

Gestiunea campionatelor de fotbal masculin

Nume: Stanciu Sergiu-Nicolas

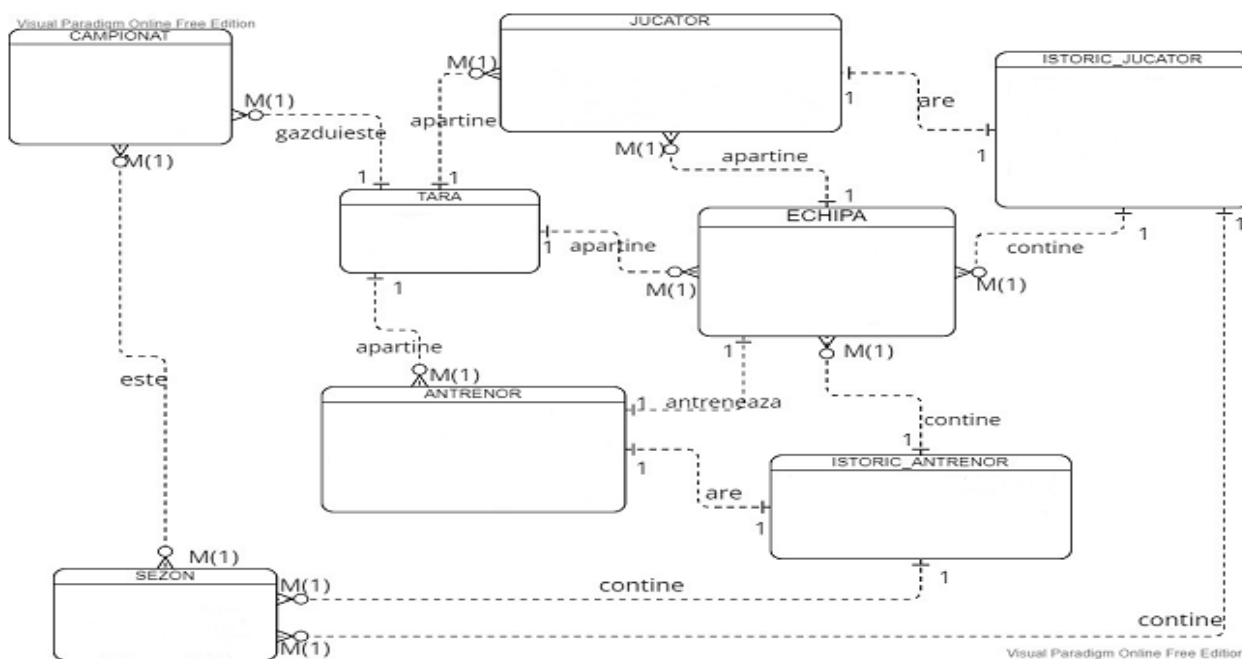
Grupa: 242

An: II, informatica

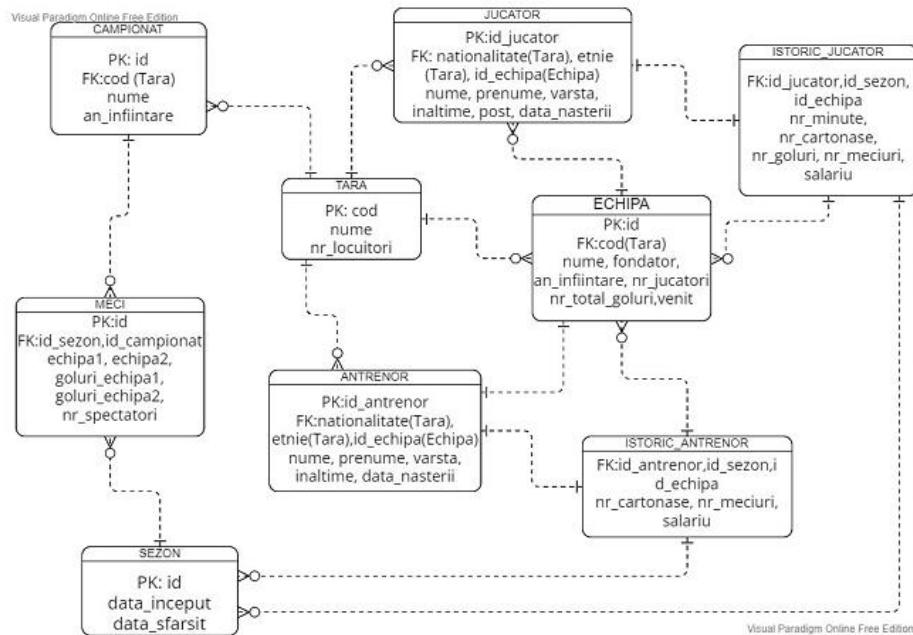
1. Prezentați pe scurt baza de date (utilitatea ei)

Baza de date pe care urmează să o prezint conține informații referitoare la meciurile de fotbal masculin care au loc în întreaga lume. Meciurile de fotbal au loc între două echipe, iar amândouă echipe trebuie să aparțină din același campionat – ca urmare nu există meciuri între echipe din campionate diferite. Meciurile țin evidența la numărul de goluri date de fiecare echipă. Fiecare echipă are un număr de jucători (care nu e fix) dar și un antrenor. O echipă poate avea mai mulți jucători, dar un jucător poate juca doar pentru o singură echipă sau chiar nici una. La fel și pentru antrenor: o echipă poate avea un singur antrenor, iar antrenorii pot antrena doar la o singură echipă sau chiar nici una. Fiecare jucător sau antrenor va avea date precum: vârsta, înălțime, naționalitate, etnie, clubul de la care fac parte. În plus, mai avem și tabela pentru sezon care poate face distincția dintre două meciuri la fel dar din sezoane diferite: două echipe pot juca una împotriva celeilalte în sezoane diferite (poate chiar cu scor diferit). Ele mai pot ajuta cu istoricul jucătorilor sau antrenorului: reținem fiecare echipă la care a jucat fiecare jucător în fiecare sezon (un jucător nu poate juca la două echipe în același sezon). La fel se aplică și pentru antrenor.

2. Realizați diagrama entitate-relație (ERD)



3. Pornind de la diagrama entitate-relatie realizati diagrama conceptuala a modelului propus, integrand toate attributele necesare.



4. Implementati in Oracle diagrama conceptuala realizata: definiti toate tabelele, implementand toate constrangerile de integritate necesare (chei primare, cheile externe etc).

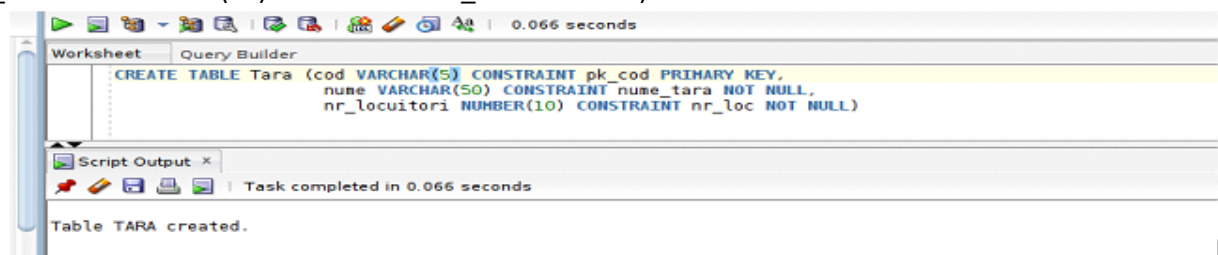
TARA

CREATE TABLE Tara

(cod VARCHAR(5) CONSTRAINT pk_cod PRIMARY KEY,

nume VARCHAR(50) CONSTRAINT nume_tara NOT NULL,

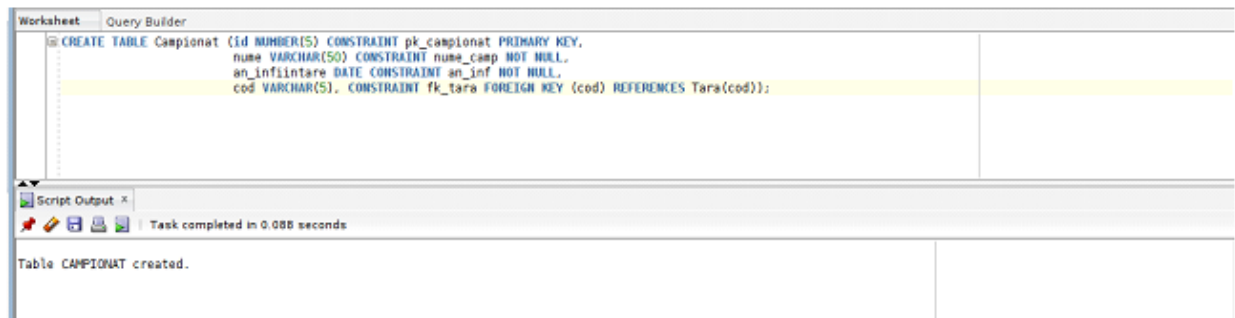
nr_locuitori NUMBER(10) CONSTRAINT nr_loc NOT NULL)



CAMPIONAT

CREATE TABLE Campionat

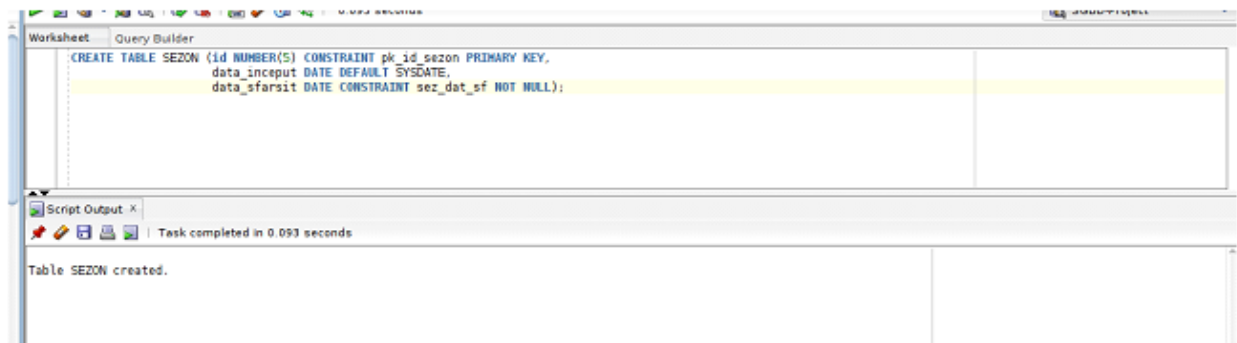
```
(id NUMBER(5) CONSTRAINT pk_campionat PRIMARY KEY,  
    nume VARCHAR(50) CONSTRAINT nume_camp NOT NULL,  
    an_infiintare DATE CONSTRAINT an_inf NOT NULL,  
    cod VARCHAR(5), CONSTRAINT fk_tara FOREIGN KEY (cod) REFERENCES  
    Tara(cod));
```



SEZON

CREATE TABLE SEZON (id NUMBER(5) CONSTRAINT pk_id_sezon PRIMARY KEY,

```
    data_inceput DATE DEFAULT SYSDATE,  
    data_sfarsit DATE CONSTRAINT sez_dat_sf NOT NULL);
```



ECHIPA

CREATE TABLE Echipa

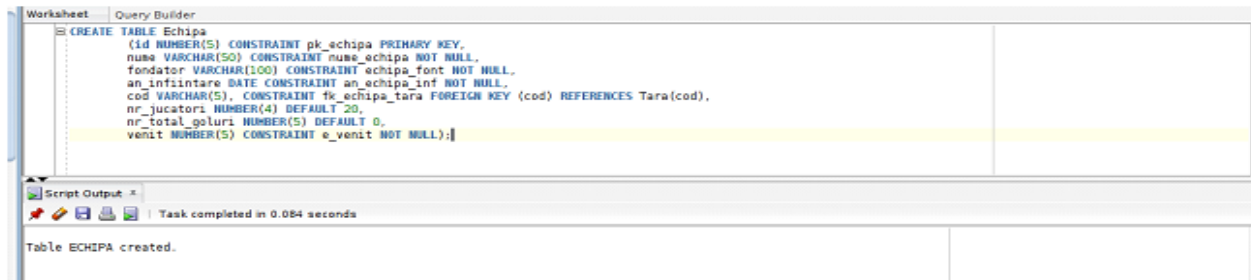
```
(id NUMBER(5) CONSTRAINT pk_echipa PRIMARY KEY,  
    nume VARCHAR(50) CONSTRAINT nume_echipa NOT NULL,  
    fondator VARCHAR(100) CONSTRAINT echipa_font NOT NULL,  
    an_infiintare DATE CONSTRAINT an_echipa_inf NOT NULL,
```

cod VARCHAR(5), CONSTRAINT fk_echipa_tara FOREIGN KEY (cod) REFERENCES
Tara(cod),

nr_jucatori NUMBER(4) DEFAULT 20,

nr_total_goluri NUMBER(5) DEFAULT 0,

venit NUMBER(5) CONSTRAINT e_venit NOT NULL);



