%type)
IS
    type echipe_nested is table of sediu_zxc.id_echipa%type;
    v_echipa echipe_nested;
    type salariat_index is table of salariat_zxc%rowtype index by pls_integer;
    v_salariat salariat_index;
    v_max number(10,2):=0;
    v_cnp number(10,0);
    v_age number(3,0):=999;
    v_check number;
    GO_TO_OTHERS EXCEPTION;
BEGIN
    if REGEXP_LIKE(v_oras,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;

    select count(*) into v_check
    from sediu_zxc
    where upper(oras) = upper(v_oras);
    if v_check = 0 then
        RAISE NO_DATA_FOUND;
    end if;

    select id_echipa
    bulk collect into v_echipa
    from sediu_zxc
    where upper(oras) = upper(v_oras);

    for i in v_echipa.FIRST..v_echipa.LAST loop
        select *
        bulk collect into v_salariat
        from salariat_zxc
        where id_echipa = v_echipa(i);
        for j in v_salariat.FIRST..v_salariat.LAST loop
            if v_salariat(j).salariu > v_max then
                v_age:= v_salariat(j).varsta;
                v_max:= v_salariat(j).salariu;
                v_cnp:= v_salariat(j).cnp_salariat;
            else
                if v_salariat(j).salariu = v_max and v_salariat(j).varsta < v_age then
                    v_age:= v_salariat(j).varsta;
                    v_max:= v_salariat(j).salariu;
                    v_cnp:= v_salariat(j).cnp_salariat;
                end if;
            end if;
        end loop;
        v_salariat.DELETE;
    end loop;
    v_echipa.DELETE;

    dbms_output.put_line('Salariatul␣cu␣CNP:␣'||v_cnp||'␣si␣salariul␣'||v_max||'␣este␣cel␣mai␣bine␣platit␣din␣orasul␣'||v_oras
        ↪ );
EXCEPTION
    when NO_DATA_FOUND then
    dbms_output.put_line('Nicio␣echipa␣nu␣are␣sediul␣in␣Orasul␣dat.');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare');

END p6_zxc;
/
-- Functioneaza:
execute p6_zxc('Bucuresti');

-- No_Data_Found:
execute p6_zxc('Barc');

-- When Others / go_to_Others:
execute p6_zxc(123);
```



```
-- Functioneaza:
execute p6_zxc('Bucuresti');
```

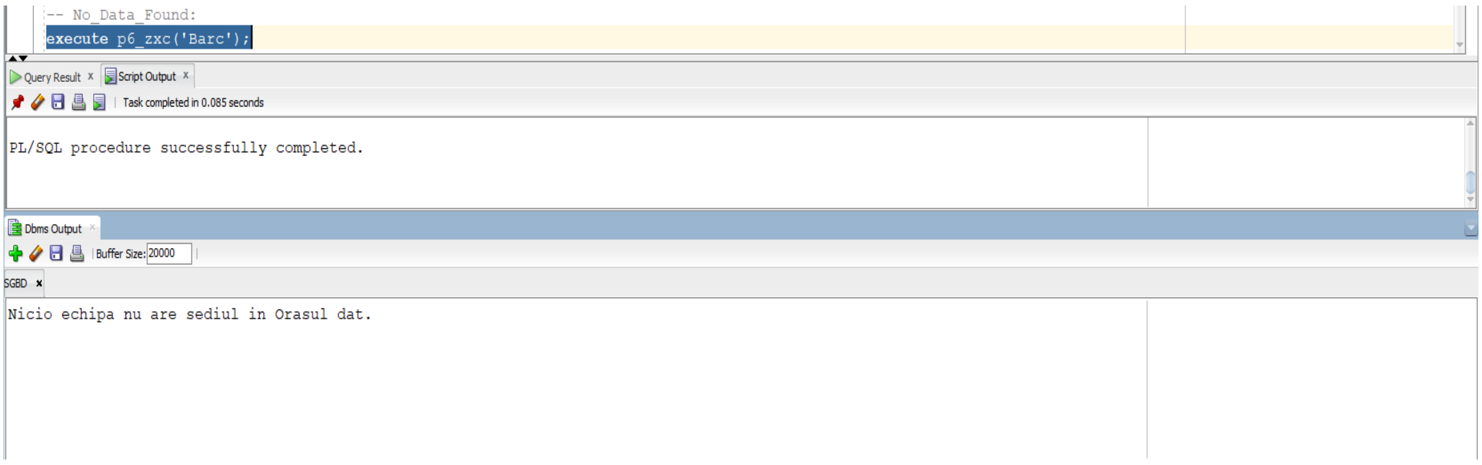
Task completed in 0.274 seconds

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

Buffer Size: 20000

```
Salariatul cu CNP: 5000000070 si salariul 29820.99 este cel mai bine platit din orasul Bucuresti
```

Subprogramul Functioneaza



```
-- No_Data_Found:
execute p6_zxc('Barc');
```

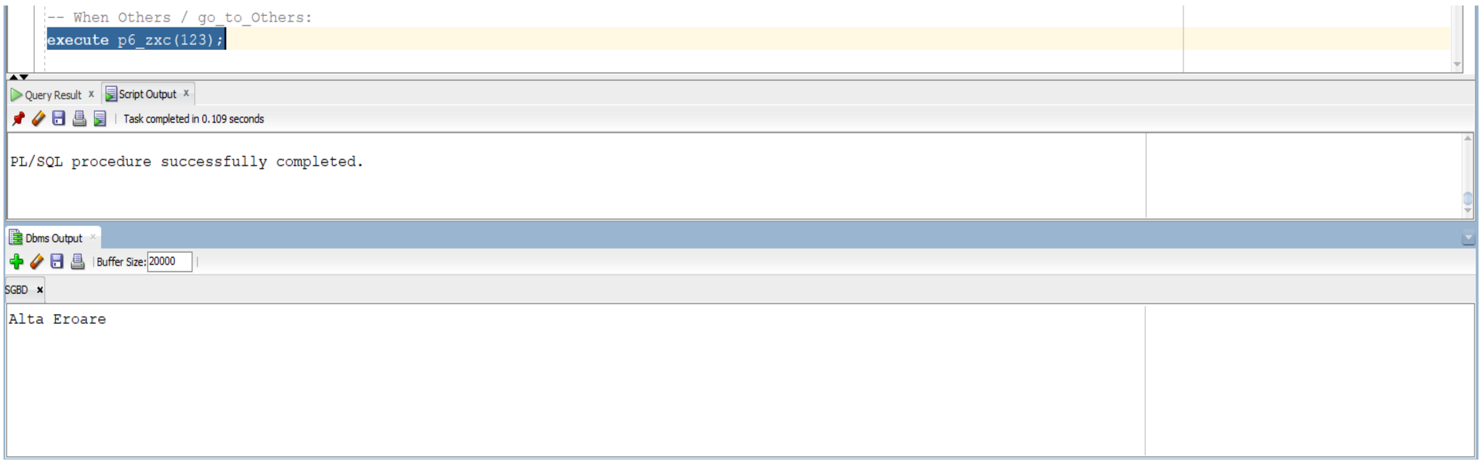
Task completed in 0.085 seconds

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

Buffer Size: 20000

```
Nicio echipa nu are sediul in Orasul dat.
```

Exceptia No-Data-Found



```
-- When Others / go_to_Others:
execute p6_zxc(123);
```

Task completed in 0.109 seconds

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

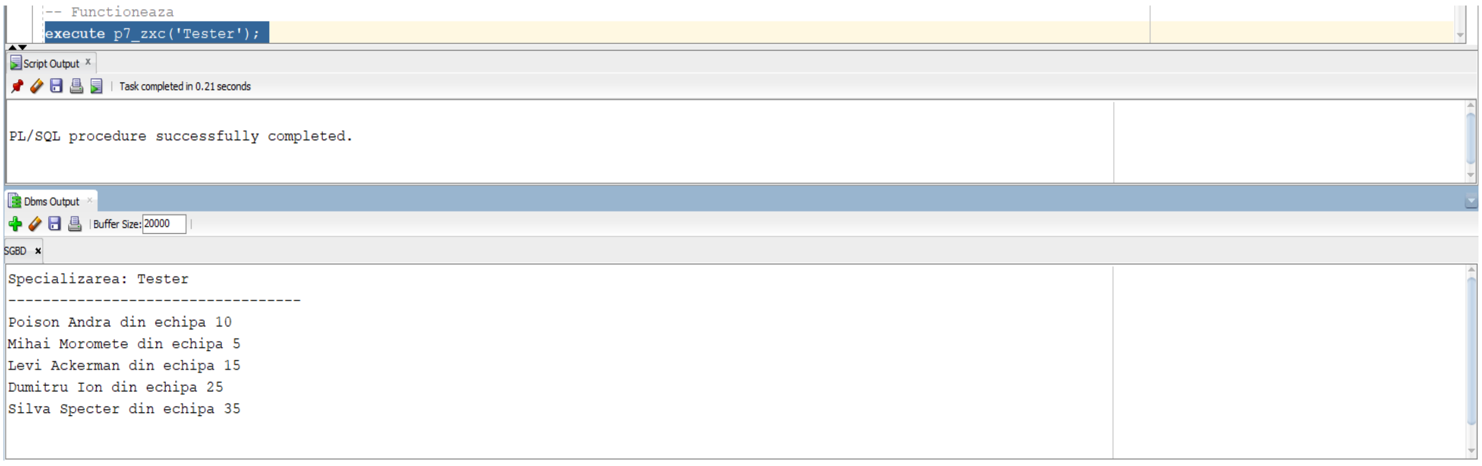
Buffer Size: 20000

```
Alta Eroare
```

Exceptia Others

7 Punctul 7:

```
/*
Formulare: Folosind o procedura stocata, afisati pentru o specializare transmisa
ca parametru detaliile angajatilor care o detin (nume, prenume si echipa din care provin).
Sa se trateze cazurile in care:
-specializarea transmisa nu este corecta / nu exista
-sunt prea multe specializari cu acelasi nume
-nici-un angajat nu detine specializarea respectiva.
Rezolvati cerinta cu ajutorul unui tip de cursor studiat.
*/
CREATE OR REPLACE PROCEDURE p7_zxc(v_specializare specializare_zxc.titlu%type)
IS

    CURSOR specializari (specializare_angajat specializare_zxc.id_specializare%TYPE) IS
    select s.nume, s.prenume, s.id_echipa
    from salariat_zxc s, detine_zxc d
    where s.cnp_salariat = d.cnp_salariat and d.id_specializare = specializare_angajat;

    v_spec specializare_zxc.id_specializare%type;
    v_nume salariat_zxc.nume%type;
    v_prenume salariat_zxc.prenume%type;
    v_echipa salariat_zxc.id_echipa%type;
    GO_TO_OTHERS EXCEPTION;
BEGIN
    if REGEXP_LIKE(v_specializare,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;

    select id_specializare
    into v_spec
    from specializare_zxc
    where lower(titlu) = lower(v_specializare);

    dbms_output.put_line('Specializarea:␣' || v_specializare);
    dbms_output.put_line('-----------------------------------');

    open specializari(v_spec);

    loop
        fetch specializari into v_nume, v_prenume, v_echipa;
        exit when specializari%NOTFOUND;
        dbms_output.put_line(v_nume||'␣'||v_prenume||'␣din␣echipa␣'||v_echipa);
    end loop;

    if specializari%rowcount = 0 then
        dbms_output.put_line('Nimeni␣nu␣are␣aceasta␣specializare');
    end if;
    close specializari;
    dbms_output.new_line;
EXCEPTION
    when TOO_MANY_ROWS then
    dbms_output.put_line('Prea␣multe␣specializari␣cu␣acelasi␣nume');

    when NO_DATA_FOUND then
    dbms_output.put_line('Nu␣exista␣specializarea␣specificata');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare');
END p7_zxc;
/
-- Functioneaza
execute p7_zxc('Tester');

-- No data Found
execute p7_zxc('Ceva');

-- Too Many Rows
insert into specializare_zxc
values(6,'Tester');
execute p7_zxc('Tester');
delete from specializare_zxc
where id_specializare = 6;

-- Others / go_to_Others:
execute p7_zxc(123);
```

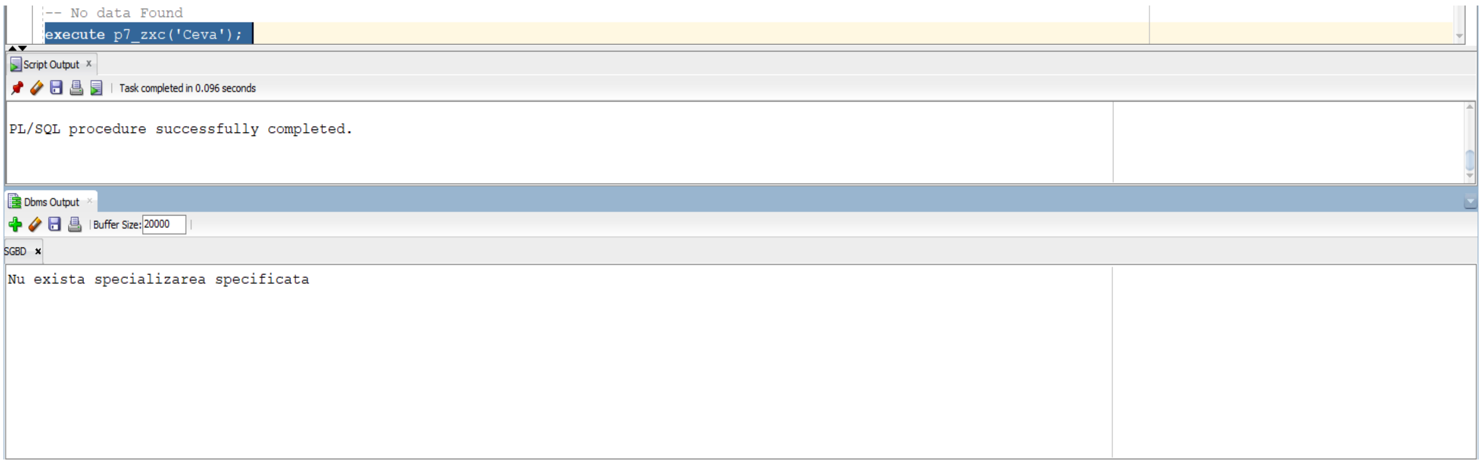


```
-- Functioneaza
execute p7_zxc('Tester');
```

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