JUCATOR

CREATE TABLE JUCATOR

(id_jucator NUMBER(5) CONSTRAINT pk_jucator PRIMARY KEY,

nume VARCHAR(50) CONSTRAINT nume_jucator NOT NULL,

prenume VARCHAR(50) CONSTRAINT prenume_jucator NOT NULL,

nationalitate VARCHAR(5), CONSTRAINT fk_juc_nat FOREIGN KEY (nationalitate)
REFERENCES Tara(cod),

etnie VARCHAR(5), CONSTRAINT fk_etnie FOREIGN KEY (etnie) REFERENCES Tara(cod),

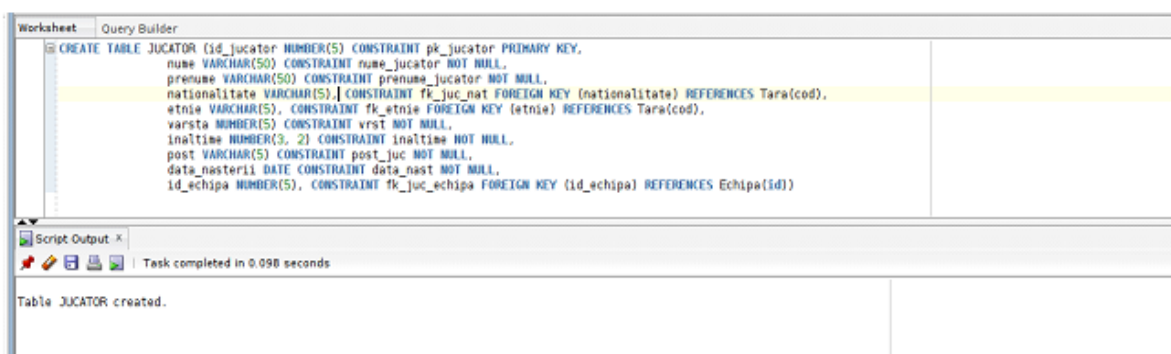
varsta NUMBER(5) CONSTRAINT vrst NOT NULL,

inaltime NUMBER(3, 2) CONSTRAINT inaltime NOT NULL,

post VARCHAR(5) CONSTRAINT post_juc NOT NULL,

data_nasterii DATE CONSTRAINT data_nast NOT NULL,

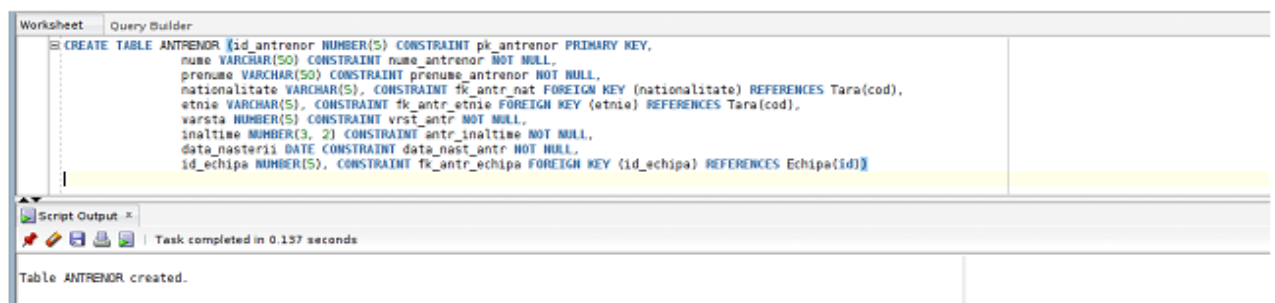
id_echipa NUMBER(5), CONSTRAINT fk_juc_echipa FOREIGN KEY (id_echipa) REFERENCES Echipa(id))



ANTRENOR

CREATE TABLE ANTRENOR

(id_antrenor NUMBER(5) CONSTRAINT pk_antrenor PRIMARY KEY,
nume VARCHAR(50) CONSTRAINT nume_antrenor NOT NULL,
prenume VARCHAR(50) CONSTRAINT prenume_antrenor NOT NULL,
nationalitate VARCHAR(5), CONSTRAINT fk_antr_nat FOREIGN KEY (nationalitate) REFERENCES Tara(cod),
etnie VARCHAR(5), CONSTRAINT fk_antr_etnie FOREIGN KEY (etnie) REFERENCES Tara(cod),
varsta NUMBER(5) CONSTRAINT vrst_antr NOT NULL,
inaltime NUMBER(3, 2) CONSTRAINT antr_inaltime NOT NULL,
data_nasterii DATE CONSTRAINT data_nast_antr NOT NULL,
id echipa NUMBER(5), CONSTRAINT fk_antr echipa FOREIGN KEY (id echipa) REFERENCES Echipa(id))



MECI

CREATE TABLE MECI

(id NUMBER(5) CONSTRAINT pk_meci PRIMARY KEY,
echipa1 NUMBER(5), CONSTRAINT fk_echipa1 FOREIGN KEY (echipa1) REFERENCES Echipa(id),
echipa2 NUMBER(5), CONSTRAINT fk_echipa2 FOREIGN KEY (echipa2) REFERENCES Echipa(id),
goluri_echipa1 NUMBER(2) DEFAULT 0,
goluri_echipa2 NUMBER(2) DEFAULT 0,
nr_spectatori NUMBER(5),
id_sezon NUMBER(5), CONSTRAINT fk_meci_sezon FOREIGN KEY (id_sezon) REFERENCES Sezon(id),
id_campionat NUMBER(5), CONSTRAINT fk_meci_campionat FOREIGN KEY (id_campionat) REFERENCES Campionat(id));



ISTORIC_JUCATOR

CREATE TABLE ISTORIC_JUCATOR

(id_jucator NUMBER(5), CONSTRAINT fk_ist_juc FOREIGN KEY (id_jucator) REFERENCES Jucator(id_jucator),

nr_minute NUMBER(5) DEFAULT 0,

nr_cartonase NUMBER(5) DEFAULT 0,

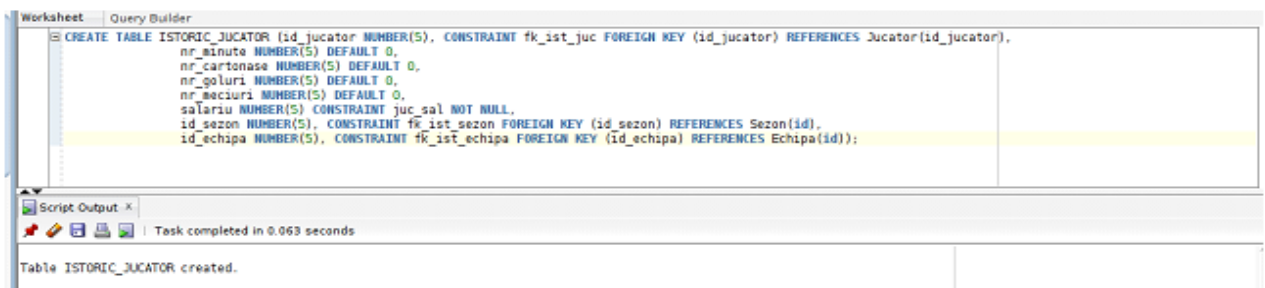
nr_goluri NUMBER(5) DEFAULT 0,

nr_meciuri NUMBER(5) DEFAULT 0,

salariu NUMBER(5) CONSTRAINT juc_sal NOT NULL,

id_sezon NUMBER(5), CONSTRAINT fk_ist_sezon FOREIGN KEY (id_sezon) REFERENCES Sezon(id),

id_echipa NUMBER(5), CONSTRAINT fk_ist_echipa FOREIGN KEY (id_echipa) REFERENCES Echipa(id));



ISTORIC_ANTRENOR

CREATE TABLE ISTORIC_ANTRENOR

(id_antrenor NUMBER(5), CONSTRAINT fk_ist_antr FOREIGN KEY (id_antrenor) REFERENCES Antrenor(id_antrenor),

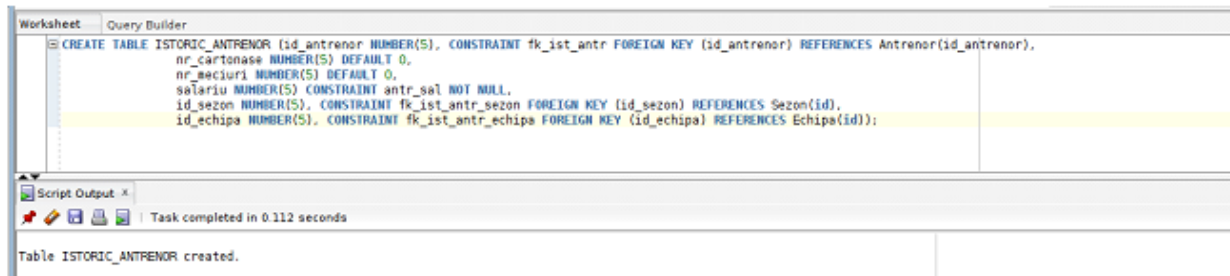
nr_cartonase NUMBER(5) DEFAULT 0,

nr_meciuri NUMBER(5) DEFAULT 0,

salariu NUMBER(5) CONSTRAINT antr_sal NOT NULL,

id_sezon NUMBER(5), CONSTRAINT fk_ist_antr_sezon FOREIGN KEY (id_sezon) REFERENCES Sezon(id),

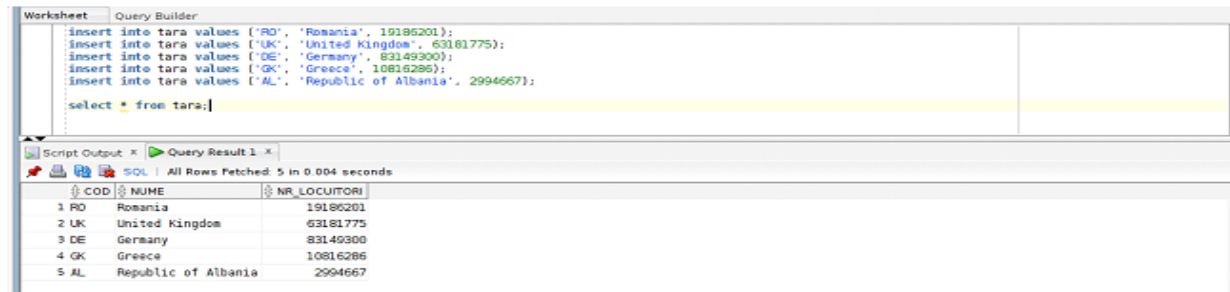
id echipa NUMBER(5), CONSTRAINT fk_ist_antr echipa FOREIGN KEY (id echipa) REFERENCES Echipa(id));



5. Adaugati informatii coerente in tabelele create (minim 5 inregistrari pentru fiecare entitate independent; minim 10 inregistrari pentru tabela asociativa).

Tara

- insert into tara values ('RO', 'Romania', 19186201);
- insert into tara values ('UK', 'United Kingdom', 63181775);
- insert into tara values ('DE', 'Germany', 83149300);
- insert into tara values ('GK', 'Greece', 10816286);
- insert into tara values ('AL', 'Republic of Albania', 2994667);



Echipa

insert into echipa values (1, 'Manchester United', 'Glazer Family', TO_DATE('25-05-1878', 'dd-mm-yyyy'), 'UK', 26, 25768, 106.6);

- insert into echipa values (2, 'Steaua Bucuresti', 'Valeriu Argaseala', TO_DATE('07-06-1947', 'dd-mm-yyyy'), 'RO', 31, 4680, 1.23);
- insert into echipa values (3, 'SC Otelul Galati', 'Primaria Galati', TO_DATE('10-09-1964', 'dd-mm-yyyy'), 'RO', 25, 10467, 2.34);
- insert into echipa values (4, 'PAOK Salonic', 'Ivan Ignatievici Savvidi', TO_DATE('03-01-1926', 'dd-mm-yyyy'), 'GK', 27, 19672, 10.3);
- insert into echipa values (5, 'Chelsea FC', 'Todd Boehly', TO_DATE('10-03-1905', 'dd-mm-yyyy'), 'UK', 32, 45670, 564);

Worksheet

Query Builder

```
insert into echipa values (1, 'Manchester United', 'Glazer Family', TO_DATE('25-05-1878', 'dd-mm-yyyy'), 'UK', 26, 25768, 106.6);
insert into echipa values (2, 'Steaua Bucuresti', 'Valeriu Argaseala', TO_DATE('07-06-1947', 'dd-mm-yyyy'), 'RO', 31, 4680, 1.23);
insert into echipa values (3, 'SC Otelul Galati', 'Primaria Galati', TO_DATE('10-09-1964', 'dd-mm-yyyy'), 'RO', 25, 10467, 2.34);
insert into echipa values (4, 'PAOK Salonic', 'Ivan Ignatievici Savvidi', TO_DATE('03-01-1926', 'dd-mm-yyyy'), 'GK', 27, 19672, 10.3);
insert into echipa values (5, 'Chelsea FC', 'Todd Boehly', TO_DATE('10-03-1905', 'dd-mm-yyyy'), 'UK', 32, 45670, 564);

select * from echipa;
```

Script Output

Query Result

Query Result 1

SQL | All Rows Fetched: 5 in 0.005 seconds

ID	NUME	FONDATOR	AN_INFINTARE	COD	NR_JUCATORI	NR_TOTAL_GOLURI	VENIT
1	Manchester United	Glazer Family	25-MAY-78	UK	26	25768	107
2	Steaua Bucuresti	Valeriu Argaseala	07-JUN-47	RO	31	4680	1
3	SC Otelul Galati	Primaria Galati	10-SEP-64	RO	25	10467	2
4	PAOK Salonic	Ivan Ignatievici Savvidi	03-JAN-26	GK	27	19672	10
5	Chelsea FC	Todd Boehly	10-MAR-05	UK	32	45670	564

Antrenor

- insert into antrenor values (1, 'Radoi', 'Mirel', 'RO', 'RO', 28, 1.78, TO_DATE('10-09-1994', 'dd-mm-yyyy'), 2);
- insert into antrenor values (2, 'Grecu', 'Ionut', 'RO', 'GK', 45, 1.80, TO_DATE('02-08-1977', 'dd-mm-yyyy'), 1);
- insert into antrenor values (3, 'Dontu', 'Marian', 'RO', 'RO', 34, 1.76, TO_DATE('26-10-1988', 'dd-mm-yyyy'), 3);
- insert into antrenor values (4, 'Klopp', 'Jurgen', 'DE', 'DE', 56, 1.85, TO_DATE('04-02-1966', 'dd-mm-yyyy'), 5);
- insert into antrenor values (5, 'Maximilian', 'Josh', 'UK', 'RO', 20, 1.90, TO_DATE('20-04-2002', 'dd-mm-yyyy'), 4);

Worksheet

Query Builder

```

1 insert into antrenor values (1, 'Radoi', 'Mirel', 'RO', 'RO', 28, 1.78, TO_DATE('10-09-1994', 'dd-mm-yyyy'), 2);
2 insert into antrenor values (2, 'Grecu', 'Ionut', 'RO', 'GK', 45, 1.80, TO_DATE('02-08-1977', 'dd-mm-yyyy'), 1);
3 insert into antrenor values (3, 'Dontu', 'Marian', 'RO', 'RO', 34, 1.76, TO_DATE('26-10-1988', 'dd-mm-yyyy'), 3);
4 insert into antrenor values (4, 'Klopp', 'Jurgen', 'DE', 'DE', 56, 1.85, TO_DATE('04-02-1966', 'dd-mm-yyyy'), 5);
5 insert into antrenor values (5, 'Maximilian', 'Josh', 'UK', 'RO', 20, 1.90, TO_DATE('20-04-2002', 'dd-mm-yyyy'), 4);
6
7 select * from antrenor;

```

Script Output x

Query Result x

Query Result 1 x

Query Result 2 x

Query Result 3 x

SQL

All Rows Fetched 5 in 0.003 seconds

ID_ANTRENOR	NUME	PRENUME	NATIONALITATE	ETNIE	VARSTA	INALTIME	DATA_NASTERII	ID_ECHIPA
1	Radoi	Mirel	RO	RO	28	1.78	10-SEP-94	2
2	Grecu	Ionut	RO	GK	45	1.80	02-AUG-77	1
3	Dontu	Marian	RO	RO	34	1.76	26-OCT-88	3
4	Klopp	Jurgen	DE	DE	56	1.85	04-FEB-66	5
5	Maximilian	Josh	UK	RO	20	1.90	20-APR-02	4

Jucator

- insert into jucator values (1, 'Coman', 'Florinel', 'RO', 'RO', 24, 1.82, 'LW', TO_DATE('10-04-1998', 'dd-mm-yyyy'), 2);
- insert into jucator values (2, 'Partenie', 'Cosmin', 'UK', 'RO', 22, 1.75, 'CAM', TO_DATE('06-06-2000', 'dd-mm-yyyy'), 3);
- insert into jucator values (3, 'Rudiger', 'Antonio', 'DE', 'DE', 29, 1.90, 'CB', TO_DATE('03-03-1993', 'dd-mm-yyyy'), 5);

- insert into jucator values (4, 'Broja', 'Armando', 'AL', 'AL', 21, 1.91, 'RW', TO_DATE('10-09-2001', 'dd-mm-yyyy'), 5);
- insert into jucator values (5, 'Cristiano', 'Ronaldo', 'GK', 'UK', 37, 1.87, 'ST', TO_DATE('05-02-1985', 'dd-mm-yyyy'), 1);

Worksheet | Query Builder

```

insert into jucator values (1, 'Coman', 'Florinel', 'RO', 'RO', 24, 1.82, 'LW', TO_DATE('10-04-1998', 'dd-mm-yyyy'), 2);
insert into jucator values (2, 'Partenie', 'Cosmin', 'UK', 'RO', 22, 1.75, 'CAM', TO_DATE('06-06-2000', 'dd-mm-yyyy'), 3);
insert into jucator values (3, 'Rudiger', 'Antonio', 'DE', 'DE', 29, 1.90, 'CB', TO_DATE('03-03-1993', 'dd-mm-yyyy'), 5);
insert into jucator values (4, 'Broja', 'Armando', 'AL', 'AL', 21, 1.91, 'RW', TO_DATE('10-09-2001', 'dd-mm-yyyy'), 5);
insert into jucator values (5, 'Cristiano', 'Ronaldo', 'GK', 'UK', 37, 1.87, 'ST', TO_DATE('05-02-1985', 'dd-mm-yyyy'), 1);

select * from jucator;

```

Script Output x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x

SQL | All Rows Fetched: 5 in 0.109 seconds

ID_JUCATOR	NUME	PRENUME	NATIONALITATE	ETNIE	VARSTA	INALTIME	POST	DATA_NASTERII	ID_ECHIPA
1	Coman	Florinel	RO	RO	24	1.82 LW	10-APR-98		2
2	Partenie	Cosmin	UK	RO	22	1.75 CAM	06-JUN-00		3
3	Rudiger	Antonio	DE	DE	29	1.9 CB	03-MAR-93		5
4	Broja	Armando	AL	AL	21	1.91 RW	10-SEP-01		5
5	Cristiano	Ronaldo	GK	UK	37	1.87 ST	05-FEB-85		1

Sezon

- insert into sezon values (2022, TO_DATE('07-10-2022', 'dd-mm-yyyy'), TO_DATE('27-05-2023', 'dd-mm-yyyy'));
- insert into sezon values (2021, TO_DATE('20-10-2022', 'dd-mm-yyyy'), TO_DATE('20-05-2023', 'dd-mm-yyyy'));
- insert into sezon values (2020, TO_DATE('28-09-2022', 'dd-mm-yyyy'), TO_DATE('04-06-2023', 'dd-mm-yyyy'));
- insert into sezon values (2019, TO_DATE('10-10-2022', 'dd-mm-yyyy'), TO_DATE('18-05-2023', 'dd-mm-yyyy'));
- insert into sezon values (2018, TO_DATE('14-10-2022', 'dd-mm-yyyy'), TO_DATE('05-06-2023', 'dd-mm-yyyy'));

Campionat

- insert into campionat values (1, 'Premier League', TO_DATE('20-02-1992', 'dd-mm-yyyy'), 'UK');
- insert into campionat values (2, 'FA Cup', TO_DATE('17-04-1871', 'dd-mm-yyyy'), 'UK');
- insert into campionat values (3, 'Liga I', TO_DATE('20-08-1909', 'dd-mm-yyyy'), 'RO');
- insert into campionat values (4, 'Super League Greece', TO_DATE('05-05-1927', 'dd-mm-yyyy'), 'GK');
- insert into campionat values (5, 'Super cupa Romaniei', TO_DATE('20-08-2003', 'dd-mm-yyyy'), 'RO');

Worksheet Query Builder

```

insert into campionat values (1, 'Premier League', TO_DATE('20-02-1992', 'dd-mm-yyyy'), 'UK');
insert into campionat values (2, 'FA Cup', TO_DATE('17-04-1871', 'dd-mm-yyyy'), 'UK');
insert into campionat values (3, 'Liga 1', TO_DATE('20-08-1999', 'dd-mm-yyyy'), 'RO');
insert into campionat values (4, 'Super League Greece', TO_DATE('05-05-1927', 'dd-mm-yyyy'), 'GR');
insert into campionat values (5, 'Super cupa Romaniei', TO_DATE('20-08-2003', 'dd-mm-yyyy'), 'RO');

select * from campionat;

```

Script Output Query Result Query Result 1 Query Result 2

SQL All Rows Fetched: 5 in 0.004 seconds

ID	NUME	AN_INFINTARE	COD
1	1 Premier League	20-FEB-92	UK
2	2 FA Cup	17-APR-71	UK
3	3 Liga 1	20-AUG-09	RO
4	4 Super League Greece	05-MAY-27	GR
5	5 Super cupa Romaniei	20-AUG-03	RO

Meci

- insert into meci values (1, 2, 3, 4, 0, 0.21, 2021, 5);
- insert into meci values (2, 2, 3, 2, 2, 2.3, 2022, 3);
- insert into meci values (3, 1, 2, 0, 1, 50, 2020, 2);
- insert into meci values (4, 4, 5, 0, 0, 23.1, 2018, 4);
- insert into meci values (5, 5, 4, 0, 1, 12.45, 2019, 2);
- insert into meci values (6, 4, 1, 1, 1, 8.56, 2021, 1);
- insert into meci values (7, 3, 2, 1, 1, 5.6, 2021, 5);
- insert into meci values (8, 1, 5, 2, 4, 20.45, 2020, 2);
- insert into meci values (9, 5, 1, 2, 2, 15.5, 2021, 1);
- insert into meci values (10, 1, 5, 3, 0, 8.9, 2019, 1);

Worksheet Query Builder

```

insert into meci values (1, 2, 3, 4, 0, 0.21, 2021, 5);
insert into meci values (2, 2, 3, 2, 2, 2.3, 2022, 3);
insert into meci values (3, 1, 2, 0, 1, 50, 2020, 2);
insert into meci values (4, 4, 5, 0, 0, 23.1, 2018, 4);
insert into meci values (5, 5, 4, 0, 1, 12.45, 2019, 2);
insert into meci values (6, 4, 1, 1, 1, 8.56, 2021, 1);
insert into meci values (7, 3, 2, 1, 1, 5.6, 2021, 5);
insert into meci values (8, 1, 5, 2, 4, 20.45, 2020, 2);
insert into meci values (9, 5, 1, 2, 2, 15.5, 2021, 1);
insert into meci values (10, 1, 5, 3, 0, 8.9, 2019, 1);

select * from meci;

```

Script Output Query Result 2 Query Result 3

SQL All Rows Fetched: 10 in 0.003 seconds

ID	ECHIPA1	ECHIPA2	GOLURI_ECHIPA1	GOLURI_ECHIPA2	NR_SPECULATORI	ID_SEZON	ID_CAMPIONAT
1	1	2	3	4	0	2021	5
2	2	2	3	2	2	2022	3
3	3	1	2	0	1	2020	2
4	4	4	5	0	0	2018	4
5	5	5	4	0	1	2019	2
6	6	4	1	1	1	2021	1
7	7	3	2	1	1	2021	5
8	8	1	5	2	4	2020	2
9	9	5	1	2	2	2021	1
10	10	1	5	3	0	2019	1