```
Specializarea: Tester
----------------------------------
Poison Andra din echipa 10
Mihai Moromete din echipa 5
Levi Ackerman din echipa 15
Dumitru Ion din echipa 25
Silva Specter din echipa 35
```

Subprogramul Functioneaza

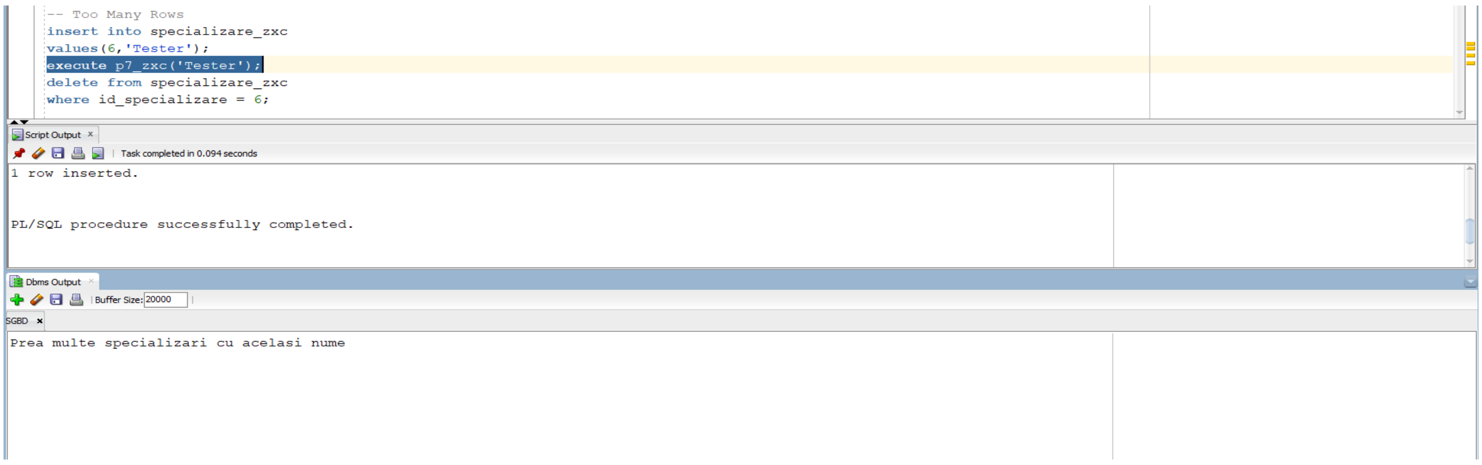


```
-- No data Found
execute p7_zxc('Ceva');
```

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

```
Nu exista specializarea specificata
```

Exceptia No-Data-Found

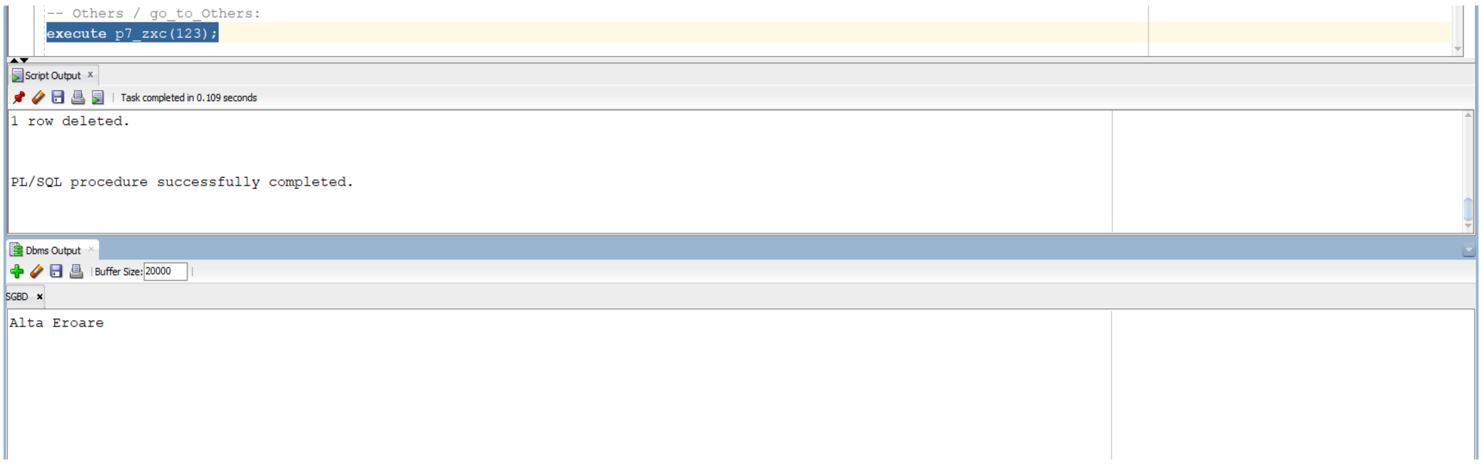


```
-- Too Many Rows
insert into specializare_zxc
values(6,'Tester');
execute p7_zxc('Tester');
delete from specializare_zxc
where id_specializare = 6;
```

```
1 row inserted.


PL/SQL procedure successfully completed.
```

```
Prea multe specializari cu acelasi nume
```

Exceptia Too-Many-Rows



```
-- Others / go_to_Others:
execute p7_zxc(123);
```

```
1 row deleted.


PL/SQL procedure successfully completed.
```

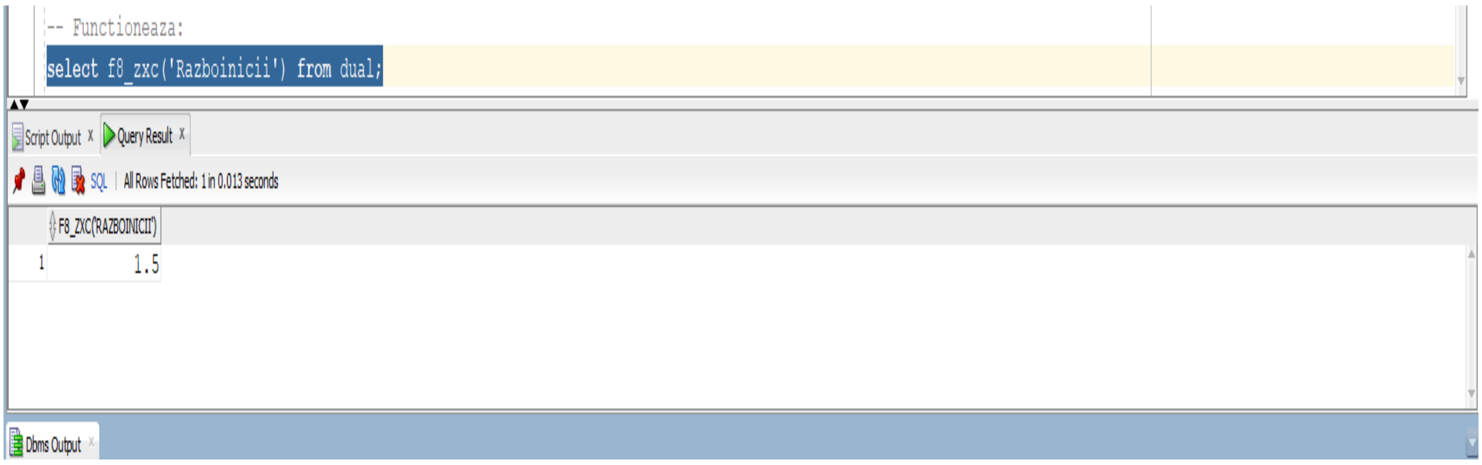
```
Alta Eroare
```

Exceptia Others

8 Punctul 8:

```
/*
Formulare: In cadrul acestei companii, unele echipe folosesc in asamblarea unui produs componente de pe piata neagra
(componente ce nu au un furnizor specificat).
Intrucat nu vrem ca reputatia companiei sa fie patata, se intentioneaza sa se calculeze o penalizare
echipei care este prinsa ca foloseste astfel de componente, penalizare ce consta in scaderea reputatiei echipei
cu 1.5*nr de componente de pe piata neagra folosite pentru toate tipurile de produse create.
Intrucat unele echipe au deja o reputatie foarte scazuta, creati un subprogram stocat de tip functie
care sa determine daca se pot calcula penalizari asupra reputatie unei echipe, si care va fi numarul
de puncte de penalizare in caz afirmativ.
Tratati cazul in care reputatia echipei este prea mica pentru a mai fi penalizata, dar si alte erori ce pot aparea.
*/
CREATE OR REPLACE FUNCTION f8_zxc(v_nume echipa_zxc.pseudonim_echipa%type)
RETURN NUMBER IS
    v_penalizare NUMBER(5,2);
    v_check NUMBER(5);
    v_rep NUMBER(5,2);
    NO_REPUTATION EXCEPTION;
    GO_TO_OTHERS EXCEPTION;
BEGIN

    if REGEXP_LIKE(v_nume,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;
    -- Daca transmitem o valoare numerica ca parametru -> sarim la others.

    select count(*) into v_check
    from echipa_zxc
    where lower(pseudonim_echipa) = lower(v_nume);
    if v_check = 0 then
        RAISE NO_DATA_FOUND;
    end if;


    select count(*) into v_penalizare
    from echipa_zxc e
    join produs_zxc p on (e.id_echipa = p.id_echipa)
    join foloseste_zxc f on (p.cod_fabricatie = f.cod_fabricatie)
    where lower(e.pseudonim_echipa) = lower(v_nume) and f.denumire_furnizor is NULL;

    v_penalizare:=v_penalizare*1.5;

    select reputatie into v_rep
    from echipa_zxc
    where lower(pseudonim_echipa) = lower(v_nume);

    if v_rep < v_penalizare then
        RAISE NO_REPUTATION;
    end if;

    return v_penalizare;
EXCEPTION

    when NO_DATA_FOUND then
    dbms_output.put_line('Nicio␣echipa␣nu␣are␣acel␣pseudonim');
    return -1;

    when NO_REPUTATION then
    dbms_output.put_line('Reputatia␣este␣prea␣mica␣pentru␣a␣se␣penaliza');
    return -2;

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare.␣Verifica␣datele␣introduse');
    return -3;
END f8_zxc;
/

-- Functioneaza:
select f8_zxc('Razboinicii') from dual;

-- No_Data_Found
variable x number
execute :x := f8_zxc('Rzb');

-- No_Reputation:
commit;
update echipa_zxc
```

```
set reputatie = 1
where pseudonim_echipa = 'Razboinicii';

variable y number
execute :y := f8_zxc('Razboinicii');
rollback;

–– Others: (in cazul nostru, transmitem un numar sau un string ce contine doar cifre sau apar alte erori.)
variable x number
execute :x := f8_zxc(1233);
```

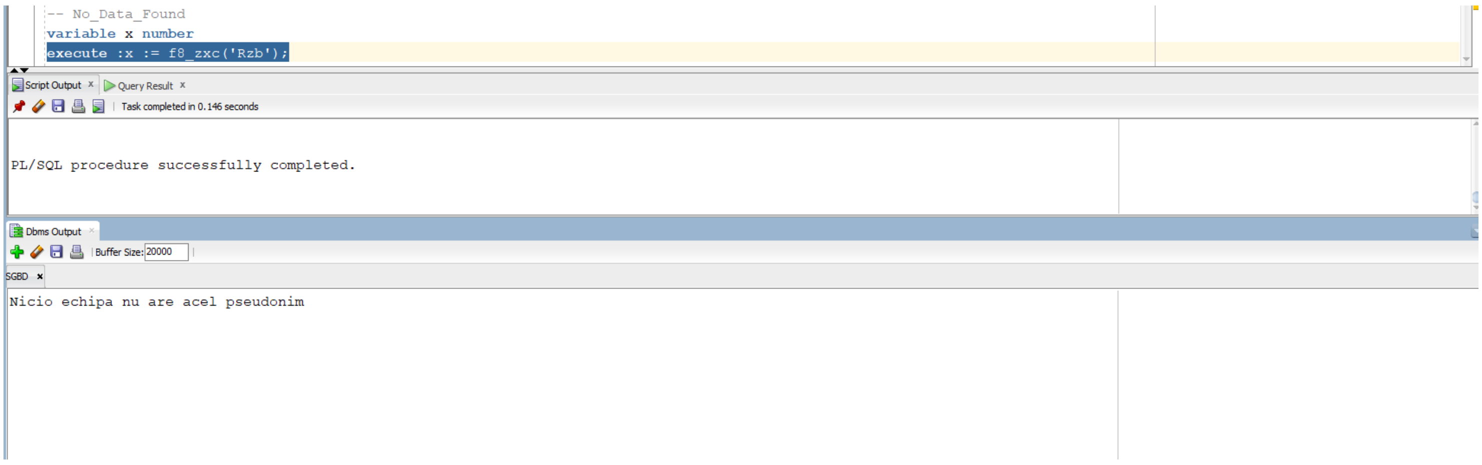


```
-- Functioneaza:
select f8_zxc('Razboinicii') from dual;
```

All Rows Fetched: 1 in 0.013 seconds

	F8_ZXC('RAZBOINICII')
1	1.5

Subprogramul Functioneaza



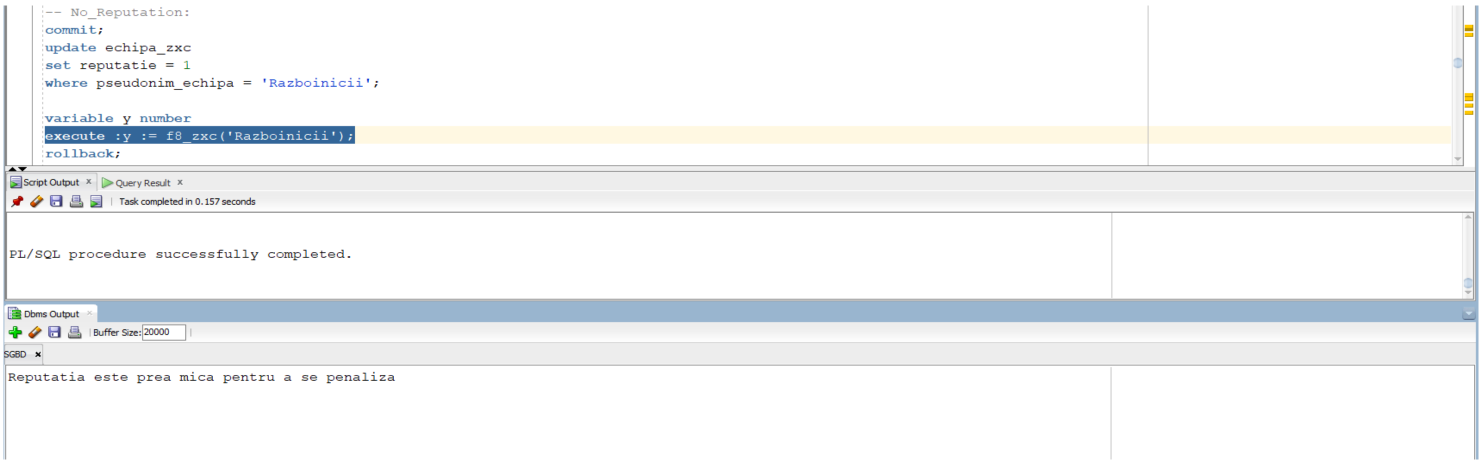
```
-- No_Data_Found
variable x number
execute :x := f8_zxc('Rzb');
```

Task completed in 0.146 seconds

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