Istoric_jucator

- insert into istoric_jucator values (1, 450, 3, 23, 47, 3400, 2019, 2);
- insert into istoric_jucator values (1, 675, 5, 44, 45, 6050, 2021, 2);
- insert into istoric_jucator values (1, 334, 1, 34, 33, 5500, 2022, 3);
- insert into istoric_jucator values (2, 156, 0, 12, 10, 3000, 2018, 3);
- insert into istoric_jucator values (2, 120, 1, 8, 5, 8400, 2019, 4);

- insert into istoric_jucator values (3, 845, 6, 4, 46, 12000, 2020, 5);
- insert into istoric_jucator values (3, 685, 12, 8, 38, 11000, 2021, 5);
- insert into istoric_jucator values (5, 860, 2, 67, 55, 20600, 2018, 4);
- insert into istoric_jucator values (5, 1000, 5, 88, 60, 45000, 2020, 1);
- insert into istoric_jucator values (5, 950, 4, 55, 50, 55000, 2022, 1);

Worksheet Query Builder

```

1 insert into istoric_jucator values (1, 450, 3, 23, 47, 3400, 2019, 2);
2 insert into istoric_jucator values (1, 675, 5, 44, 45, 6050, 2021, 2);
3 insert into istoric_jucator values (1, 334, 1, 34, 33, 5500, 2022, 3);
4 insert into istoric_jucator values (2, 156, 0, 12, 10, 3000, 2018, 3);
5 insert into istoric_jucator values (2, 120, 1, 8, 5, 8400, 2019, 4);
6 insert into istoric_jucator values (3, 845, 6, 4, 46, 12000, 2020, 5);
7 insert into istoric_jucator values (3, 685, 12, 8, 38, 11000, 2021, 5);
8 insert into istoric_jucator values (5, 860, 2, 67, 55, 20600, 2018, 4);
9 insert into istoric_jucator values (5, 1000, 5, 88, 60, 45000, 2020, 1);
10 insert into istoric_jucator values (5, 950, 4, 55, 50, 55000, 2022, 1);
11
12 select * from istoric_jucator;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x

SQL All Rows Fetched: 10 in 0.011 seconds

	ID_JUCATOR	NR_MINUTE	NR_CARTONASE	NR_GOLURI	NR_MECIURI	SALARIU	ID_SEZON	ID_ECHIPA
1	1	450	3	23	47	3400	2019	2
2	1	675	5	44	45	6050	2021	2
3	1	334	1	34	33	5500	2022	3
4	2	156	0	12	10	3000	2018	3
5	2	120	1	8	5	8400	2019	4
6	3	845	6	4	46	12000	2020	5
7	3	685	12	8	38	11000	2021	5
8	5	860	2	67	55	20600	2018	4
9	5	1000	5	88	60	45000	2020	1
10	5	950	4	55	50	55000	2022	1

Istoric_antrenor

- insert into istoric_antrenor values (1, 0, 44, 6000, 2020, 4);
- insert into istoric_antrenor values (1, 1, 34, 8600, 2022, 4);
- insert into istoric_antrenor values (2, 1, 20, 2300, 2018, 2);
- insert into istoric_antrenor values (2, 0, 32, 4000, 2019, 2);
- insert into istoric_antrenor values (2, 0, 33, 4100, 2021, 3);
- insert into istoric_antrenor values (3, 3, 55, 6500, 2019, 1);
- insert into istoric_antrenor values (3, 3, 57, 8300, 2020, 5);
- insert into istoric_antrenor values (3, 5, 60, 8200, 2022, 1);
- insert into istoric_antrenor values (4, 0, 20, 3800, 2018, 4);
- insert into istoric_antrenor values (4, 1, 34, 5000, 2020, 5);

Worksheet

Query Builder

1

insert into istoric_antrenor values (1, 0, 44, 6000, 2020, 4);

2

insert into istoric_antrenor values (1, 1, 34, 8600, 2022, 4);

3

insert into istoric_antrenor values (2, 1, 20, 2300, 2018, 2);

4

insert into istoric_antrenor values (2, 0, 32, 4000, 2019, 2);

5

insert into istoric_antrenor values (2, 0, 33, 4100, 2021, 3);

6

insert into istoric_antrenor values (3, 3, 55, 6500, 2019, 1);

7

insert into istoric_antrenor values (3, 3, 57, 8300, 2020, 5);

8

insert into istoric_antrenor values (3, 5, 60, 8200, 2022, 1);

9

insert into istoric_antrenor values (4, 0, 20, 3800, 2018, 4);

10

insert into istoric_antrenor values (4, 1, 34, 5000, 2020, 5);

11

12

select * from istoric_antrenor;

Script Output x

Query Result 1 x

Query Result 3 x

Query Result 4 x

SQL

All Rows Fetched: 10 in 0.006 seconds

	ID_ANTRENOR	NR_CARTONASE	NR_MECIURI	SALARIU	ID_SEZON	ID_ECHIPA
1	1	0	44	6000	2020	4
2	1	1	34	8600	2022	4
3	2	1	20	2300	2018	2
4	2	0	32	4000	2019	2
5	2	0	33	4100	2021	3
6	3	3	55	6500	2019	1
7	3	3	57	8300	2020	5
8	3	5	60	8200	2022	1
9	4	0	20	3800	2018	4
10	4	1	34	5000	2020	5

6. Formulati in limbaj natural o problema care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze doua tipuri diferite de colectii studiate. Apelati subprogramul.

--Pentru fiecare echipa sa se afiseze numarul de jucatori inregistrati in baza de date si varsta medie a jucatorilor

--din acea echipa.

CREATE or REPLACE PROCEDURE procedure6_ssn IS

-- cream un tip nou de date care va retine numele echipei, numarul de jucatori din echipa respectiva care

-- sunt inregistrati in baza de date si varsta medie a acelor jucatori

TYPE tip_data echipa is RECORD

(nume_echipa echipa.nume%TYPE, numar_jucatori echipa.nr_jucatori%TYPE, varsta_medie jucator.varsta%TYPE);

-- toate datele pentru echipe vor fi retinute intr-un tablou imbricat

TYPE tab_echipe is TABLE of tip_data echipa;

-- vom folosi un vector pentru a retine numarul de jucatori inregistrati pentru fiecare echipa

TYPE jucatori echipa is VARRAY(8) of echipa.nr_jucatori%TYPE;

-- cu ajutorul unui cursor vom parcurge toate echipele

CURSOR e IS

SELECT id, nume from echipa;

```

t tab_echipe := tab_echipe();
type_e tip_data_echipa;
v jucatori_echipa := jucatori_echipa();

BEGIN

    FOR i in e LOOP

-- Vom salva datele pentru numele echipei, numarul de jucatori curenti la echipa si varsta medie a acestora

        SELECT AVG(varsta), COUNT(*)
        into type_e.varsta_medie, type_e.numar_jucatori
        from jucator j where j.id_echipa = i.id;

        v.EXTEND;

        SELECT COUNT(*) into v(v.last)
        from jucator j where j.id_echipa = i.id;

        type_e.ume_echipa := i.ume;

-- adaugam variabila in tabloul imbricat care cere tipul variabilei noastre

        t.EXTEND;

        t(t.LAST) := type_e;

-- verificam daca exista jucatori inregistrati pentru echipa curenta

-- daca da, afisam toate datele

        if type_e.numar_jucatori is not null then

            DBMS_OUTPUT.put_line('La echipa ' || t(t.LAST).ume_echipa || ' sunt in momentul de fata ' ||
t(t.LAST).numar_jucatori || ' inregistrati, cu varsta medie de ' || t(t.LAST).varsta_medie || ' ani');

-- altfel afisam un mesaj corespunzator

            elsif type_e.varsta_medie is null then

```

```

        DBMS_OUTPUT.put_line('La echipa ' || t(t.LAST).nume echipa || ' nu exista jucatori
inregistrati');

    end if;

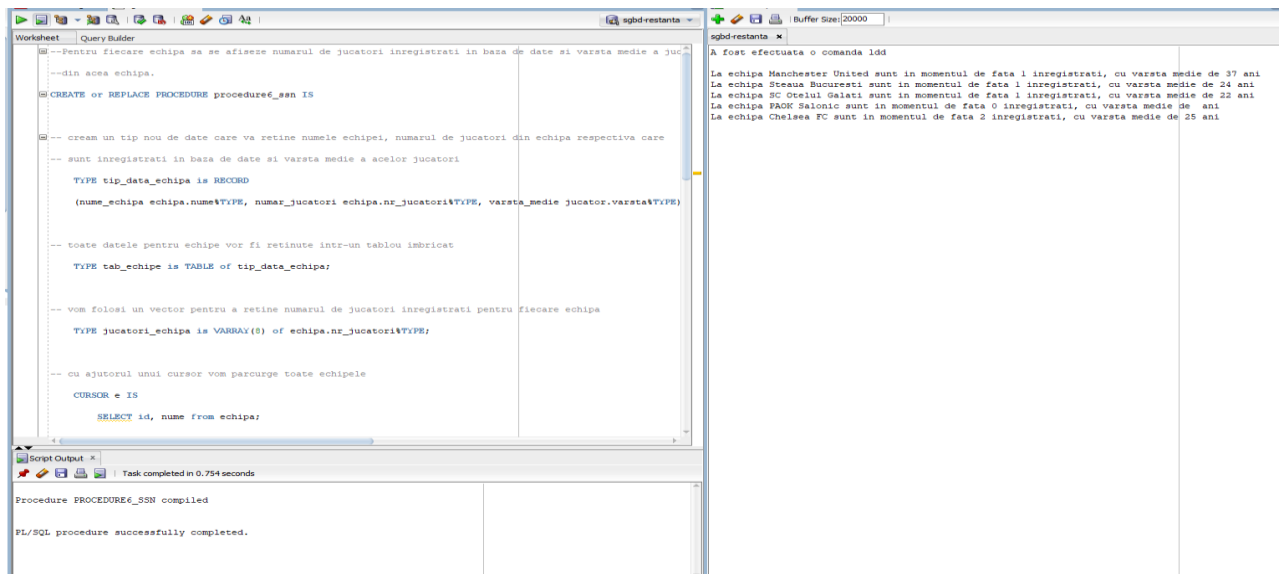
END LOOP;

END procedure6_ssn;

/

EXECUTE procedure6_ssn;

```



7. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent care sa utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind un cursor parametrizat. Apelati subprogramul.

-- cream doua cursoare pentru a verifica data echipele curent sunt eligibile pentru cresterea veniturilor:

-- primul cursor itereaza prin acele echipe ale caror antrenori au o varsta mai mica daca parametrul dat in cursor

-- al doilea cursor itereaza prin acele echipe care au un an de infiintare inainte de anul 1950

```
CREATE OR REPLACE PROCEDURE procedure7_ssn (antrenor_max_age NUMBER) IS
```

-- cream cursorul care va lua toate echipele cu antrenorul mai tanar de 35 de ani

```
CURSOR c(antrenor_max_age NUMBER) IS
```

```
SELECT * FROM echipa e
```

```
LEFT JOIN antrenor a ON e.id = a.id_echipa
```

```
WHERE a.varsta <= antrenor_max_age
```

```
FOR UPDATE of e.venit NOWAIT;
```

```
CURSOR c2 IS
```

```
SELECT * FROM echipa e
```

```
WHERE EXTRACT(YEAR from e.an_infiintare) < 1950
```

```
FOR UPDATE OF e.venit NOWAIT;
```

```
BEGIN
```

```
-- iteram cursorul cu echipe
```

```
FOR i in c(antrenor_max_age) LOOP
```

```
-- in functie de venitul lor actual, actualizam datele
```

```
if i.venit < 10 THEN
```

```
UPDATE ECHIPA
```

```
SET venit = venit * 2
```

```
WHERE CURRENT OF c;
```

```
-- verificam etnia antrenorului
```

```
else
```

```
UPDATE ECHIPA
```

```
SET venit = venit + ((venit * 8)/100)
```

```
WHERE CURRENT of c;
```

```
end if;
```

```
END LOOP;
```

```
FOR j IN c2 LOOP
```

```
UPDATE echipa
```

```
SET venit = venit + ((venit * 10) / 100)
```

```
WHERE CURRENT OF c2;
```

```
END LOOP;
```

```
END procedure7_ssn;
```

```
/
```

```
-- afisam datele inainte si dupa apelarea procedurii
```

```
select * from echipa;
```

```
execute procedure7_ssn(35);
```

```
select * from echipa;
```

```
-- cream doua cursoare pentru a verifica date echipele curent sunt eligibile pentru cresterea veniturilor:
-- primul cursor itereaza prin acele echipe ale caror antrenori au o varsta mai mica daca parametrul dat in cursor
-- al doilea cursor itereaza prin acele echipe care au un an de infiintare inainte de anul 1950

CREATE OR REPLACE PROCEDURE procedure7_ssn (antrenor_max_age NUMBER) IS
-- cream cursorul care va lua toate echipele cu antrenorul mai tanar de 35 de ani
CURSOR c(antrenor_max_age NUMBER) IS
    SELECT * FROM echipa e
    LEFT JOIN antrenor a ON e.id = a.id_echipe
    WHERE a.varsta <= antrenor_max_age
FOR UPDATE OF e.venit NOWAIT;

CURSOR c2 IS
    SELECT * FROM echipa e
    WHERE EXTRACT(YEAR FROM e.an_infiintare) < 1950
FOR UPDATE OF e.venit NOWAIT;

BEGIN
-- iteram cursorul cu echipe
```

A fost efectuată o comandă SQL

Script Output x Query Result x Query Result 1 x

Task completed in 0.946 seconds

Procedure PROCEDURE7_SSN compiled

>>Query Run In:Query Result

PL/SQL procedure successfully completed.

>>Query Run In:Query Result 1

Inainte de update:

ID	NUME	FONDATOR	AN_INFIINTARE	COD	NR_JUCATORI	NR_TOTAL_GOLURI	VENIT
1	Manchester United	Glazer Family	25-MAY-78	UK	26	25768	143
2	Steaua Bucuresti	Valeriu Argaseala	07-JUN-47	RO	31	4680	9
3	SC Otelul Galati	Primaria Galati	10-SEP-64	RO	25	10467	16
4	PAOK Salonic	Ivan Ignatievici Savvidi	03-JAN-26	GK	27	19672	17
5	Chelsea FC	Todd Boehly	10-MAR-05	UK	32	45670	750

Dupa update:

ID	NUME	FONDATOR	AN_INFIINTARE	COD	NR_JUCATORI	NR_TOTAL_GOLURI	VENIT
1	Manchester United	Glazer Family	25-MAY-78	UK	26	25768	157
2	Steaua Bucuresti	Valeriu Argaseala	07-JUN-47	RO	31	4680	20
3	SC Otelul Galati	Primaria Galati	10-SEP-64	RO	25	10467	17
4	PAOK Salonic	Ivan Ignatievici Savvidi	03-JAN-26	GK	27	19672	20
5	Chelsea FC	Todd Boehly	10-MAR-05	UK	32	45670	825

8. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip functie care sa utilizeze intr-o singura comanda SQL 3 dintre tabelele definite. Definiti minim 2 exceptii. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

--Se citeste numele unui jucator de la tastatura. Sa se returneze venitul total al acestuia,


```

--de-a lungul anilor. Daca acesta a fost in trecut si la alte echipe, sa se
--returneze suma tuturor veniturilor sale din prezent si din trecut si sa se afiseze in consola
--venitul facut la fiecare echipa in parte. Se vor arunca erori in
--cazul in care: exista mai multi jucatori cu numele dat, nu exista niciun jucator cu numele
--dat sau nu exista venituri inregistrate pe numele acestuia inca. (ne vom folosi de tabelele
--jucator, echipa si istoric_jucator)

```

```

CREATE or REPLACE FUNCTION func8(v_nume jucator.prenume%TYPE)

```

```

RETURN istoric_jucator.salariu%TYPE IS

```

```

-- crearea exceptiilor ce vor fi aruncate pe parcursul programului in cazul in care apar erori

```

```

    exceptie_0_rows EXCEPTION;

```

```

    exceptie_more_rows EXCEPTION;

```

```

    PRAGMA EXCEPTION_INIT(exceptie_0_rows, -20000);

```

```

    PRAGMA EXCEPTION_INIT(exceptie_more_rows, -20001);

```

```

-- acest tip de data va fi folosit pentru a retine numele echipei si salariul facut de jucator la acea
echipa

```

```

    TYPE t echipa IS RECORD

```

```

        (nume echipa.nume%TYPE, v_total_jucator istoric_jucator.salariu%TYPE);

```

```

-- cream aceasta colectie pentru a retine salariile totale de la toate echipele

```

```

    TYPE jucator_venit IS TABLE OF t echipa INDEX BY PLS_INTEGER;

```

```

-- aceasta variabila va retine venitul total al jucatorului din toti anii

```

```

    venit_total istoric_jucator.salariu%TYPE := 0;

```

```

    counter NUMBER;

```

```

    v_jucator jucator_venit;

```

```

BEGIN

```

```

    dbms_output.put_line('Rezultate pentru jucatorul cu numele: ' || v_nume);

```

```

-- verificam cati jucatori exista cu numele jucatorului venit din parametru

```

```

    SELECT COUNT(*) INTO counter

```

```

    FROM jucator WHERE LOWER(prenume) = LOWER(v_nume);

```

```

-- in functie de rezultat, aruncam exceptii

IF counter = 0 THEN

    RAISE_APPLICATION_ERROR(-20000, 'Nu exista jucator cu acest nume');

ELSIF counter > 1 THEN

    RAISE_APPLICATION_ERROR(-20001, 'Prea multi jucatori cu acest nume');

END IF;


-- verificam daca jucatorul nostru are vreun istoric

SELECT COUNT(*) into counter

FROM jucator j RIGHT OUTER JOIN istoric_jucator ij ON j.id_jucator = ij.id_jucator

WHERE LOWER(j.prenume) = LOWER(v_nume);


-- daca nu, aruncam o exceptie

IF counter = 0 THEN

    RAISE_APPLICATION_ERROR(-20000, 'Nu exista istoric despre acest jucator, ca urmare nu i se
cunosc veniturile');

END IF;


-- preluam atat numele echipelor, cat si salariul total castigat la acestea pentru jucatorul nostru

SELECT e.numa, SUM(ij.salariu) BULK COLLECT INTO v_jucator from istoric_jucator ij

LEFT JOIN jucator j on ij.id_jucator = j.id_jucator

LEFT JOIN echipa e on e.id = ij.id echipa

WHERE LOWER(j.prenume) = LOWER(v_nume)

GROUP BY ij.id echipa, e.numa;


-- afisam datele corespunzatoare pentru fiecare echipa la care jucatorul nostru a avut istoric si
-- adunam salariu respectiv la suma totala

FOR i in 1..v_jucator.COUNT LOOP

    venit_total := venit_total + v_jucator(i).v_total_jucator;

    DBMS_OUTPUT.put_line(v_jucator(i).numa echipa || ' - venit total de-a lungul anilor: ' ||
v_jucator(i).v_total_jucator);

```

```
END LOOP;
```

```
RETURN venit_total;
```

```
EXCEPTION
```

```
WHEN exceptie_0_rows THEN
```

```
    dbms_output.put_line(sqlerrm(-20000));
```

```
    RETURN 0;
```

```
WHEN exceptie_more_rows THEN
```

```
    dbms_output.put_line(sqlerrm(-20001));
```

```
    RETURN 0;
```

```
END func8;
```

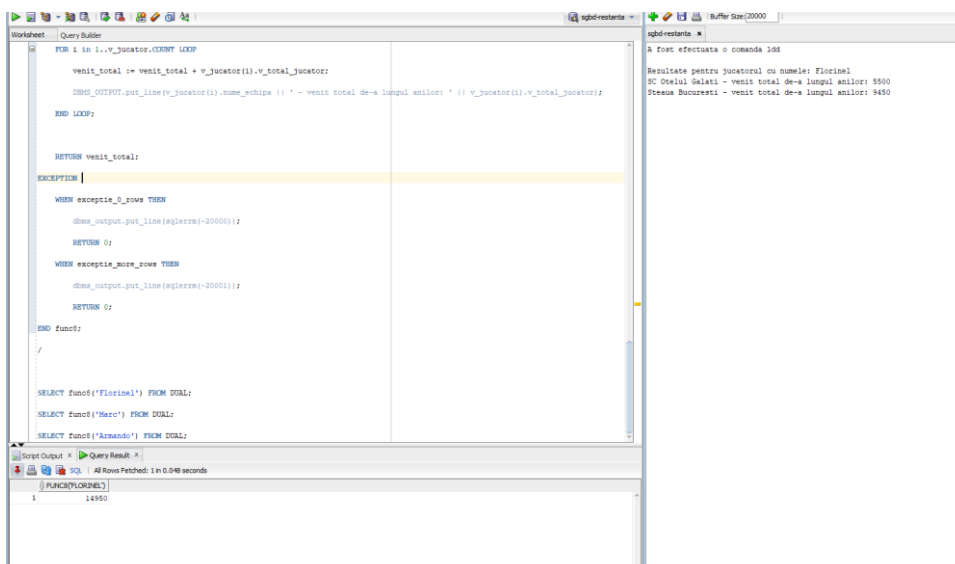
```
/
```

```
SELECT func8('Florinel') FROM DUAL;
```

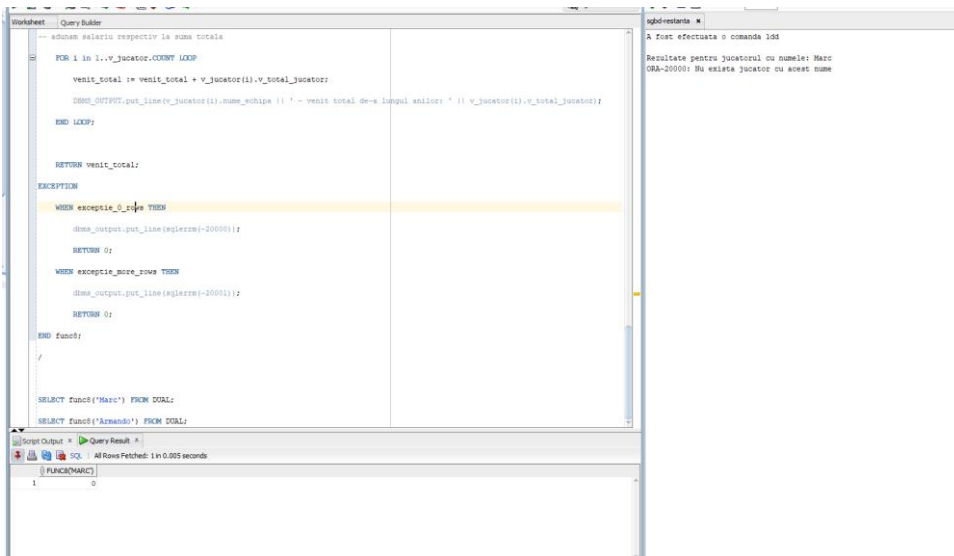
```
SELECT func8('Marc') FROM DUAL;
```