```
Nicio echipa nu are acel pseudonim
```

Exceptia No-Data-Found



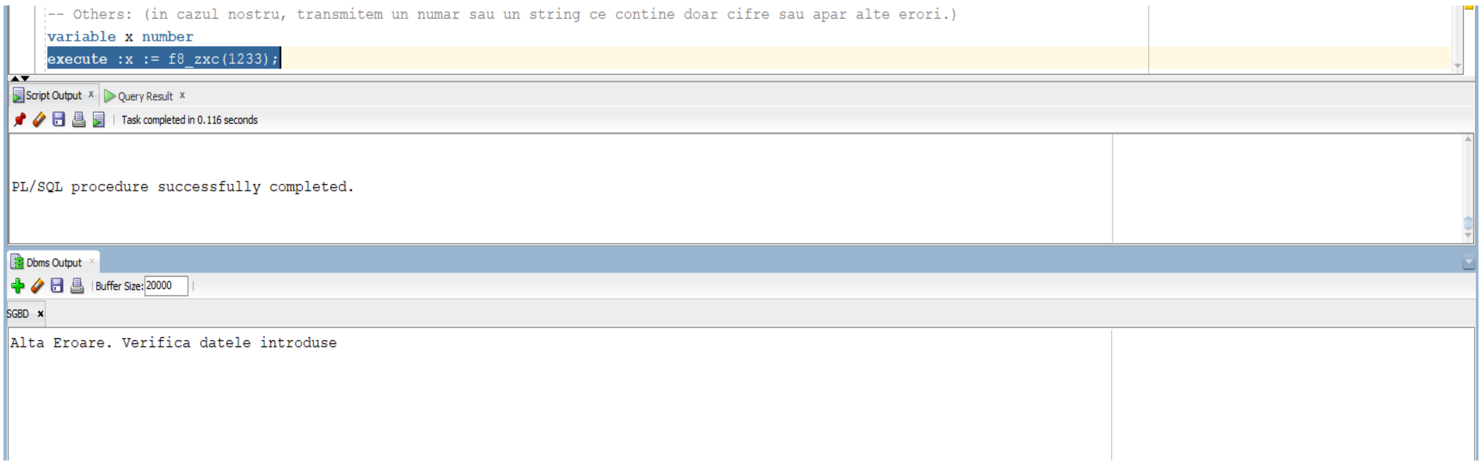
```
-- No_Reputation:
commit;
update echipa_zxc
set reputatie = 1
where pseudonim_echipa = 'Razboinicii';

variable y number
execute :y := f8_zxc('Razboinicii');
rollback;
```

Task completed in 0.157 seconds

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

```
Reputatia este prea mica pentru a se penaliza
```

Exceptia No-Reputation



```
-- Others: (in cazul nostru, transmitem un numar sau un string ce contine doar cifre sau apar alte erori.)
variable x number
execute :x := f8_zxc(1233);
```

```
Task completed in 0.116 seconds

PL/SQL procedure successfully completed.
```

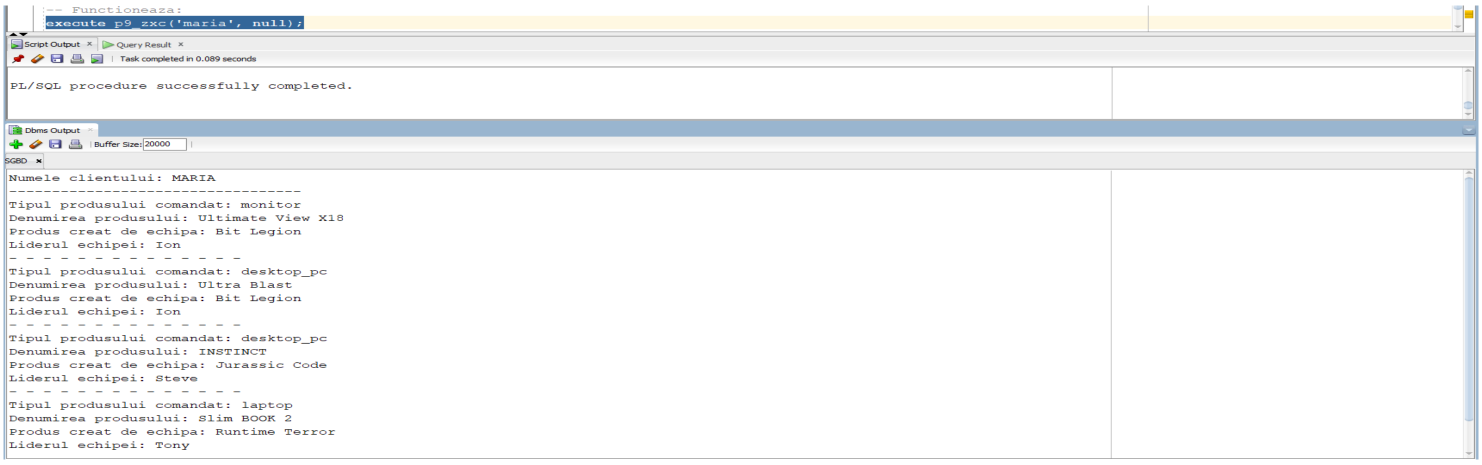
```
Alta Eroare. Verifica datele introduse
```

Exceptia Others

9 Punctul 9:

```
/* Formulare: Creati o procedura stocata cu doi parametrii:
- nume client -> camp obligatoriu
- prenume client -> camp optional (poate fii null)
Cu ajutorul acestei proceduri sa se afiseze tipul si numele produselor,pseudonimul echipelor ce le-au
creat , dar si numele liderilor acestora pentru toate produsele comandate de un client al carui nume
este dat ca parametru ( fie doar numele de familie, fie numele si prenumele)
Tratati erorile ce pot aparea.*/

CREATE OR REPLACE PROCEDURE p9_zxc(v_client_n client_zxc.nume%type, v_client_p client_zxc.prenume%type)
IS
    type tip_record_zxc is RECORD(
    v_tp produs_zxc.tip_produs%type,
    v_produs produs_zxc.denumire_produs%type,
    v_ech echipa_zxc.pseudonim_echipa%type,
    v_lider salariat_zxc.prenume%type);
    type record_table_zxc is TABLE of tip_record_zxc;
    v_rec record_table_zxc;
    v_check NUMBER;
    EMPTY_INP EXCEPTION;
    GO_TO_OTHERS EXCEPTION; -- o vom folosi pentru a declansa automat clauza When Others.
BEGIN

    if v_client_n is NULL and v_client_p is NULL then
        RAISE EMPTY_INP;
    end if;

    if REGEXP_LIKE(v_client_n,'^[0-9]+$') or REGEXP_LIKE(v_client_p,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;
    -- Daca transmitem o valoare numerica ca parametru -> sarim la others.

    CASE
    WHEN v_client_n is NULL then
        RAISE EMPTY_INP;
    WHEN v_client_p is NULL then
        select count(*) into v_check
        from client_zxc
        where upper(nume)= upper(v_client_n);


        if v_check = 0 then
            RAISE NO_DATA_FOUND;
        else
            if v_check > 1 then
                RAISE TOO_MANY_ROWS;
            end if;
        end if;

        select p.tip_produs, p.denumire_produs,e.pseudonim_echipa,s.nume
        bulk collect into v_rec
        from salariat_zxc s join salariat_zxc mg on(s.cnp_salariat = mg.cnp_salariat)
        join echipa_zxc e on (mg.cnp_salariat=e.lider)
        join produs_zxc p on (p.id_echipa = e.id_echipa)
        join comanda_zxc cm on (cm.cod_fabricatie=p.cod_fabricatie)
        join client_zxc c on (cm.id_client=c.id_client)
        where upper(c.nume)=upper(v_client_n);

        dbms_output.put_line('Numele␣clientului:␣' || upper(v_client_n));
        dbms_output.put_line('−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−');

        for i in v_rec.FIRST..v_rec.LAST loop
            dbms_output.put_line('Tipul␣produsului␣comandat:␣' || v_rec(i).v_tp);
            dbms_output.put_line('Denumirea␣produsului:␣' || v_rec(i).v_produs);
            dbms_output.put_line('Produs␣creat␣de␣echipa:␣' || v_rec(i).v_ech);
            dbms_output.put_line('Liderul␣echipei:␣' || v_rec(i).v_lider);
            dbms_output.put_line('−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−');
        end loop;


        WHEN v_client_p is not NULL then
        select count(*) into v_check
        from client_zxc
        where upper(nume)= upper(v_client_n) and upper(prenume)= upper(v_client_p);


        if v_check = 0 then
            RAISE NO_DATA_FOUND;
        else
            if v_check > 1 then
                RAISE TOO_MANY_ROWS;
            end if;
        end if;

        select p.tip_produs, p.denumire_produs,e.pseudonim_echipa,s.nume
        bulk collect into v_rec
        from salariat_zxc s join salariat_zxc mg on(s.cnp_salariat = mg.cnp_salariat)
        join echipa_zxc e on (mg.cnp_salariat=e.lider)
        join produs_zxc p on (p.id_echipa = e.id_echipa)
        join comanda_zxc cm on (cm.cod_fabricatie=p.cod_fabricatie)
        join client_zxc c on (cm.id_client=c.id_client)
        where upper(c.nume)=upper(v_client_n) and upper(c.prenume)= upper(v_client_p);

        dbms_output.put_line('Numele␣clientului:␣' || upper(v_client_n));
        dbms_output.put_line('−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−');

        for i in v_rec.FIRST..v_rec.LAST loop
            dbms_output.put_line('Tipul␣produsului␣comandat:␣' || v_rec(i).v_tp);
            dbms_output.put_line('Denumirea␣produsului:␣' || v_rec(i).v_produs);
            dbms_output.put_line('Produs␣creat␣de␣echipa:␣' || v_rec(i).v_ech);
            dbms_output.put_line('Liderul␣echipei:␣' || v_rec(i).v_lider);
            dbms_output.put_line('−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−␣−');
        end loop;
        END CASE;

EXCEPTION
    when TOO_MANY_ROWS then
    dbms_output.put_line('Prea␣multi␣clienti␣cu␣acelasi␣nume.␣Incearca␣sa␣specifici␣si␣prenumele');

    when NO_DATA_FOUND then
    dbms_output.put_line('Nu␣exista␣un␣client␣cu␣numele␣dat');

    when EMPTY_INP then
    dbms_output.put_line('Eroare:␣Campul␣obligatoriu␣sau␣ambele␣campuri␣sunt␣NULL');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare.␣Verifica␣parametrii␣introdusi');

END p9_zxc;
/

−− Functioneaza:
execute p9_zxc('maria', null);
execute p9_zxc('ion', 'dorian');

−− TOO_MANY_ROWS: Programul iti va cere si prenumele pentru a face o selectie corecta
execute p9_zxc('ion', null);

−− No_Data_Found:
execute p9_zxc('lap',null);
execute p9_zxc('Ion','ion');
```

```
-- Empty_Inp:
execute p9_zxc(null,'dorian');
execute p9_zxc(null, null);

-- WHEN OTHERS / go_to_OTHERS
execute p9_zxc(123, 'dorian');
```

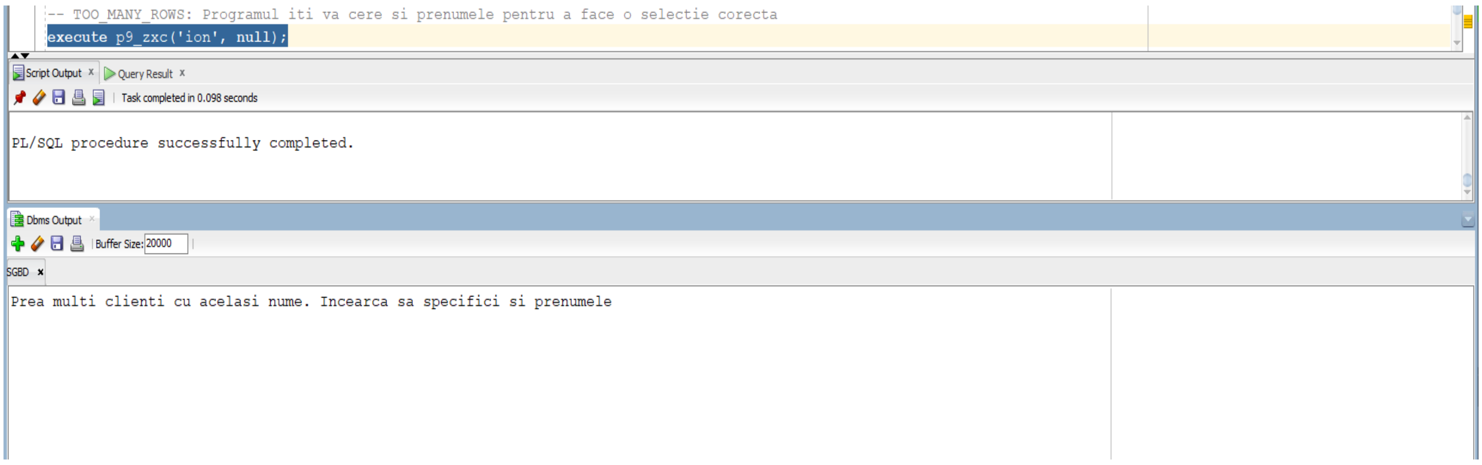


```
-- Functioneaza:
execute p9_zxc('maria', null);
```

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

```
Numele clientului: MARIA
----------------------------------
Tipul produsului comandat: monitor
Denumirea produsului: Ultimate View X18
Produs creat de echipa: Bit Legion
Liderul echipei: Ion
- - - - - - - - - - - - - -
Tipul produsului comandat: desktop_pc
Denumirea produsului: Ultra Blast
Produs creat de echipa: Bit Legion
Liderul echipei: Ion
- - - - - - - - - - - - - -
Tipul produsului comandat: desktop_pc
Denumirea produsului: INSTINCT
Produs creat de echipa: Jurassic Code
Liderul echipei: Steve
- - - - - - - - - - - - - -
Tipul produsului comandat: laptop
Denumirea produsului: Slim BOOK 2
Produs creat de echipa: Runtime Terror
Liderul echipei: Tony
```

Subprogramul Functioneaza

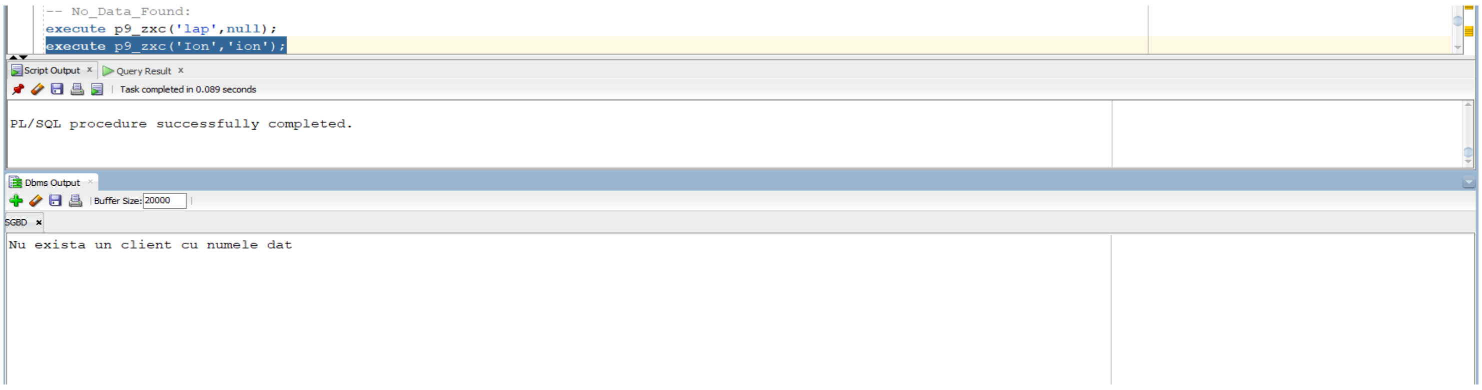


```
-- TOO_MANY_ROWS: Programul iti va cere si prenumele pentru a face o selectie corecta
execute p9_zxc('ion', null);
```

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