```
SELECT func8('Armando') FROM DUAL;
```

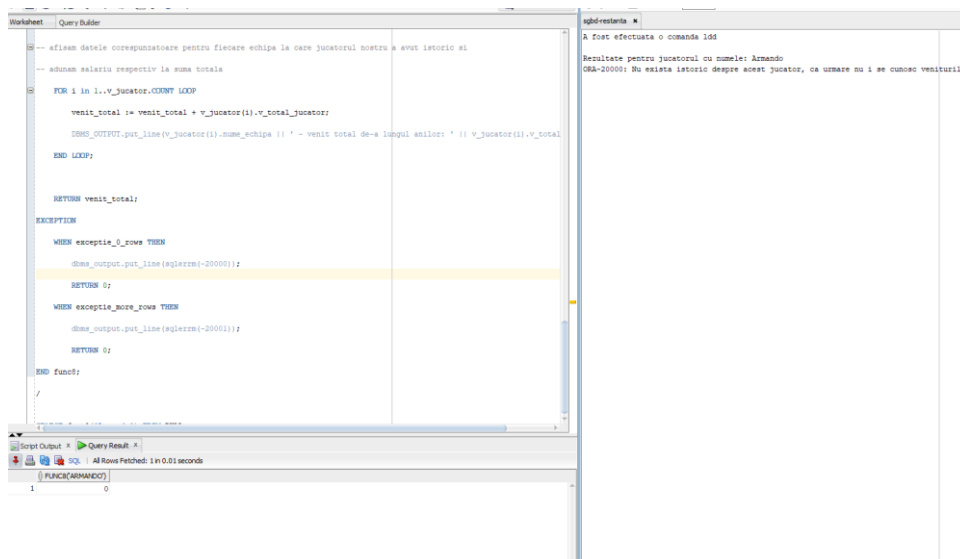
Pentru jucatorul cu prenumele 'Florinel' care exista in baza noastra de date si are si istoric:



Pentru jucatorul cu prenumele 'Marc' care nu exista in baza noastra de date:



Pentru jucatorul cu prenumele 'Armando' care exista in baza noastra de date, dar nu are istoric:



9. Formulati in limbaj natural o problema pe care sa o rezolvati folosind un subprogram stocat independent de tip procedura care sa utilizeze intr-o singura comanda SQL 5 dintre tabelele definite. Tratatati toate exceptiile care pot aparea, incluzand exceptiile NO_DATA_FOUND si TOO_MANY_ROWS. Apelati subprogramul astfel incat sa evidentiati toate cazurile tratate.

--Se da numele unui antrenor ca parametru. Sa se afiseze numele unei echipe pe care a antrenat-o,
 --numele tarii de unde provine, numarul de meciuri pe care le are jucate pana in prezent
 --(istoric_antrenor) si data de inceput si final a unui sezon la care a participat acesta,

--in trecut (istoric_antrenor->sezon)

--(ne vom folosi de tabelele antrenor, echipa,tara, istoric_antrenor, sezon)

CREATE OR REPLACE PROCEDURE proc9_ssn (nume_antrenor antrenor.prenume%TYPE) IS

id_antrenor antrenor.id_antrenor%TYPE;

nume_complet antrenor.prenume%TYPE;

nume echipa echipa.nume%TYPE;

nume_tara tara.nume%TYPE;

interval_sezon VARCHAR2(50);

nr_meciuri istoric_antrenor.nr_meciuri%TYPE;

BEGIN

SELECT ia.id_antrenor, a.nume || ' ' || a.prenume as "Numele Complet", e.nume, t.nume as Tara,
SUM(ia.nr_meciuri),

(select s.data_inceput || ' - ' || s.data_sfarsit from istoric_antrenor ia LEFT JOIN sezon s ON
ia.id_sezon = s.id

LEFT JOIN antrenor a ON a.id_antrenor = ia.id_antrenor WHERE LOWER(nume_antrenor) =
LOWER(a.prenume) and rownum = 1) as sezon

INTO id_antrenor, nume_complet, nume echipa, nume_tara, nr_meciuri, interval_sezon

from istoric_antrenor ia

LEFT JOIN antrenor a ON ia.id_antrenor = a.id_antrenor

LEFT JOIN echipa e ON e.id = ia.id echipa

LEFT JOIN tara t ON a.etnie = t.cod

LEFT OUTER JOIN sezon s ON ia.id_sezon = s.id

WHERE LOWER(nume_antrenor) = LOWER(a.prenume)

group by ia.id_antrenor, a.nume || ' ' || a.prenume, e.nume, t.nume;

-- Afisam toate datele pe care le-am gasit

DBMS_OUTPUT.put_line('Numele complet al antrenorului: ' || nume_complet || ' si are id-ul ' ||
id_antrenor);

DBMS_OUTPUT.put_line('Numele uneia dintre echipe pe care a antrenat-o: ' || nume echipa);

DBMS_OUTPUT.put_line('Numele tarii in care s-a nascut: ' || nume_tara);

```
DBMS_OUTPUT.put_line('Numarul de meciuri la activ: ' || nr_meciuri);
```

```
DBMS_OUTPUT.put_line('Intervalul unui sezon in care antrenorul a fost la o echipa: ' ||  
interval_sezon);
```

EXCEPTION

```
WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista antrenori cu acest  
nume.');
```

```
WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi antrenori  
cu acest nume.');
```

```
END proc9_ssn;
```

```
/
```

```
EXECUTE proc9_ssn('MIREL');
```

The screenshot displays the Oracle SQL Developer environment. The main window is titled 'Query Builder' and contains a PL/SQL procedure named `proc9_ssn`. The procedure's code is as follows:

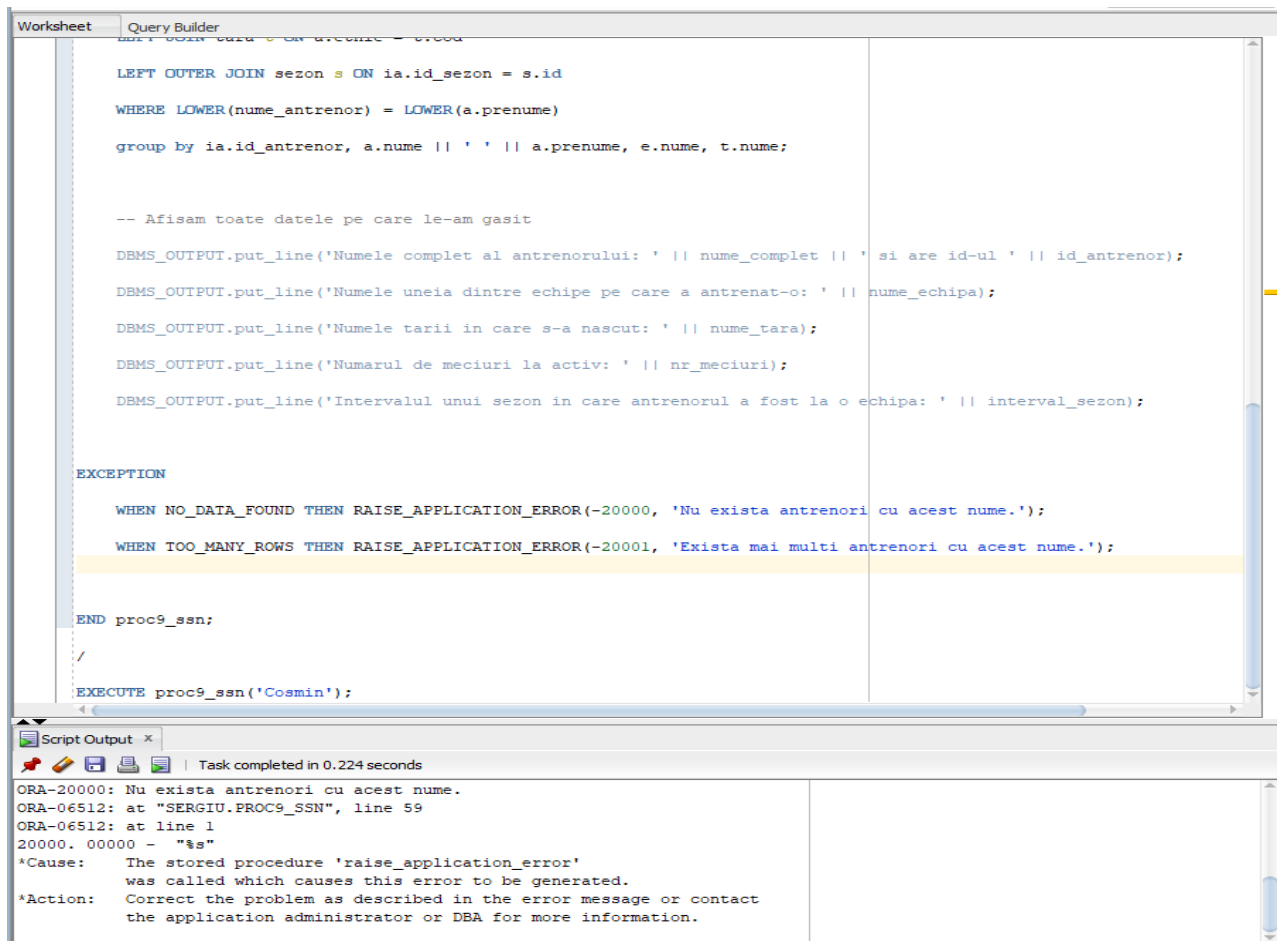
```
LEFT OUTER JOIN sezon s ON ia.id_sezon = s.id  
WHERE LOWER(nume_antrenor) = LOWER(a.prenume)  
group by ia.id_antrenor, a.numa || ' ' || a.prenume, e.numa, t.numa;  
  
-- Afisam toate datele pe care le-am gasit  
DBMS_OUTPUT.put_line('Numele complet al antrenorului: ' || nume_complet || ' si are id-ul ' || id_antrenor);  
DBMS_OUTPUT.put_line('Numele uneia dintre echipe pe care a antrenat-o: ' || nume echipa);  
DBMS_OUTPUT.put_line('Numele tarii in care s-a nascut: ' || nume_tara);  
DBMS_OUTPUT.put_line('Numarul de meciuri la activ: ' || nr_meciuri);  
DBMS_OUTPUT.put_line('Intervalul unui sezon in care antrenorul a fost la o echipa: ' || interval_sezon);  
  
EXCEPTION  
WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista antrenori cu acest nume.');
```

The procedure is executed with the argument 'MIREL'. The 'Script Output' window at the bottom shows the following results:

```
A fost efectuata o comanda ldd  
Numele complet al antrenorului: Radoi Mirel si are id-ul 1  
Numele uneia dintre echipe pe care a antrenat-o: FAGK Salonic  
Numele tarii in care s-a nascut: Romania  
Numarul de meciuri la activ: 78  
Intervalul unui sezon in care antrenorul a fost la o echipa: 07-OCT-22 - 27-MAY-23
```

The 'Script Output' window also indicates that the procedure was compiled successfully and the PL/SQL procedure was successfully completed.

Cand numele nu exista in tabela de antrenori:



The screenshot shows a SQL IDE with a 'Query Builder' tab. The main editor contains a PL/SQL procedure named 'proc9_ssn'. The procedure starts with a 'LEFT OUTER JOIN' between 'sezon' and 'ia' tables, followed by a 'WHERE' clause filtering by 'nume_antrenor'. It then uses 'group by' on 'ia.id_antrenor' and prints several lines of data using 'DBMS_OUTPUT.put_line'. An 'EXCEPTION' block handles 'NO_DATA_FOUND' and 'TOO_MANY_ROWS' errors. The procedure ends with 'END proc9_ssn;' and is executed with 'EXECUTE proc9_ssn('Cosmin');'. Below the editor, the 'Script Output' window shows the execution results, including error messages and a detailed cause and action for the error.

```
LEFT OUTER JOIN sezon s ON ia.id_sezon = s.id
WHERE LOWER(nume_antrenor) = LOWER(a.prenume)
group by ia.id_antrenor, a.nume || ' ' || a.prenume, e.nume, t.nume;

-- Afisam toate datele pe care le-am gasit
DBMS_OUTPUT.put_line('Numele complet al antrenorului: ' || nume_complet || ' si are id-ul ' || id_antrenor);
DBMS_OUTPUT.put_line('Numele uneia dintre echipe pe care a antrenat-o: ' || nume echipa);
DBMS_OUTPUT.put_line('Numele tarii in care s-a nascut: ' || nume_tara);
DBMS_OUTPUT.put_line('Numarul de meciuri la activ: ' || nr_meciuri);
DBMS_OUTPUT.put_line('Intervalul unui sezon in care antrenorul a fost la o echipa: ' || interval_sezon);

EXCEPTION

WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista antrenori cu acest nume.');
```

```
END proc9_ssn;

/

EXECUTE proc9_ssn('Cosmin');
```

Script Output x | Task completed in 0.224 seconds

```
ORA-20000: Nu exista antrenori cu acest nume.
ORA-06512: at "SERGIU.PROC9_SSN", line 59
ORA-06512: at line 1
20000. 00000 - "%s"
*Cause:      The stored procedure 'raise_application_error'
              was called which causes this error to be generated.
*Action:     Correct the problem as described in the error message or contact
              the application administrator or DBA for more information.
```

10. Definiti un trigger de tip LMD la nivel de comanda. Declansati trigger-ul.

-- Declansator care interzice stergerea

-- datelor din tabelul istoric_jucator

CREATE OR REPLACE TRIGGER trigger10_ssn

BEFORE DELETE ON istoric_jucator

BEGIN

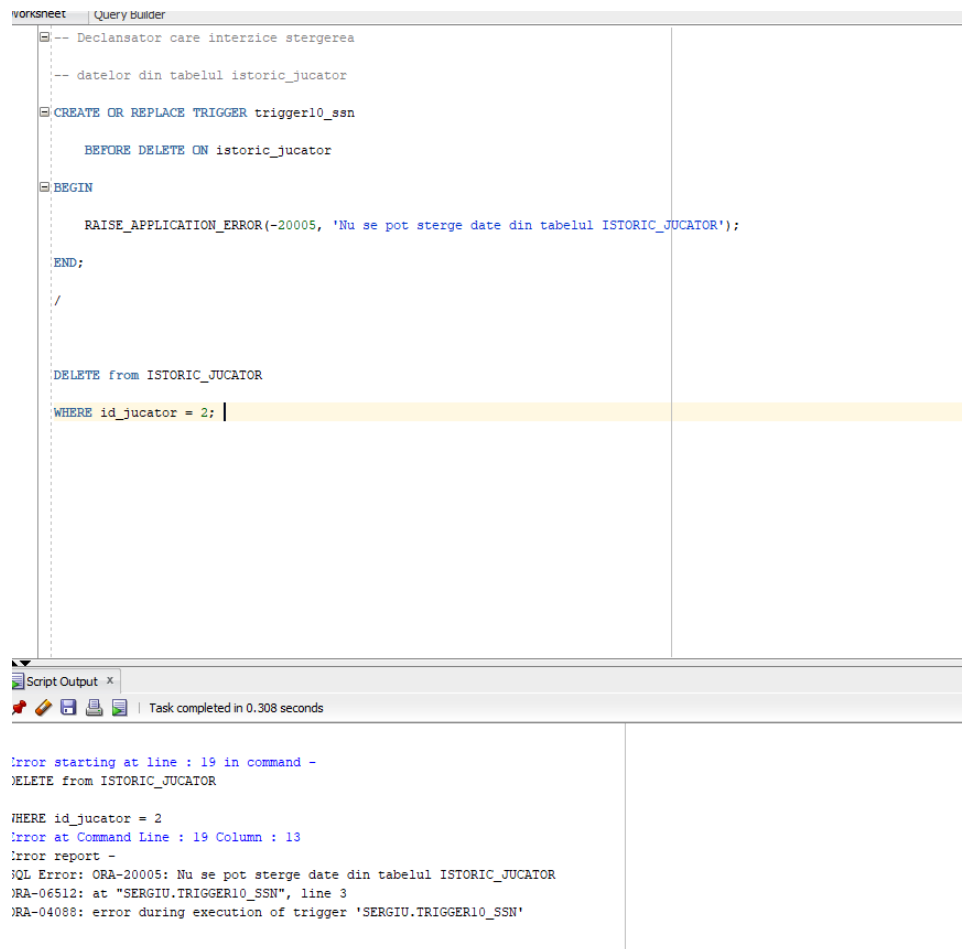
RAISE_APPLICATION_ERROR(-20005, 'Nu se pot sterge date din tabelul ISTORIC_JUCATOR');

END;

/

DELETE from ISTORIC_JUCATOR

WHERE id_jucator = 2;



```
-- Declansator care interzice stergerea
-- datelor din tabelul istoric_jucator

CREATE OR REPLACE TRIGGER trigger10_ssn
    BEFORE DELETE ON istoric_jucator
BEGIN
    RAISE_APPLICATION_ERROR(-20005, 'Nu se pot sterge date din tabelul ISTORIC_JUCATOR');
END;
/

DELETE from ISTORIC_JUCATOR
WHERE id_jucator = 2;
```

Script Output x

Task completed in 0.308 seconds

Error starting at line : 19 in command -
DELETE from ISTORIC_JUCATOR
WHERE id_jucator = 2
Error at Command Line : 19 Column : 13
Error report -
SQL Error: ORA-20005: Nu se pot sterge date din tabelul ISTORIC_JUCATOR
ORA-06512: at "SERGIU.TRIGGER10_SSN", line 3
ORA-04088: error during execution of trigger 'SERGIU.TRIGGER10_SSN'

11. Definiti un trigger de tip LMD la nivel de linie. Declansati trigger-ul.

- Declansator care interzice schimbarea brusca a pozitiei unui jucator in teren
- schimbare brusca: trecerea din fundas in mijlocas sau atacant si vice versa
- schimbare corecta: trecerea din fundas central la fundas lateral
- note: toate pozitiile de fundas au pe a doua pozitie sau a treia litera B
- toate pozitiile de mijlocas au pe a doua pozitie sau a treia litera M
- restul pozitiilor sunt de atacant (ST, CF, LW, RW)

CREATE OR REPLACE TRIGGER trigger11

BEFORE UPDATE on jucator

FOR EACH ROW

BEGIN

IF(((NEW.post like '_B' or :NEW.post like '___B') and (:OLD.post like '_M' or :OLD.post like '___M' or :OLD.post like '___')) or


```
((:NEW.post like '_M' or :NEW.post like '__M') and (:OLD.post like '_B' or :OLD.post like '__B' or :OLD.post like '__')) or
```

```
(:NEW.post like '__' and (:OLD.post like '_M' or :OLD.post like '__M' or :OLD.post like '_B' or :OLD.post like '__B')))) THEN
```

```
RAISE_APPLICATION_ERROR(-20001, 'Trecerea de la vechiul post la noul post e prea brusca');
```

```
END IF;
```

```
END;
```

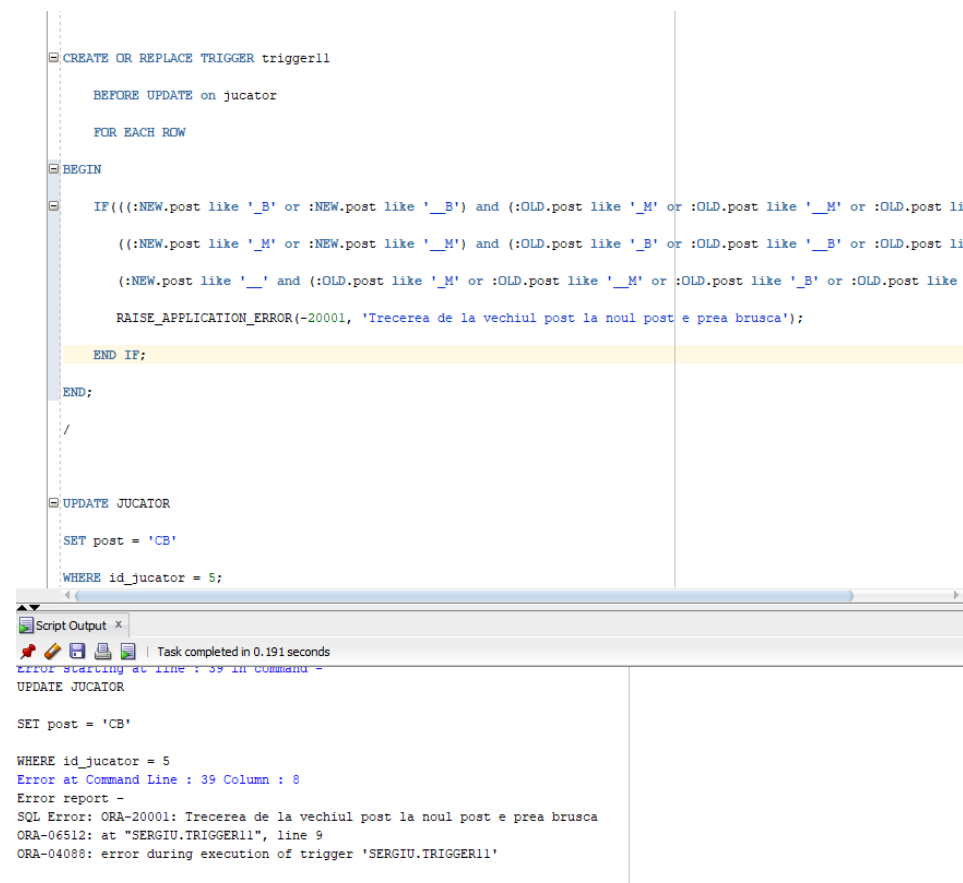
```
/
```

```
UPDATE JUCATOR
```

```
SET post = 'CB'
```

```
WHERE id_jucator = 5;
```

-- Jucatorul cu id-ul 5 are pozitia ST



```
CREATE OR REPLACE TRIGGER trigger11
BEFORE UPDATE on jucator
FOR EACH ROW
BEGIN
IF(((NEW.post like '_B' or NEW.post like '__B') and (:OLD.post like '_M' or :OLD.post like '__M' or :OLD.post li
((NEW.post like '_M' or NEW.post like '__M') and (:OLD.post like '_B' or :OLD.post like '__B' or :OLD.post li
(:NEW.post like '__' and (:OLD.post like '_M' or :OLD.post like '__M' or :OLD.post like '_B' or :OLD.post like
RAISE_APPLICATION_ERROR(-20001, 'Trecerea de la vechiul post la noul post e prea brusca');
END IF;
END;
/

UPDATE JUCATOR
SET post = 'CB'
WHERE id_jucator = 5;
```

Script Output x

Task completed in 0.191 seconds

Error starting at line : 39 in Command -

UPDATE JUCATOR

SET post = 'CB'

WHERE id_jucator = 5

Error at Command Line : 39 Column : 8

Error report -

SQL Error: ORA-20001: Trecerea de la vechiul post la noul post e prea brusca

ORA-06512: at "SERGIU.TRIGGER11", line 9

ORA-04088: error during execution of trigger 'SERGIU.TRIGGER11'

12. Trigger care afiseaza un mesaj de fiecare data cand este rulata o comanda LDD

```
CREATE OR REPLACE TRIGGER triggerex12
```

```
    AFTER CREATE OR ALTER OR DROP ON SCHEMA
```

```
BEGIN
```

```
    DBMS_OUTPUT.PUT_LINE('A fost efectuata o comanda ldd');
```

```
END;
```

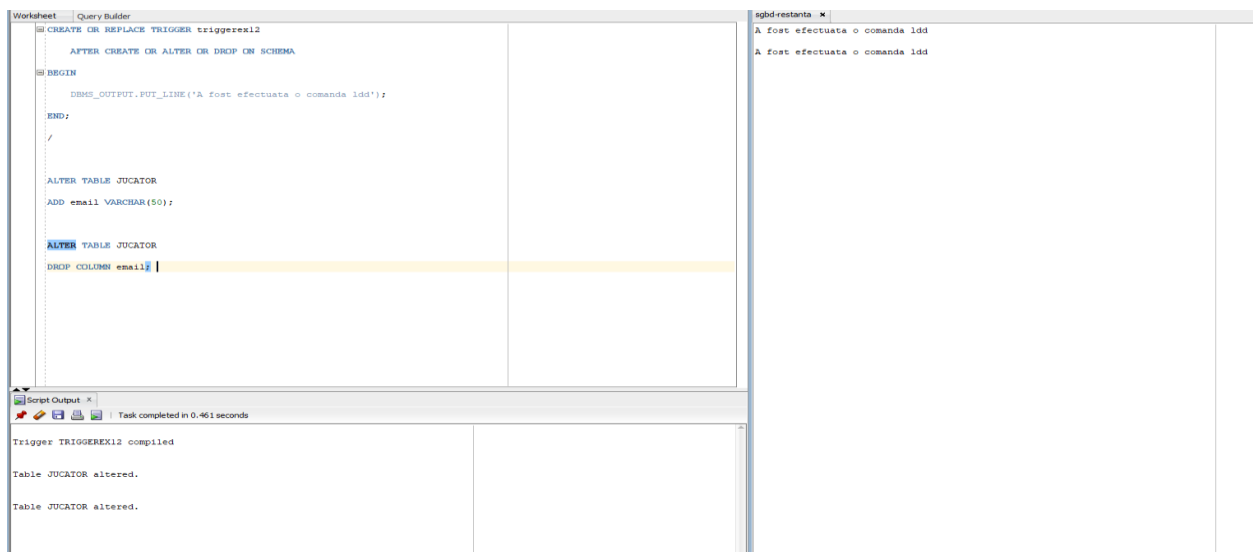
```
/
```

```
ALTER TABLE JUCATOR
```

```
ADD email VARCHAR(50);
```

```
ALTER TABLE JUCATOR
```

```
DROP COLUMN email;
```



13. Definiti un pachet care sa contina toate obiectele definite in cadrul proiectului.

-- 13. Pachet cu toate obiectele din proiect

```
CREATE or REPLACE PACKAGE pachet13_ssn AS
```

```
    PROCEDURE procedure6_ssn;
```

```
    PROCEDURE procedure7_ssn;
```

```
    FUNCTION func8(v_nume jucator.prenume%TYPE) RETURN istoric_jucator.salariu%TYPE;
```

```

PROCEDURE proc9_ssn(ume_antrenor antrenor.prenume%TYPE);
END pachet13_ssn;
/
CREATE or REPLACE PACKAGE BODY pachet13_ssn AS
    -- procedura 6
    PROCEDURE procedure6_ssn IS
        -- cream un tip nou de date care va retine numele echipei, numarul de jucatori din echipa
        respectiva care
        -- sunt inregistrati in baza de date si varsta medie a acelor jucatori
        TYPE tip_data echipa is RECORD
            (ume_echipa echipa.ume%TYPE, numar_jucatori echipa.nr_jucatori%TYPE, varsta_medie
            jucator.varsta%TYPE);
        -- toate datele pentru echipe vor fi retinute intr-un tablou imbricat
        TYPE tab_echipe is TABLE of tip_data echipa;
        -- vom folosi un vector pentru a retine numarul de jucatori inregistrati pentru fiecare echipa
        TYPE jucatori_echipa is VARRAY(8) of echipa.nr_jucatori%TYPE;
        -- cu ajutorul unui cursor vom parcurge toate echipele
        CURSOR e IS
            SELECT id, ume from echipa;

        t tab_echipe := tab_echipe();
        type_e tip_data echipa;
        v jucatori_echipa := jucatori_echipa();