```
Prea multi clienti cu acelasi nume. Incearca sa specifici si prenumele
```

Exceptia Too-Many-Rows

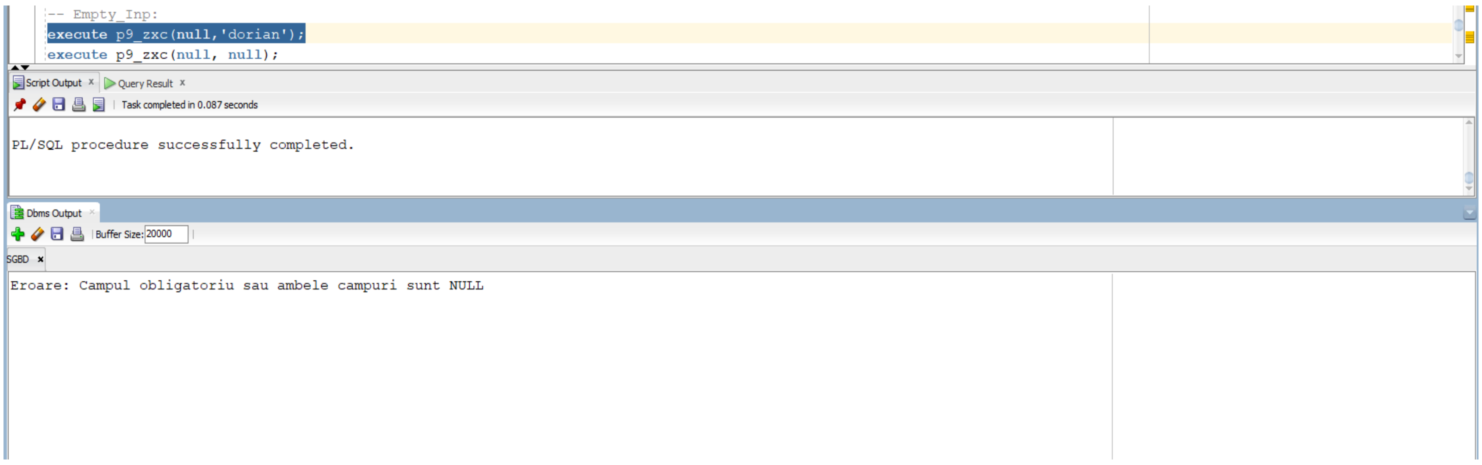


```
-- No_Data_Found:
execute p9_zxc('lap',null);
execute p9_zxc('Ion','ion');
```

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

```
Nu exista un client cu numele dat
```

Exceptia No-Data-Found

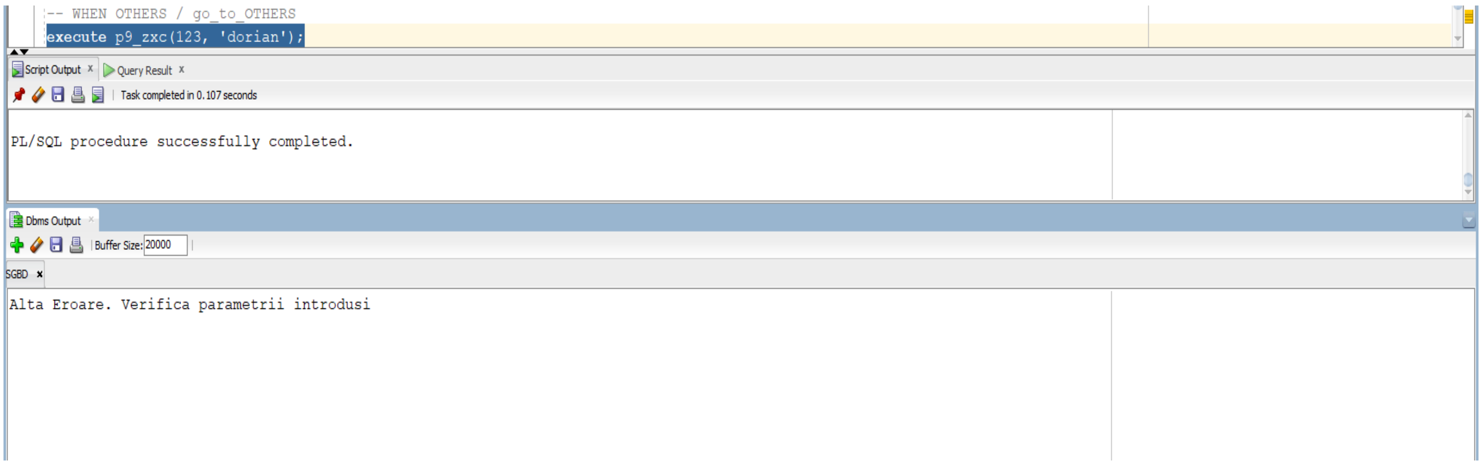


```
-- Empty_Inp:
execute p9_zxc(null,'dorian');
execute p9_zxc(null, null);
```

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

```
Eroare: Campul obligatoriu sau ambele campuri sunt NULL
```

Exceptia Empty-Input



```
-- WHEN OTHERS / go_to_OTHERS
execute p9_zxc(123, 'dorian');
```

Task completed in 0.107 seconds

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

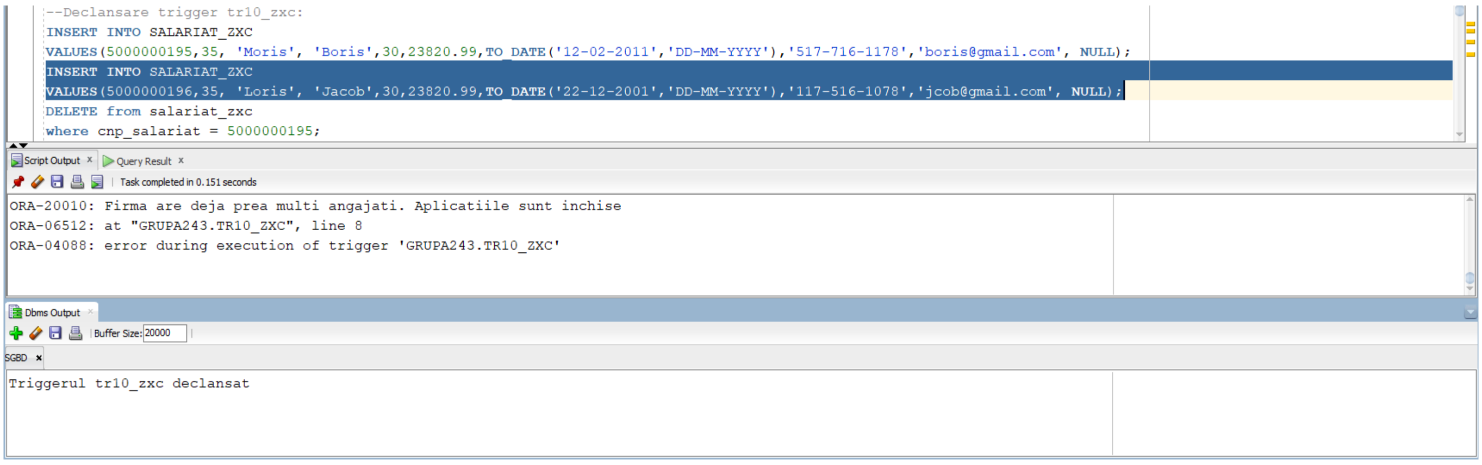
Buffer Size: 20000

```
Alta Eroare. Verifica parametrii introdusi
```

Exceptia Others

10 Punctul 10:

```
/* Formulare: Intrucat firma noastra are un nr limitat de sedii, se doreste sa se limiteze numarul total de posturi
disponibile in companie.
Sa se proiecteze un trigger ce opreste inserarea noilor angajati atunci cand firma si-a atins maximul de 19 angajati*/
CREATE OR REPLACE TRIGGER tr10_zxc
BEFORE INSERT ON salariat_zxc
DECLARE
v_count NUMBER;
BEGIN
    select count(*) into v_count
    from salariat_zxc;
    if (v_count >=19) then
    dbms_output.put_line('Triggerul␣tr10_zxc␣declansat');
        RAISE_APPLICATION_ERROR(-20010, 'Firma␣are␣deja␣prea␣multi␣angajati.␣Aplicatiile␣sunt␣inchise');
    end if;
END tr10_zxc;
/
--Declansare trigger tr10_zxc:
INSERT INTO SALARIAT_ZXC
VALUES(5000000195,35, 'Moris', 'Boris',30,23820.99,TO_DATE('12-02-2011','DD-MM-YYYY'),'517-716-1178','
    ↪ boris@gmail.com', NULL);
INSERT INTO SALARIAT_ZXC
VALUES(5000000196,35, 'Loris', 'Jacob',30,23820.99,TO_DATE('22-12-2001','DD-MM-YYYY'),'117-516-1078','
    ↪ jcob@gmail.com', NULL);
DELETE from salariat_zxc
where cnp_salariat = 5000000195;
```



```
--Declansare trigger tr10_zxc:
INSERT INTO SALARIAT_ZXC
VALUES(5000000195,35, 'Moris', 'Boris',30,23820.99,TO_DATE('12-02-2011','DD-MM-YYYY'),'517-716-1178','boris@gmail.com', NULL);
INSERT INTO SALARIAT_ZXC
VALUES(5000000196,35, 'Loris', 'Jacob',30,23820.99,TO_DATE('22-12-2001','DD-MM-YYYY'),'117-516-1078','jcob@gmail.com', NULL);
DELETE from salariat_zxc
where cnp_salariat = 5000000195;
```

Task completed in 0.151 seconds

```
ORA-20010: Firma are deja prea multi angajati. Aplicatiile sunt inchise
ORA-06512: at "GRUPA243.TR10_ZXC", line 8
ORA-04088: error during execution of trigger 'GRUPA243.TR10_ZXC'
```

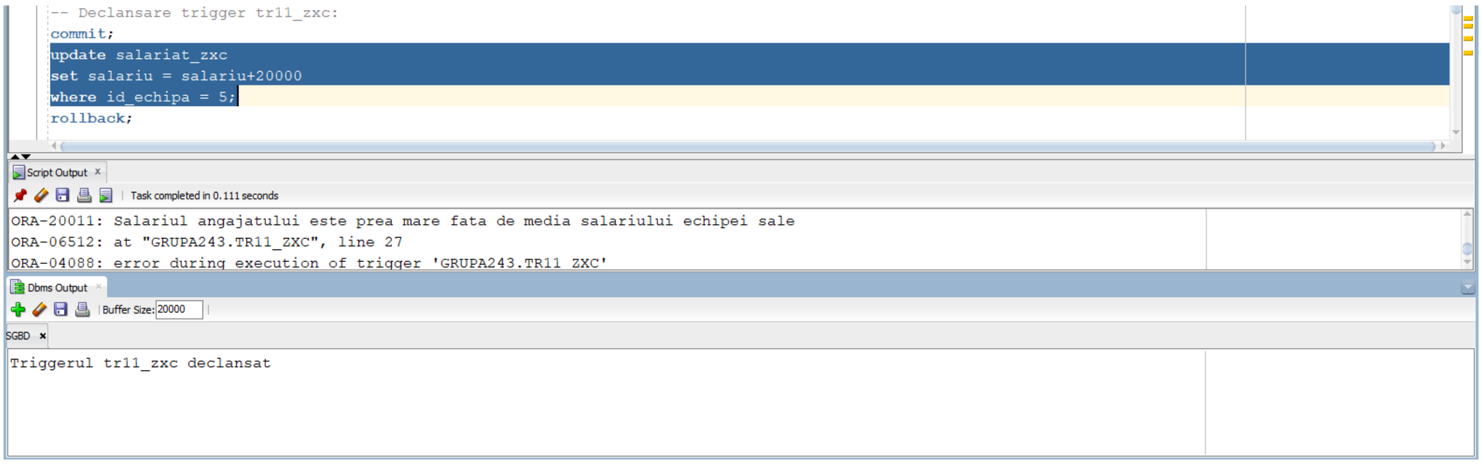
Buffer Size: 20000

```
Triggerul tr10_zxc declansat
```

Tr10-zxc a fost declansat

11 Punctul 11:

```
/* Formulare: Salariatii activi pot primi o marire de salariu, insa aceasta marire nu are voie sa
depaseasca 15% din valoarea medie a salariilor membrilor echipei sale.
Astfel, se cere definirea unui trigger ce se ocupa cu gestionarea ulterioarelor modificari ale salariilor obtinute
de angajatii unei anumite echipe.
Triggerul se va declansa in momentul in care un salariat primeste o marire de salariu ce depaseste
15% din valoarea salariului mediu al echipei din care face parte.*/
CREATE OR REPLACE TRIGGER tr11_zxc
FOR UPDATE OF salariu ON salariat_zxc
COMPOUND TRIGGER

type salarii_rec_zxc is RECORD(
v_slr salariat_zxc.salariu%type,
v_ech echipa_zxc.id_echipa%type);
type salarii_colect_zxc is table of salarii_rec_zxc;
info_slr salarii_colect_zxc;
type medie_salarii_zxc is table of salariat_zxc.salariu%type index by varchar2(120);
avg_slr medie_salarii_zxc;

BEFORE STATEMENT IS
BEGIN
    select avg(salariu), NVL(id_echipa,−1)
    bulk collect into info_slr
    from salariat_zxc
    group by id_echipa;

    for i in info_slr.first..info_slr.last loop
        avg_slr(info_slr(i).v_ech):=info_slr(i).v_slr;
    end loop;
end before statement;

AFTER EACH ROW IS
BEGIN
    if :NEW.salariu − :OLD.salariu > (0.15)∗avg_slr(:NEW.id_echipa) then
    dbms_output.put_line('Triggerul␣tr11_zxc␣declansat');
        RAISE_APPLICATION_ERROR(−20011, 'Salariul␣angajatului␣este␣prea␣mare␣fata␣de␣media␣salariului␣echipei␣
            ↪ sale');
    end if;
end after each row;
END tr11_zxc;
/
−− Declansare trigger tr11_zxc:
commit;
update salariat_zxc
set salariu = salariu+20000
where id_echipa = 5;
rollback;
```



```
-- Declansare trigger tr11_zxc:
commit;
update salariat_zxc
set salariu = salariu+20000
where id_echipa = 5;
rollback;
```

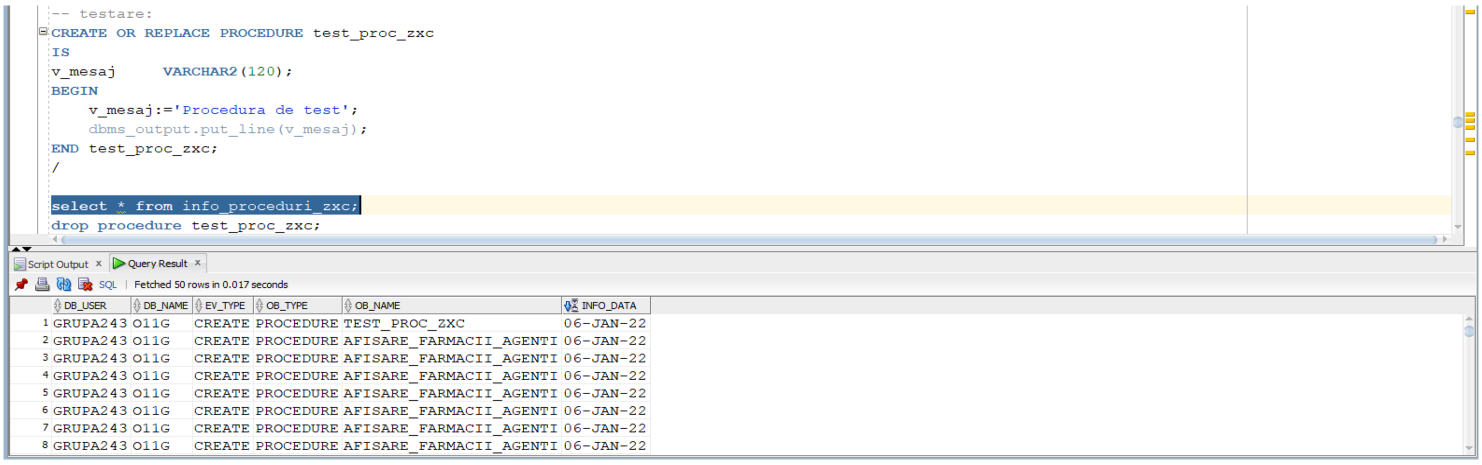
```
ORA-20011: Salariul angajatului este prea mare fata de media salariului echipei sale
ORA-06512: at "GRUPA243.TR11_ZXC", line 27
ORA-04088: error during execution of trigger 'GRUPA243.TR11_ZXC'
```

```
Triggerul tr11_zxc declansat
```

Tr11-zxc a fost declansat

12 Punctul 12:

```
/* Formulare: Creati un trigger la nivel de schema ce va memora informatii despre toate modificarile
tip LDD efectuate asupra PROCEDURILOR stocate in baza de date (CREATE , ALTER si DROP).
Salvati informatiile intr-o tabela independenta (care nu se afla in nicio schema relationala cu
celelalte tabele ale bazei de date.*/
CREATE TABLE info_proceduri_zxc
(
    db_user VARCHAR2(220),
    db_name VARCHAR2(220),
    ev_type VARCHAR2(220),
    ob_type VARCHAR2(220),
    ob_name VARCHAR2(220),
    info_data DATE
);


CREATE OR REPLACE TRIGGER tr12_zxc
AFTER CREATE OR ALTER OR DROP
ON SCHEMA
DECLARE
v_tip_ob VARCHAR2(220);
BEGIN
    v_tip_ob:=SYS.DICTIONARY_OBJ_TYPE;
    if upper(v_tip_ob) like 'PROCEDURE' then
        INSERT INTO info_proceduri_zxc
        VALUES(SYS.LOGIN_USER, SYS.DATABASE_NAME, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_TYPE,
            ↪ SYS.DICTIONARY_OBJ_NAME , SYSDATE);
    end if;
END tr12_zxc;
/

-- testare:
CREATE OR REPLACE PROCEDURE test_proc_zxc
IS
v_mesaj VARCHAR2(120);
BEGIN
    v_mesaj:='Procedura␣de␣test';
    dbms_output.put_line(v_mesaj);
END test_proc_zxc;
/

select * from info_proceduri_zxc;
drop procedure test_proc_zxc;
```



```
-- testare:
CREATE OR REPLACE PROCEDURE test_proc_zxc
IS
v_mesaj     VARCHAR2(120);
BEGIN
    v_mesaj:='Procedura de test';
    dbms_output.put_line(v_mesaj);
END test_proc_zxc;
/

select * from info_proceduri_zxc;
drop procedure test_proc_zxc;
```

Fetched 50 rows in 0.017 seconds

```
  DB_USER  DB_NAME EV_TYPE OB_TYPE   OB_NAME                  INFO_DATA
1 GRUPA243 O11G    CREATE  PROCEDURE TEST_PROC_ZXC            06-JAN-22
2 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
3 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
4 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
5 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
6 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
7 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
8 GRUPA243 O11G    CREATE  PROCEDURE AFISARE_FARMACII_AGENTI 06-JAN-22
```

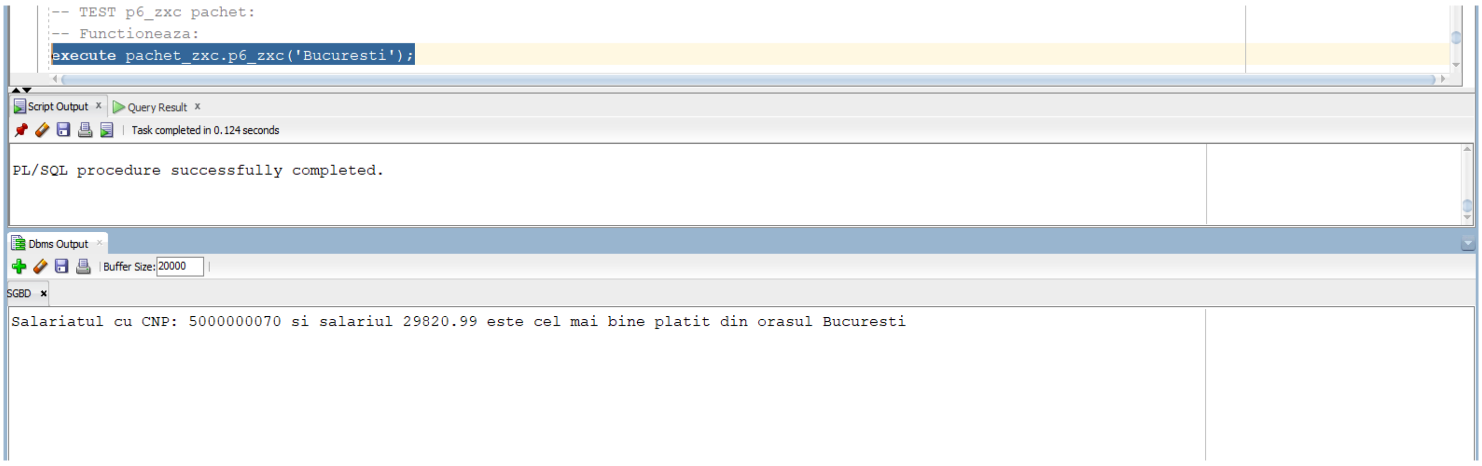
Informatiile prelucrate de Tr12-zxc au fost salvate

13 Punctul 13:

```
CREATE OR REPLACE PACKAGE pachet_zxc
AS
PROCEDURE p6_zxc(v_oras sediu_zxc.oras%type);
PROCEDURE p7_zxc(v_specializare specializare_zxc.titlu%type);
FUNCTION f8_zxc(v_nume echipa_zxc.pseudonim_echipa%type) RETURN NUMBER;
PROCEDURE p9_zxc(v_client_n client_zxc.nume%type, v_client_p client_zxc.prenume%type);
PROCEDURE test_proc_zxc;
END pachet_zxc;
/


CREATE OR REPLACE PACKAGE BODY pachet_zxc
IS
PROCEDURE p6_zxc(v_oras sediu_zxc.oras%type)
IS
    type echipe_nested is table of sediu_zxc.id_echipa%type;
    v_echipa echipe_nested;
    type salariat_index is table of salariat_zxc%rowtype index by pls_integer;
    v_salariat salariat_index;
    v_max number(10,2):=0;
    v_cnp number(10,0);
    v_age number(3,0):=999;
    v_check number;
    GO_TO_OTHERS EXCEPTION;
BEGIN
    if REGEXP_LIKE(v_oras,'^[0−9]+$') then
        RAISE GO_TO_OTHERS;
    end if;

    select count(∗) into v_check
    from sediu_zxc
    where upper(oras) = upper(v_oras);
    if v_check = 0 then
        RAISE NO_DATA_FOUND;
    end if;

    select id_echipa
    bulk collect into v_echipa
    from sediu_zxc
    where upper(oras) = upper(v_oras);

    for i in v_echipa.FIRST..v_echipa.LAST loop
        select ∗
        bulk collect into v_salariat
        from salariat_zxc
        where id_echipa = v_echipa(i);
        for j in v_salariat.FIRST..v_salariat.LAST loop
            if v_salariat(j).salariu > v_max then
                v_age:= v_salariat(j).varsta;
                v_max:= v_salariat(j).salariu;
                v_cnp:= v_salariat(j).cnp_salariat;
            else
                if v_salariat(j).salariu = v_max and v_salariat(j).varsta < v_age then
                    v_age:= v_salariat(j).varsta;
                    v_max:= v_salariat(j).salariu;
                    v_cnp:= v_salariat(j).cnp_salariat;
                end if;
            end if;
        end loop;
        v_salariat.DELETE;
    end loop;
    v_echipa.DELETE;

    dbms_output.put_line('Salariatul␣cu␣CNP:␣'||v_cnp||'␣si␣salariul␣'||v_max||'␣este␣cel␣mai␣bine␣platit␣din␣orasul␣'||v_oras
        ↪ );
EXCEPTION
    when NO_DATA_FOUND then
    dbms_output.put_line('Nicio␣echipa␣nu␣are␣sediul␣in␣Orasul␣dat.');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare');

END p6_zxc;


PROCEDURE p7_zxc(v_specializare specializare_zxc.titlu%type)
```

```
IS

    CURSOR specializari (specializare_angajat specializare_zxc.id_specializare%TYPE) IS
    select s.nume, s.prenume, s.id_echipa
    from salariat_zxc s, detine_zxc d
    where s.cnp_salariat = d.cnp_salariat and d.id_specializare = specializare_angajat;

    v_spec specializare_zxc.id_specializare%type;
    v_nume salariat_zxc.nume%type;
    v_prenume salariat_zxc.prenume%type;
    v_echipa salariat_zxc.id_echipa%type;
    GO_TO_OTHERS EXCEPTION;
BEGIN
    if REGEXP_LIKE(v_specializare,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;

    select id_specializare
    into v_spec
    from specializare_zxc
    where lower(titlu) = lower(v_specializare);

    dbms_output.put_line('Specializarea:␣' || v_specializare);
    dbms_output.put_line('----------------------------------');

    open specializari(v_spec);

    loop
        fetch specializari into v_nume, v_prenume, v_echipa;
        exit when specializari%NOTFOUND;
        dbms_output.put_line(v_nume||'␣'||v_prenume||'␣din␣echipa␣'||v_echipa);
    end loop;

    if specializari%rowcount = 0 then
        dbms_output.put_line('Nimeni␣nu␣are␣aceasta␣specializare');
    end if;
    close specializari;
    dbms_output.new_line;
EXCEPTION
    when TOO_MANY_ROWS then
    dbms_output.put_line('Prea␣multe␣specializari␣cu␣acelasi␣nume');

    when NO_DATA_FOUND then
    dbms_output.put_line('Nu␣exista␣specializarea␣specificata');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare');
END p7_zxc;


FUNCTION f8_zxc(v_nume echipa_zxc.pseudonim_echipa%type)
RETURN NUMBER IS
    v_penalizare NUMBER(5,2);
    v_check NUMBER(5);
    v_rep NUMBER(5,2);
    NO_REPUTATION EXCEPTION;
    GO_TO_OTHERS EXCEPTION;
BEGIN

    if REGEXP_LIKE(v_nume,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;
    -- Daca transmitem o valoare numerica ca parametru -> sarim la others.

    select count(*) into v_check
    from echipa_zxc
    where lower(pseudonim_echipa) = lower(v_nume);
    if v_check = 0 then
        RAISE NO_DATA_FOUND;
    end if;


    select count(*) into v_penalizare
    from echipa_zxc e
    join produs_zxc p on (e.id_echipa = p.id_echipa)
    join foloseste_zxc f on (p.cod_fabricatie = f.cod_fabricatie)
    where lower(e.pseudonim_echipa) = lower(v_nume) and f.denumire_furnizor is NULL;

    v_penalizare:=v_penalizare*1.5;

    select reputatie into v_rep
    from echipa_zxc
    where lower(pseudonim_echipa) = lower(v_nume);

    if v_rep < v_penalizare then
        RAISE NO_REPUTATION;
    end if;

    return v_penalizare;
EXCEPTION

    when NO_DATA_FOUND then
    dbms_output.put_line('Nicio␣echipa␣nu␣are␣acel␣pseudonim');
    return -1;

    when NO_REPUTATION then
    dbms_output.put_line('Reputatia␣este␣prea␣mica␣pentru␣a␣se␣penaliza');
    return -2;

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare.␣Verifica␣datele␣introduse');
    return -3;
END f8_zxc;


PROCEDURE p9_zxc(v_client_n client_zxc.nume%type, v_client_p client_zxc.prenume%type)
IS
    type tip_record_zxc is RECORD(
    v_tp produs_zxc.tip_produs%type,
    v_produs produs_zxc.denumire_produs%type,
    v_ech echipa_zxc.pseudonim_echipa%type,
    v_lider salariat_zxc.prenume%type);
    type record_table_zxc is TABLE of tip_record_zxc;
    v_rec record_table_zxc;
    v_check NUMBER;
    EMPTY_INP EXCEPTION;
    GO_TO_OTHERS EXCEPTION; -- o vom folosi pentru a declansa automat clauza When Others.
BEGIN

    if v_client_n is NULL and v_client_p is NULL then
        RAISE EMPTY_INP;
    end if;

    if REGEXP_LIKE(v_client_n,'^[0-9]+$') or REGEXP_LIKE(v_client_p,'^[0-9]+$') then
        RAISE GO_TO_OTHERS;
    end if;
    -- Daca transmitem o valoare numerica ca parametru -> sarim la others.

    CASE
    WHEN v_client_n is NULL then
        RAISE EMPTY_INP;
    WHEN v_client_p is NULL then
        select count(*) into v_check
        from client_zxc
        where upper(nume)= upper(v_client_n);


        if v_check = 0 then
            RAISE NO_DATA_FOUND;
        else
            if v_check > 1 then
                RAISE TOO_MANY_ROWS;
            end if;
        end if;

        select p.tip_produs, p.denumire_produs,e.pseudonim_echipa,s.nume
        bulk collect into v_rec
        from salariat_zxc s join salariat_zxc mg on(s.cnp_salariat = mg.cnp_salariat)
        join echipa_zxc e on (mg.cnp_salariat=e.lider)
        join produs_zxc p on (p.id_echipa = e.id_echipa)
        join comanda_zxc cm on (cm.cod_fabricatie=p.cod_fabricatie)
        join client_zxc c on (cm.id_client=c.id_client)
        where upper(c.nume)=upper(v_client_n);

        dbms_output.put_line('Numele␣clientului:␣' || upper(v_client_n));
        dbms_output.put_line('------------------------------------');
        for i in v_rec.FIRST..v_rec.LAST loop
            dbms_output.put_line('Tipul␣produsului␣comandat:␣' || v_rec(i).v_tp);
            dbms_output.put_line('Denumirea␣produsului:␣' || v_rec(i).v_produs);
            dbms_output.put_line('Produs␣creat␣de␣echipa:␣' || v_rec(i).v_ech);
            dbms_output.put_line('Liderul␣echipei:␣' || v_rec(i).v_lider);
            dbms_output.put_line('-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-');
        end loop;


        WHEN v_client_p is not NULL then
        select count(*) into v_check
        from client_zxc
        where upper(nume)= upper(v_client_n) and upper(prenume)= upper(v_client_p);


        if v_check = 0 then
            RAISE NO_DATA_FOUND;
        else
            if v_check > 1 then
                RAISE TOO_MANY_ROWS;
            end if;
        end if;

        select p.tip_produs, p.denumire_produs,e.pseudonim_echipa,s.nume
        bulk collect into v_rec
        from salariat_zxc s join salariat_zxc mg on(s.cnp_salariat = mg.cnp_salariat)
        join echipa_zxc e on (mg.cnp_salariat=e.lider)
        join produs_zxc p on (p.id_echipa = e.id_echipa)
        join comanda_zxc cm on (cm.cod_fabricatie=p.cod_fabricatie)
        join client_zxc c on (cm.id_client=c.id_client)
        where upper(c.nume)=upper(v_client_n) and upper(c.prenume)= upper(v_client_p);

        dbms_output.put_line('Numele␣clientului:␣' || upper(v_client_n));
        dbms_output.put_line('-----------------------------------');

        for i in v_rec.FIRST..v_rec.LAST loop
            dbms_output.put_line('Tipul␣produsului␣comandat:␣' || v_rec(i).v_tp);
            dbms_output.put_line('Denumirea␣produsului:␣' || v_rec(i).v_produs);
            dbms_output.put_line('Produs␣creat␣de␣echipa:␣' || v_rec(i).v_ech);
            dbms_output.put_line('Liderul␣echipei:␣' || v_rec(i).v_lider);
            dbms_output.put_line('-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-␣-');
        end loop;
        END CASE;

EXCEPTION
    when TOO_MANY_ROWS then
    dbms_output.put_line('Prea␣multi␣clienti␣cu␣acelasi␣nume.␣Incearca␣sa␣specifici␣si␣prenumele');

    when NO_DATA_FOUND then
    dbms_output.put_line('Nu␣exista␣un␣client␣cu␣numele␣dat');

    when EMPTY_INP then
    dbms_output.put_line('Eroare:␣Campul␣obligatoriu␣sau␣ambele␣campuri␣sunt␣NULL');

    when OTHERS then
    dbms_output.put_line('Alta␣Eroare.␣Verifica␣parametrii␣introdusi');

END p9_zxc;

-- am inscris si procedura de test folosita la triggerul 12 in pachet. ea nu serveste la nimic
PROCEDURE test_proc_zxc
IS
v_mesaj VARCHAR2(120);
BEGIN
    v_mesaj:='Procedura␣de␣test';
    dbms_output.put_line(v_mesaj);
END test_proc_zxc;

END pachet_zxc;
/

-- TESTARE SUBPROGRAME PACHET:


-- TEST p6_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p6_zxc('Bucuresti');
```

```
-- No_Data_Found:
execute pachet_zxc.p6_zxc(’Barc’);

-- When Others / go_to_Others:
execute pachet_zxc.p6_zxc(123);



-- TEST p7_zxc pachet:
-- Functioneaza
execute pachet_zxc.p7_zxc(’Tester’);

-- No data Found
execute pachet_zxc.p7_zxc(’Ceva’);

-- Too Many Rows
insert into specializare_zxc
values(6,’Tester’);
execute pachet_zxc.p7_zxc(’Tester’);
delete from specializare_zxc
where id_specializare = 6;

-- Others / go_to_Others:
execute pachet_zxc.p7_zxc(123);



-- TEST f8_zxc pachet
-- Functioneaza:
select pachet_zxc.f8_zxc(’Razboinicii’) from dual;

-- No_Data_Found
variable x number
execute :x := pachet_zxc.f8_zxc(’Rzb’);

-- No_Reputation:
commit;
update echipa_zxc
set reputatie = 1
where pseudonim_echipa = ’Razboinicii’;

variable y number
execute :y := pachet_zxc.f8_zxc(’Razboinicii’);
rollback;

-- Others: (in cazul nostru, transmitem un numar sau un string ce contine doar cifre sau apar alte erori.)
variable x number
execute :x := pachet_zxc.f8_zxc(1233);



-- TEST p9_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p9_zxc(’maria’, null);
execute pachet_zxc.p9_zxc(’ion’, ’dorian’);

-- TOO_MANY_ROWS: Programul iti va cere si prenumele pentru a face o selectie corecta
execute pachet_zxc.p9_zxc(’ion’, null);

-- No_Data_Found:
execute pachet_zxc.p9_zxc(’lap’,null);
execute pachet_zxc.p9_zxc(’Ion’,’ion’);

-- Empty_Inp:
execute pachet_zxc.p9_zxc(null,’dorian’);
execute pachet_zxc.p9_zxc(null, null);

-- WHEN OTHERS / go_to_OTHERS
execute pachet_zxc.p9_zxc(123, ’dorian’);


-- TEST test_proc_zxc:
execute pachet_zxc.test_proc_zxc;
```

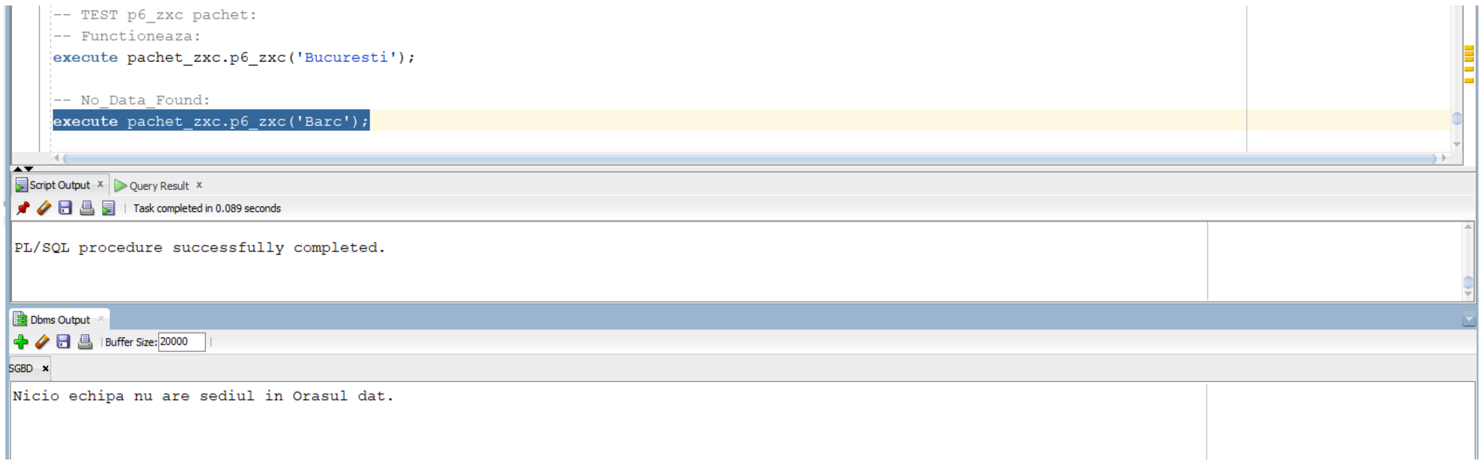


```
-- TEST p6_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p6_zxc('Bucuresti');
```

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

```
Salariatul cu CNP: 5000000070 si salariul 29820.99 este cel mai bine platit din orasul Bucuresti
```

Subprogramul 6 apelat din pachet: functioneaza

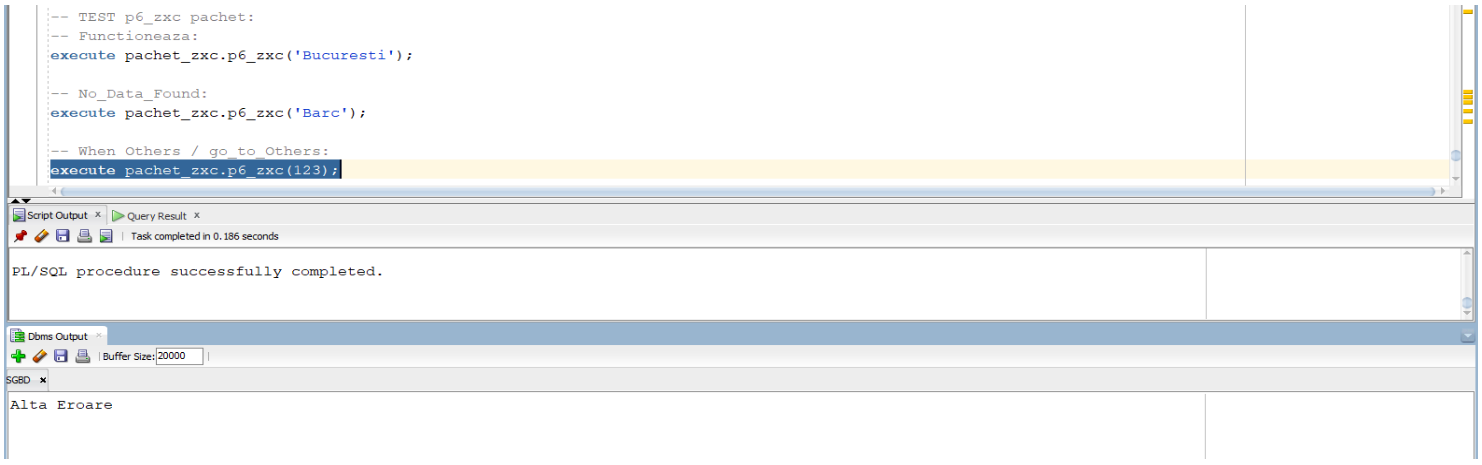


```
-- TEST p6_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p6_zxc('Bucuresti');

-- No_Data_Found:
execute pachet_zxc.p6_zxc('Barc');
```

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

```
Nicio echipa nu are sediul in Orasul dat.
```

Subprogramul 6 apelat din pachet: No-Data-Found

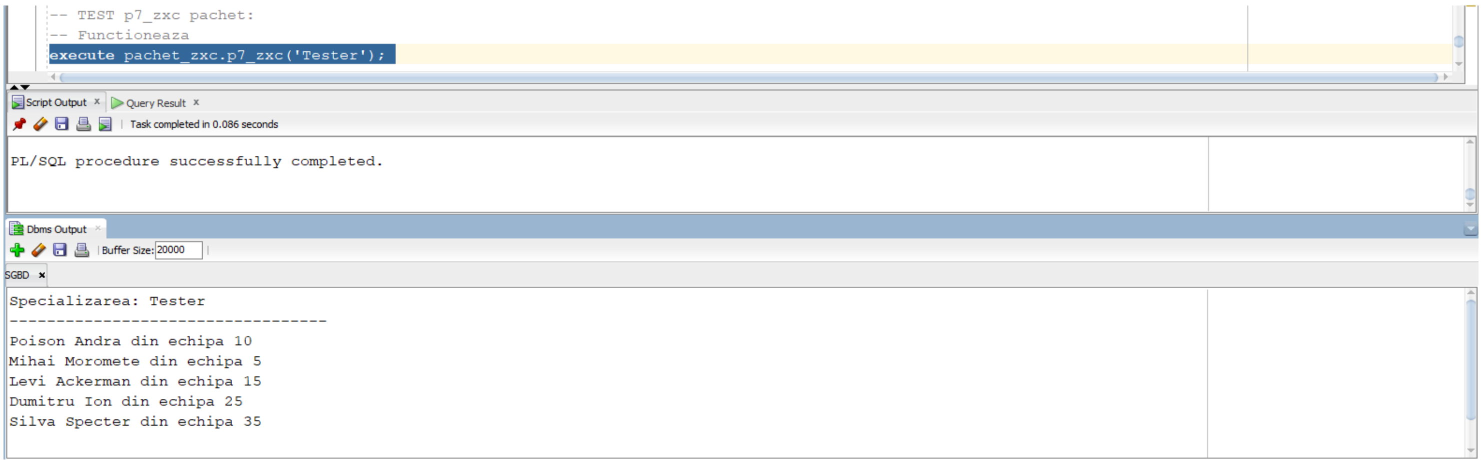


```
-- TEST p6_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p6_zxc('Bucuresti');

-- No_Data_Found:
execute pachet_zxc.p6_zxc('Barc');

-- When Others / go_to_Others:
execute pachet_zxc.p6_zxc(123);
```

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

```
Alta Eroare
```

Subprogramul 6 apelat din pachet: Others

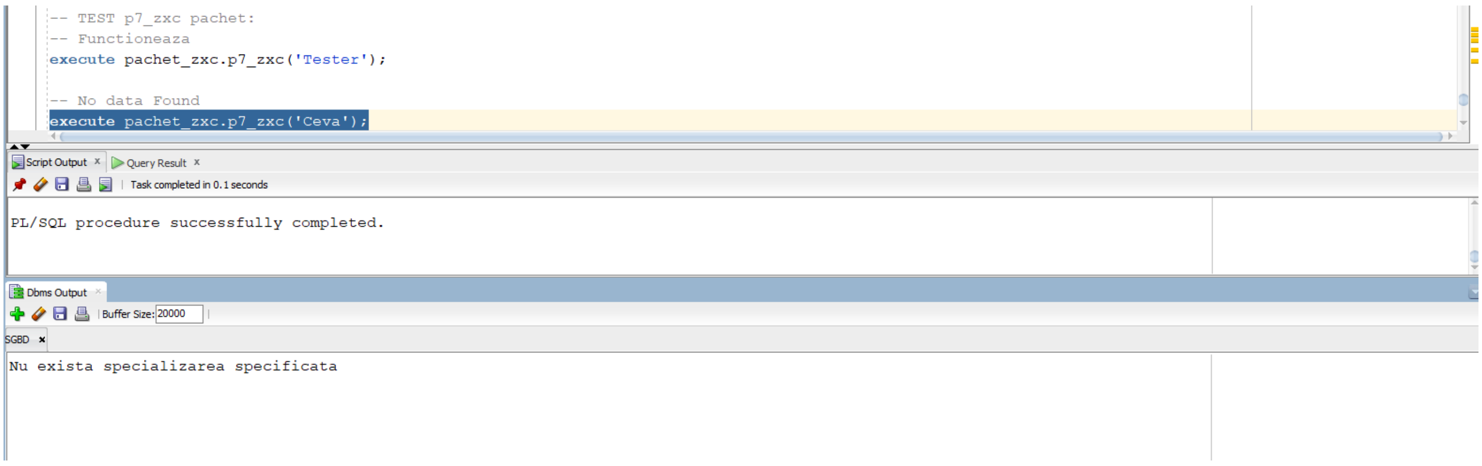


```
-- TEST p7_zxc pachet:
-- Functioneaza
execute pachet_zxc.p7_zxc('Tester');
```

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

```
Specializarea: Tester
----------------------------------
Poison Andra din echipa 10
Mihai Moromete din echipa 5
Levi Ackerman din echipa 15
Dumitru Ion din echipa 25
Silva Specter din echipa 35
```

Subprogramul 7 apelat din pachet: Functioneaza

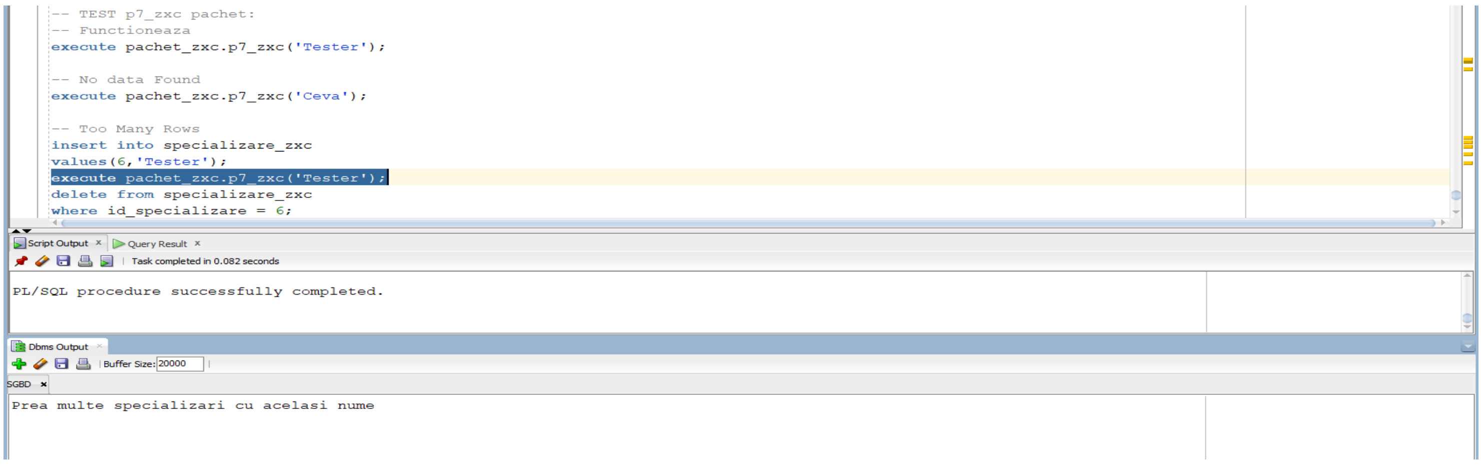


```
-- TEST p7_zxc pachet:
-- Functioneaza
execute pachet_zxc.p7_zxc('Tester');

-- No data Found
execute pachet_zxc.p7_zxc('Ceva');
```

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

```
Nu exista specializarea specificata
```

Subprogramul 7 apelat din pachet: No-Data-Found

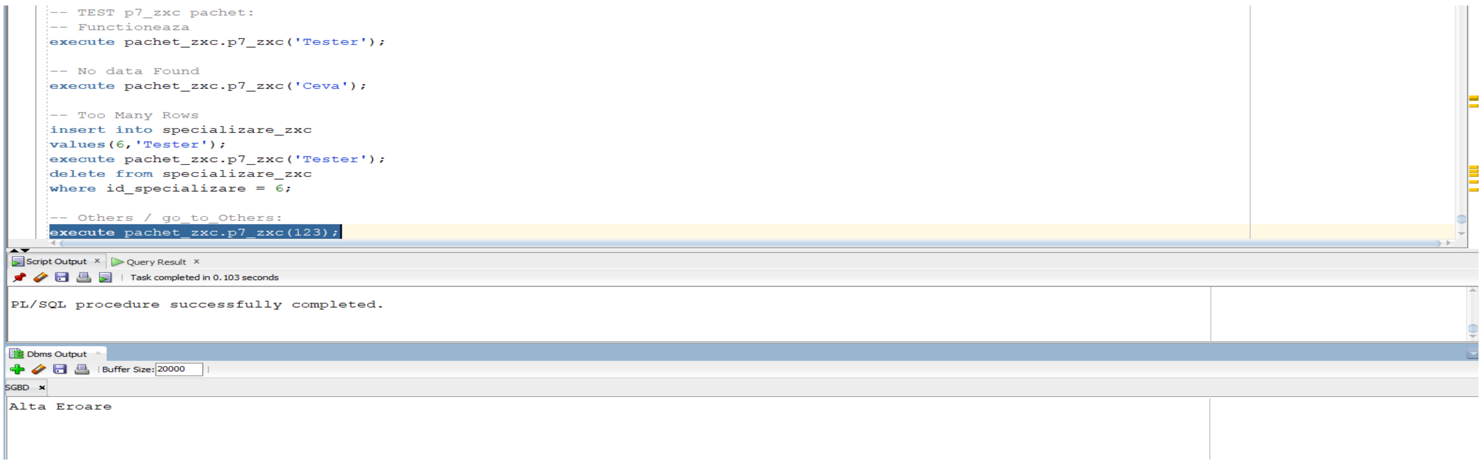


```
-- TEST p7_zxc pachet:
-- Functioneaza
execute pachet_zxc.p7_zxc('Tester');

-- No data Found
execute pachet_zxc.p7_zxc('Ceva');

-- Too Many Rows
insert into specializare_zxc
values(6,'Tester');
execute pachet_zxc.p7_zxc('Tester');
delete from specializare_zxc
where id_specializare = 6;
```

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

```
Prea multe specializari cu acelasi nume
```

Subprogramul 7 apelat din pachet: Too-Many-Rows

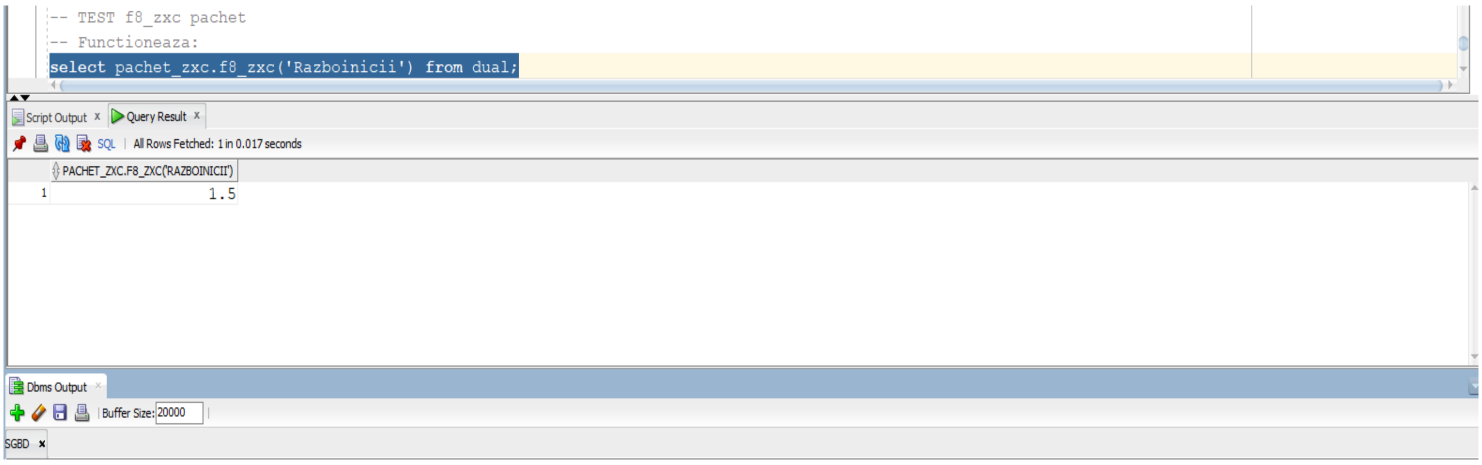


```
-- TEST p7_zxc pachet:
-- Functioneaza
execute pachet_zxc.p7_zxc('Tester');

-- No data Found
execute pachet_zxc.p7_zxc('Ceva');

-- Too Many Rows
insert into specializare_zxc
values(6,'Tester');
execute pachet_zxc.p7_zxc('Tester');
delete from specializare_zxc
where id_specializare = 6;

-- Others / go_to_Others:
execute pachet_zxc.p7_zxc(123);
```

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

```
Alta Eroare
```

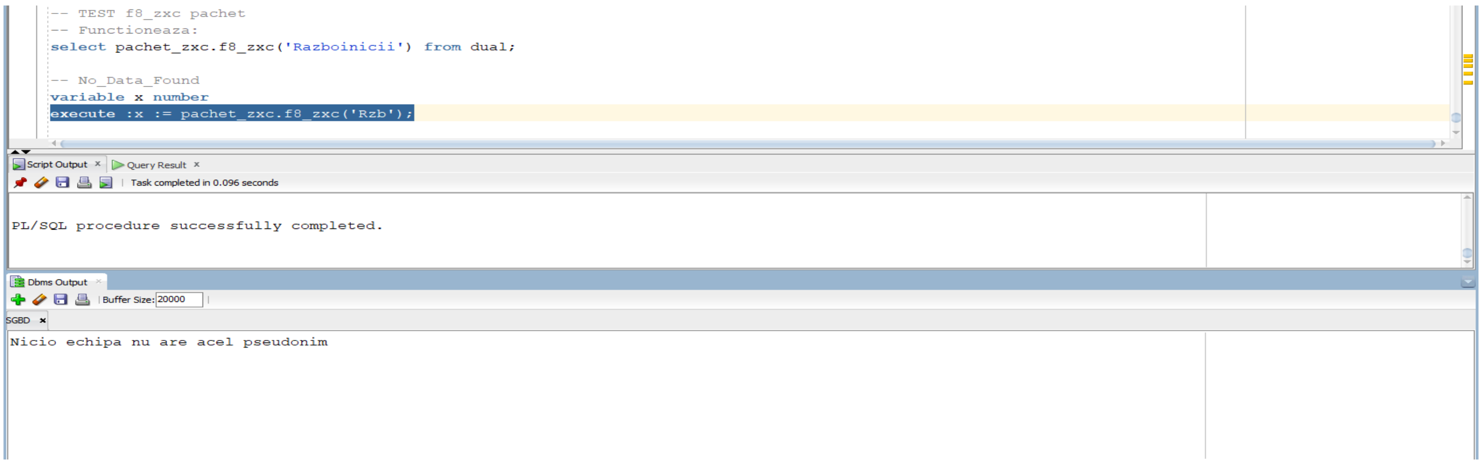
Subprogramul 7 apelat din pachet: Others



```
-- TEST f8_zxc pachet
-- Functioneaza:
select pachet_zxc.f8_zxc('Razboinicii') from dual;
```

```
  PACHET_ZXC.F8_ZXC('RAZBOINICII')
1                              1.5
```

Subprogramul 8 apelat din pachet: Functioneaza



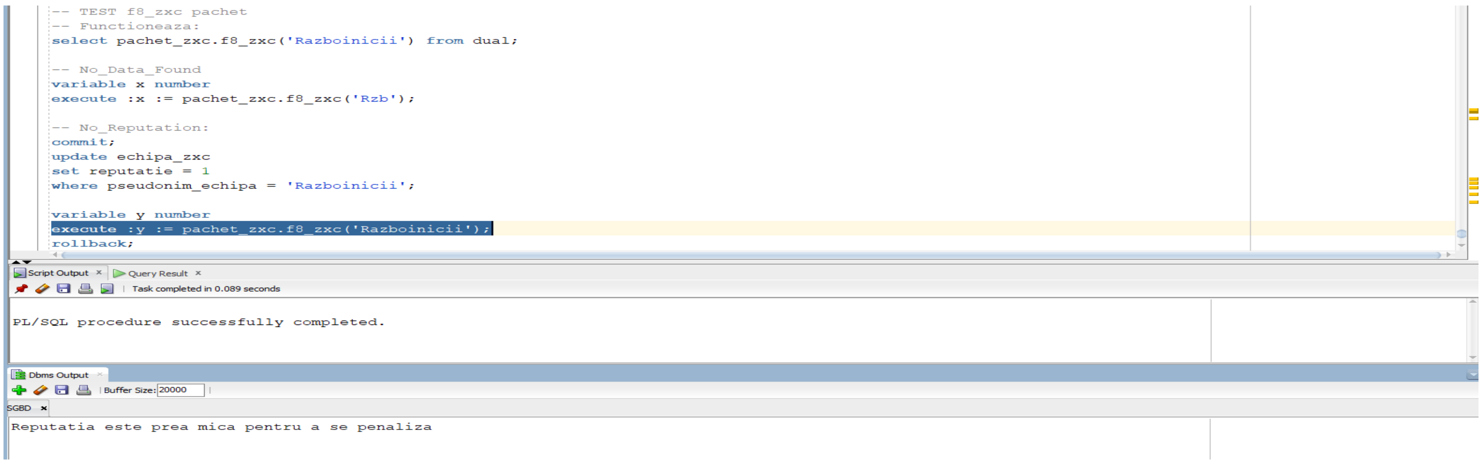
```
-- TEST f8_zxc pachet
-- Functioneaza:
select pachet_zxc.f8_zxc('Razboinicii') from dual;

-- No_Data_Found
variable x number
execute :x := pachet_zxc.f8_zxc('Rzb');
```

Task completed in 0.096 seconds

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

```
Nicio echipa nu are acel pseudonim
```

Subprogramul 8 apelat din pachet: No-Data-Found



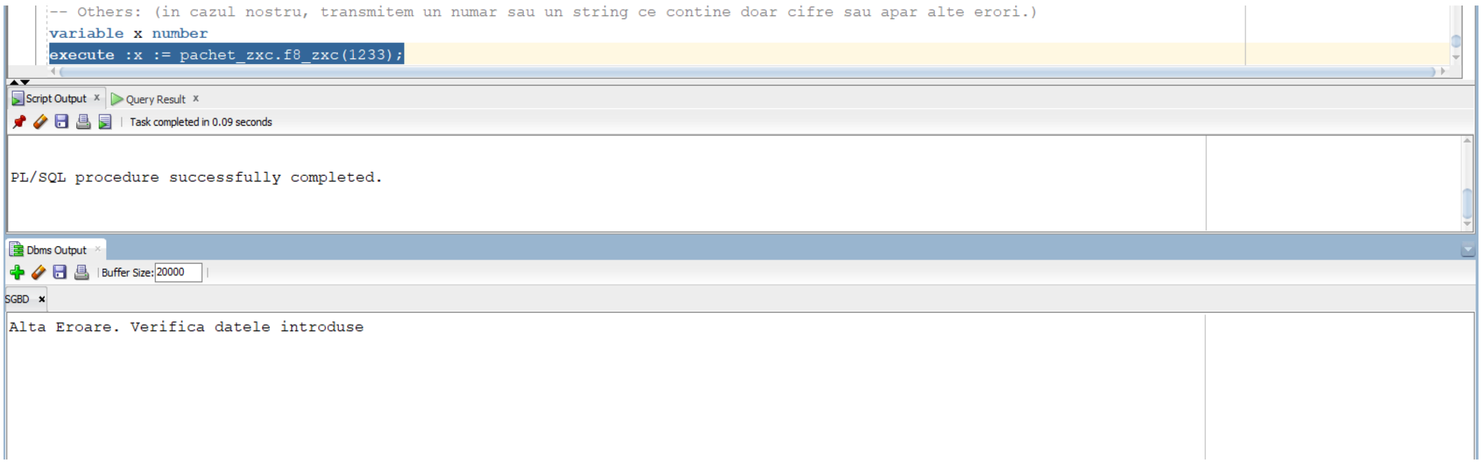
```
-- TEST f8_zxc pachet
-- Functioneaza:
select pachet_zxc.f8_zxc('Razboinicii') from dual;

-- No_Data_Found
variable x number
execute :x := pachet_zxc.f8_zxc('Rzb');

-- No_Reputation:
commit;
update echipa_zxc
set reputatie = 1
where pseudonim_echipa = 'Razboinicii';

variable y number
execute :y := pachet_zxc.f8_zxc('Razboinicii');
rollback;
```

Task completed in 0.089 seconds

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

```
Reputatia este prea mica pentru a se penaliza
```

Subprogramul 8 apelat din pachet: No-Reputation



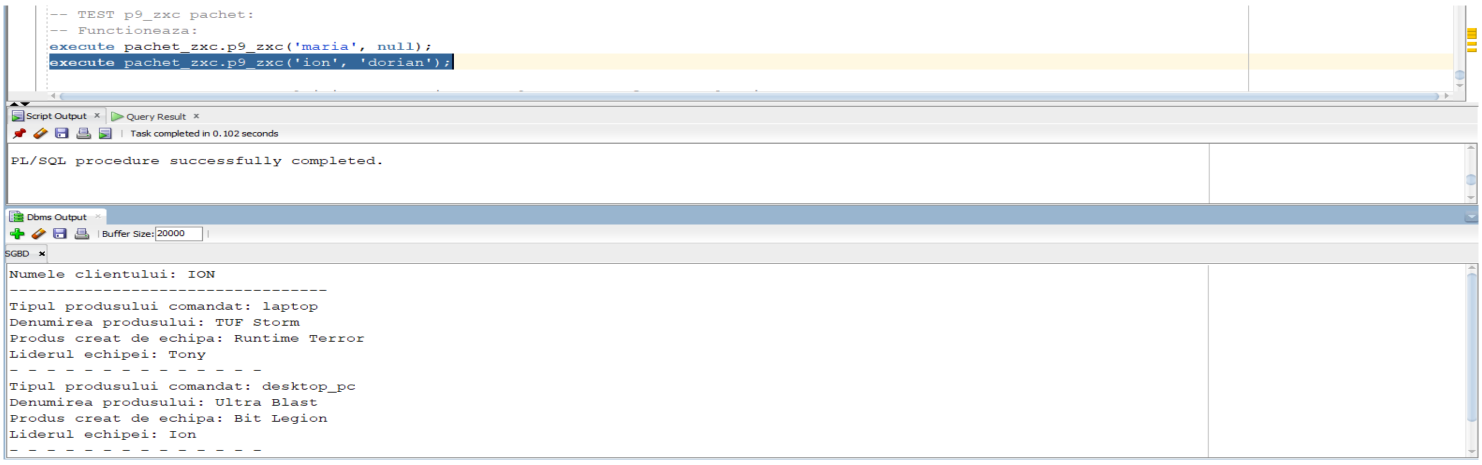
```
-- Others: (in cazul nostru, transmitem un numar sau un string ce contine doar cifre sau apar alte erori.)
variable x number
execute :x := pachet_zxc.f8_zxc(1233);
```

Task completed in 0.09 seconds

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

```
Alta Eroare. Verifica datele introduse
```

Subprogramul 8 apelat din pachet: Others



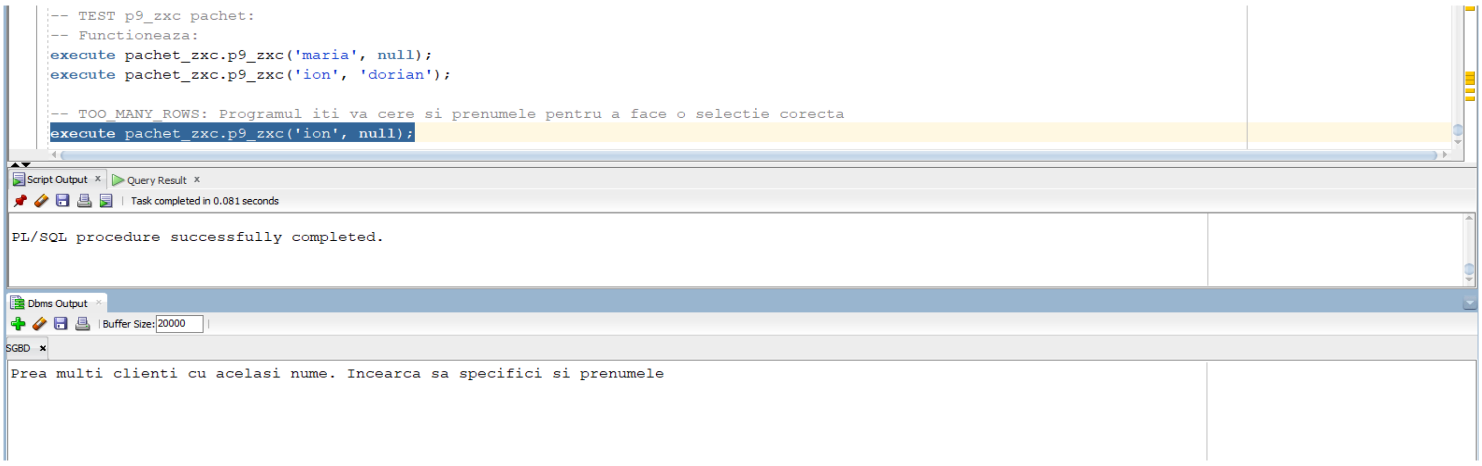
```
-- TEST p9_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p9_zxc('maria', null);
execute pachet_zxc.p9_zxc('ion', 'dorian');
```

Task completed in 0.102 seconds

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

```
Numele clientului: ION
----------------------------------
Tipul produsului comandat: laptop
Denumirea produsului: TUF Storm
Produs creat de echipa: Runtime Terror
Liderul echipei: Tony
- - - - - - - - - - - - - - -
Tipul produsului comandat: desktop_pc
Denumirea produsului: Ultra Blast
Produs creat de echipa: Bit Legion
Liderul echipei: Ion
- - - - - - - - - - - - - - -
```

Subprogramul 9 apelat din pachet: Functioneaza

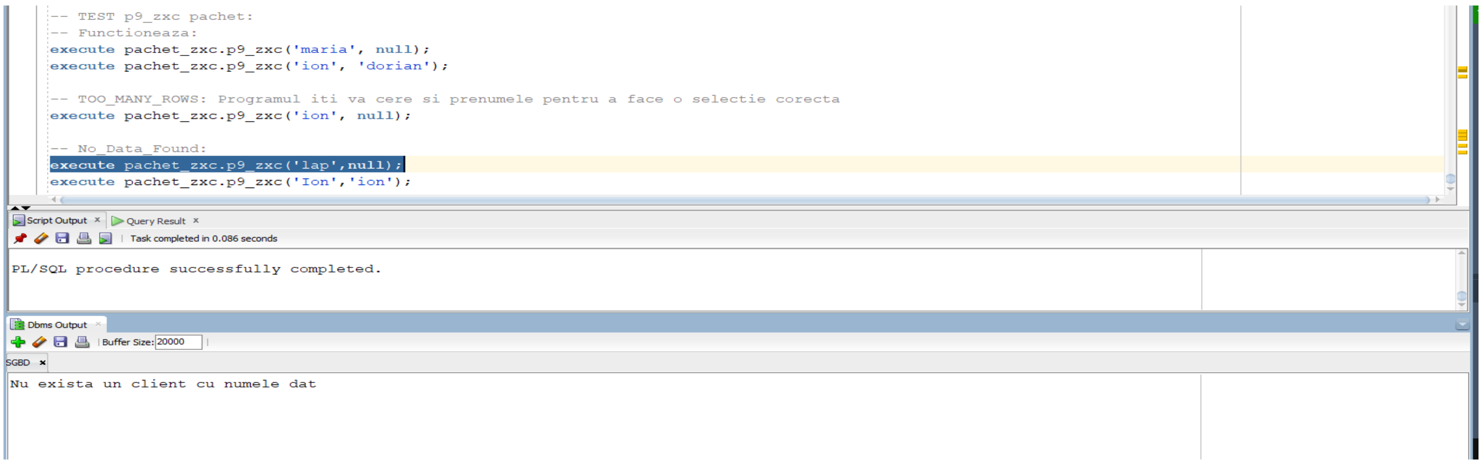


```
-- TEST p9_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p9_zxc('maria', null);
execute pachet_zxc.p9_zxc('ion', 'dorian');

-- TOO_MANY_ROWS: Programul iti va cere si prenumele pentru a face o selectie corecta
execute pachet_zxc.p9_zxc('ion', null);
```

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

```
Prea multi clienti cu acelasi nume. Incearca sa specifici si prenumele
```

Subprogramul 9 apelat din pachet: Too-Many-Rows

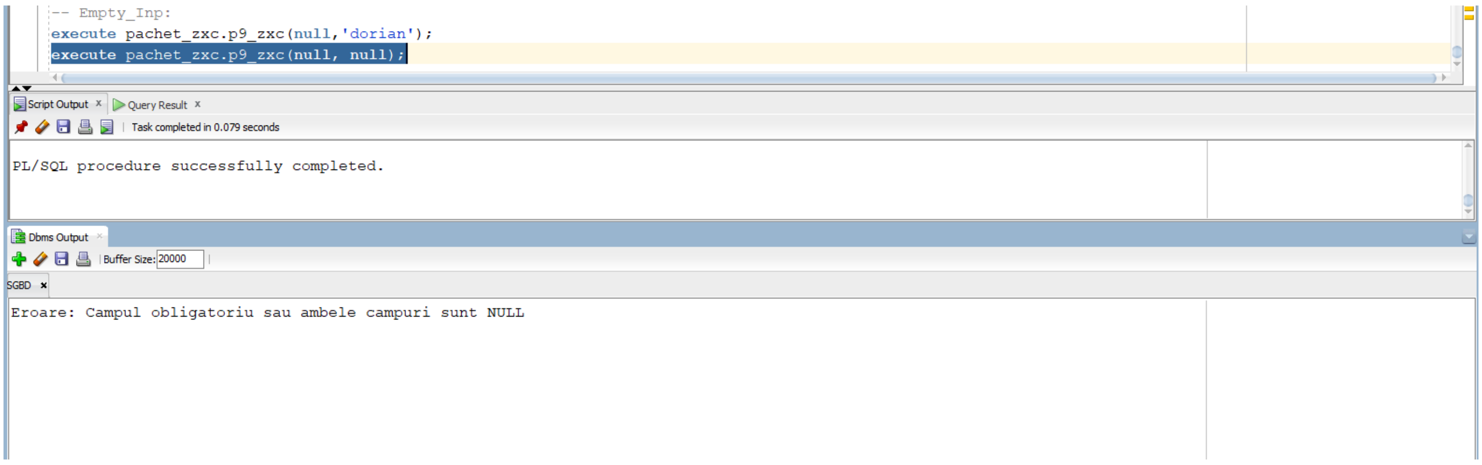


```
-- TEST p9_zxc pachet:
-- Functioneaza:
execute pachet_zxc.p9_zxc('maria', null);
execute pachet_zxc.p9_zxc('ion', 'dorian');

-- TOO_MANY_ROWS: Programul iti va cere si prenumele pentru a face o selectie corecta
execute pachet_zxc.p9_zxc('ion', null);

-- No_Data_Found:
execute pachet_zxc.p9_zxc('lap',null);
execute pachet_zxc.p9_zxc('Ion','ion');
```

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

```
Nu exista un client cu numele dat
```

Subprogramul 9 apelat din pachet: No-Data-Found

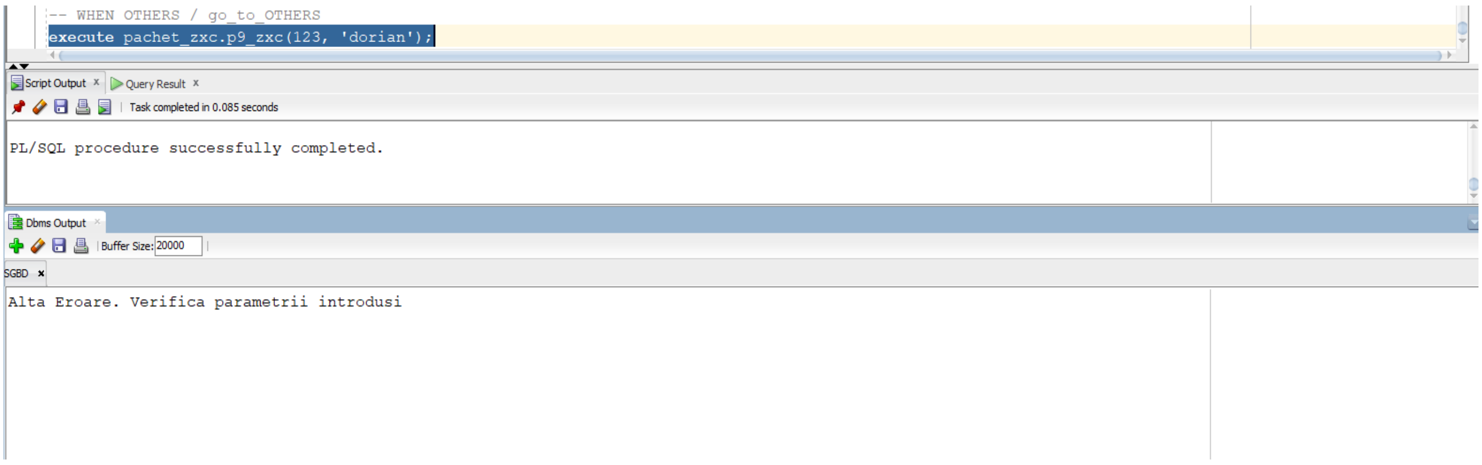


```
-- Empty_Inp:
execute pachet_zxc.p9_zxc(null,'dorian');
execute pachet_zxc.p9_zxc(null, null);
```

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

```
Eroare: Campul obligatoriu sau ambele campuri sunt NULL
```

Subprogramul 9 apelat din pachet: Empty-Input

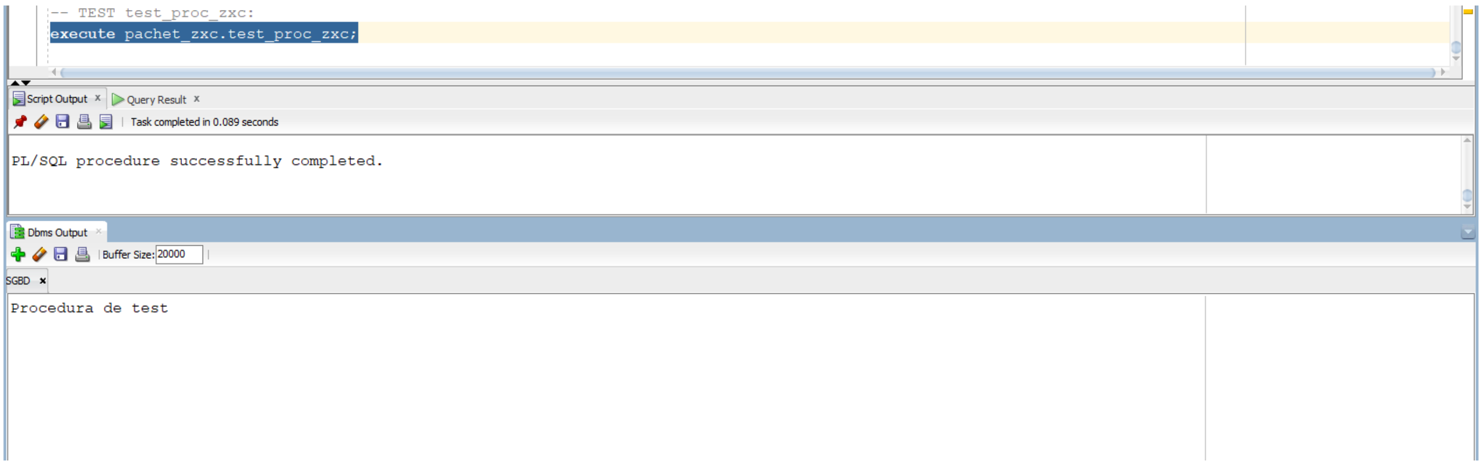


```
-- WHEN OTHERS / go_to_OTHERS
execute pachet_zxc.p9_zxc(123, 'dorian');
```

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

```
Alta Eroare. Verifica parametrii introdusi
```

Subprogramul 9 apelat din pachet: Others



```
-- TEST test_proc_zxc:
execute pachet_zxc.test_proc_zxc;
```

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

```
Procedura de test
```

Procedura de test: apel din pachet