    BEGIN
        FOR i in e LOOP
            -- Vom salva datele pentru numele echipei, numarul de jucatori curenti la echipa si varsta medie a
            acestora
            SELECT AVG(varsta), COUNT(*)
                into type_e.varsta_medie, type_e.numar_jucatori
                from jucator j where j.id_echipa = i.id;

```

```

v.EXTEND;

SELECT COUNT(*) into v(v.last)
from jucator j where j.id echipa = i.id;

type_e.nume echipa := i.nume;

-- adaugam variabila in tabloul imbricat care cere tipul variabilei noastre
t.EXTEND;
t(t.LAST) := type_e;

-- verificam daca exista jucatori inregistrati pentru echipa curenta
-- daca da, afisam toate datele
if type_e.numar_jucatori is not null then
    DBMS_OUTPUT.put_line('La echipa ' || t(t.LAST).nume echipa || ' sunt in momentul de fata '
|| t(t.LAST).numar_jucatori || ' inregistrati, cu varsta medie de ' || t(t.LAST).varsta_medie || ' ani');
-- altfel afisam un mesaj corespunzator
elsif type_e.varsta_medie is null then
    DBMS_OUTPUT.put_line('La echipa ' || t(t.LAST).nume echipa || ' nu exista jucatori
inregistrati');
end if;
END LOOP;
END procedure6_ssn;

-- functia 8
FUNCTION func8(v_nume jucator.prenume%TYPE)
RETURN istoric_jucator.salariu%TYPE IS

TYPE t echipa is RECORD
(nume echipa echipa.nume%TYPE, v_total_jucator istoric_jucator.salariu%TYPE);
TYPE jucator_venit IS TABLE OF t echipa index BY PLS_INTEGER;

```

```

venit_total istoric_jucator.salariu%TYPE := 0;

counter NUMBER;

v_jucator jucator_venit;

BEGIN

SELECT COUNT(*) into counter
FROM jucator WHERE LOWER(prenume) = LOWER(v_nume);

IF counter = 0 THEN
    RAISE_APPLICATION_ERROR(-20000, 'Nu exista nici un jucator cu acest nume');
ELSIF counter > 1 THEN
    RAISE_APPLICATION_ERROR(-20001, 'Sunt mai multi jucatori cu acest nume');
END IF;

SELECT COUNT(*) into counter
FROM jucator j RIGHT OUTER JOIN istoric_jucator ij ON j.id_jucator = ij.id_jucator
WHERE LOWER(j.prenume) = LOWER(v_nume);

IF counter = 0 THEN
    RAISE_APPLICATION_ERROR(-20002, 'Nu exista istoric despre acest jucator, ca urmare nu i se
cunosc veniturile');
END IF;

SELECT e.nume, SUM(ij.salariu) BULK COLLECT INTO v_jucator from istoric_jucator ij
LEFT JOIN jucator j on ij.id_jucator = j.id_jucator
LEFT JOIN echipa e on e.id = ij.id echipa
WHERE LOWER(j.prenume) = LOWER('Florinel')
GROUP BY ij.id echipa, e.nume;

FOR i in 1..v_jucator.COUNT LOOP
    venit_total := venit_total + v_jucator(i).v_total_jucator;

    DBMS_OUTPUT.put_line(v_jucator(i).nume echipa || ' - venit total de-a lungul anilor: ' ||
v_jucator(i).v_total_jucator);

```

```

END LOOP;

RETURN venit_total;
END func8;

-- procedura 7
PROCEDURE procedure7_ssn IS
CURSOR c IS
    SELECT * from echipa e
    LEFT JOIN antrenor a ON e.id = a.id echipa
    WHERE a.varsta <= 35
FOR UPDATE of e.venit NOWAIT;

BEGIN
    FOR i in c LOOP
        if i.venit < 10 THEN
            UPDATE ECHIPA
            SET venit = venit * 2
            WHERE CURRENT OF c;
        else
            if i.etnie like 'RO' THEN
                UPDATE ECHIPA
                SET venit = venit + ((venit * 15)/100)
                WHERE CURRENT OF c;
            else
                UPDATE ECHIPA
                SET venit = venit + ((venit * 8)/100)
                WHERE CURRENT of c;
            end if;
        end if;
    end if;

```

```
END LOOP;

END procedure7_ssn;
```

```
PROCEDURE proc9_ssn (nume_antrenor antrenor.prenume%TYPE) IS
```

```
id_antrenor antrenor.id_antrenor%TYPE;
nume_complet antrenor.prenume%TYPE;
nume echipa echipa.nume%TYPE;
nume_tara tara.nume%TYPE;
interval_sezon VARCHAR2(50);
nr_meciuri istoric_antrenor.nr_meciuri%TYPE;
```

```
BEGIN
```

```
SELECT ia.id_antrenor, a.nume || ' ' || a.prenume as "Numele Complet", e.nume, t.nume as
Tara, SUM(ia.nr_meciuri),
```

```
(select s.data_inceput || ' - ' || s.data_sfarsit from istoric_antrenor ia LEFT JOIN sezon s ON
ia.id_sezon = s.id
```

```
LEFT JOIN antrenor a ON a.id_antrenor = ia.id_antrenor WHERE LOWER(nume_antrenor) =
LOWER(a.prenume) and rownum = 1) as sezon
```

```
INTO id_antrenor, nume_complet, nume echipa, nume_tara, nr_meciuri, interval_sezon
```

```
from istoric_antrenor ia
```

```
LEFT JOIN antrenor a ON ia.id_antrenor = a.id_antrenor
```

```
LEFT JOIN echipa e ON e.id = ia.id echipa
```

```
LEFT JOIN tara t ON a.etnie = t.cod
```

```
LEFT OUTER JOIN sezon s ON ia.id_sezon = s.id
```

```
WHERE LOWER(nume_antrenor) = LOWER(a.prenume)
```

```
group by ia.id_antrenor, a.nume || ' ' || a.prenume, e.nume, t.nume;
```

```
-- Afisam toate datele pe care le-am gasit
```

```
DBMS_OUTPUT.put_line('Numele complet al antrenorului: ' || nume_complet || ' si are id-ul '
|| id_antrenor);
```

```
DBMS_OUTPUT.put_line('Numele uneia dintre echipe pe care a antrenat-o: ' || nume echipa);
```

```
DBMS_OUTPUT.put_line('Numele tarii in care s-a nascut: ' || nume_tara);
```

```

DBMS_OUTPUT.put_line('Numarul de meciuri la activ: ' || nr_meciuri);

DBMS_OUTPUT.put_line('Intervalul unui sezon in care antrenorul a fost la o echipa: ' ||
interval_sezon);

```

EXCEPTION

```

    WHEN NO_DATA_FOUND THEN RAISE_APPLICATION_ERROR(-20000, 'Nu exista antrenori cu
acest nume.');
```

```

    WHEN TOO_MANY_ROWS THEN RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi
antrenori cu acest nume.');
```

```

END proc9_ssn;
```

```

END pachet13_ssn;
```

```

/
```

```

EXECUTE pachet13_ssn.procedure6_ssn;
```

```

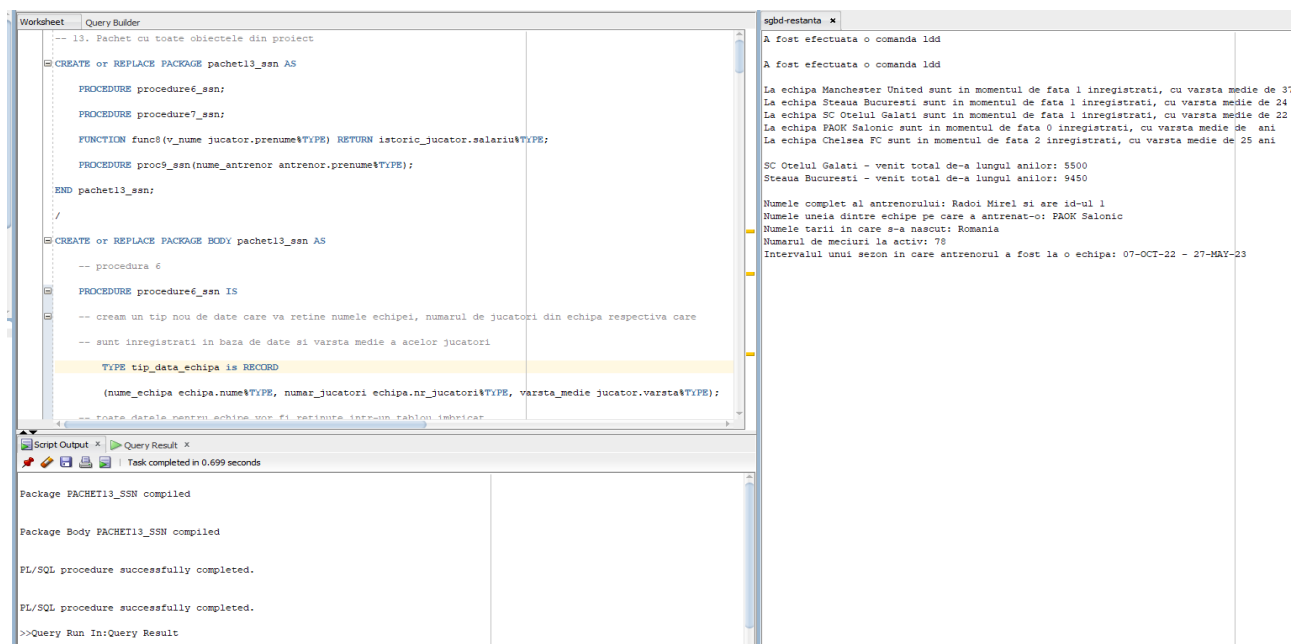
EXECUTE pachet13_ssn.procedure7_ssn;
```

```

SELECT pachet13_ssn.func8('Florinel') from DUAL;
```

```

EXECUTE pachet13_ssn.proc9_ssn('MIREL');
```



The screenshot displays the Oracle SQL Developer environment. The left pane shows the source code of the PL/SQL package 'pachet13_ssn' and its body. The right pane shows the output of the execution, including status messages and query results.

Source Code (Left Pane):

```

-- 13. Pachet cu toate obiectele din proiect
CREATE or REPLACE PACKAGE pachet13_ssn AS
    PROCEDURE procedure6_ssn;
    PROCEDURE procedure7_ssn;
    FUNCTION func8(v_nume_jucator.prenume%TYPE) RETURN istoric_jucator.salariu%TYPE;
    PROCEDURE proc9_ssn(nume_antrenor antrenor.prenume%TYPE);
END pachet13_ssn;

/

CREATE or REPLACE PACKAGE BODY pachet13_ssn AS
    -- procedura 6
    PROCEDURE procedure6_ssn IS
        -- cream un tip nou de date care va retine numele echipei, numarul de jucatori din echipa respectiva care
        -- sunt inregistrati in baza de date si varsta medie a acelor jucatori
        TYPE tip_data echipa IS RECORD
        (
            nume echipa.nume%TYPE, numar_jucatori echipa.nr_jucatori%TYPE, varsta_medie jucator.varsta%TYPE);
        -- toate datele pentru echipa vor fi varianta intr-un tablou ischiar
    END procedure6_ssn;

```

Output (Right Pane):

```

sgbd-restanta x
A fost efectuata o comanda ldd
A fost efectuata o comanda ldd
La echipa Manchester United sunt in momentul de fata 1 inregistrati, cu varsta medie de 31
La echipa Steaua Bucuresti sunt in momentul de fata 1 inregistrati, cu varsta medie de 24
La echipa SC Otelul Galati sunt in momentul de fata 1 inregistrati, cu varsta medie de 22
La echipa PAKK Salonic sunt in momentul de fata 0 inregistrati, cu varsta medie de 25 ani
La echipa Chelsea FC sunt in momentul de fata 2 inregistrati, cu varsta medie de 25 ani
SC Otelul Galati - venit total de-a lungul anilor: 5500
Steaua Bucuresti - venit total de-a lungul anilor: 9450
Numele complet al antrenorului: Radoi Mirel si are id-ul 1
Numele uneia dintre echipe pe care a antrenat-o: PAKK Salonic
Numele tarii in care s-a nascut: Romania
Numarul de meciuri la activ: 70
Intervalul unui sezon in care antrenorul a fost la o echipa: 07-OCT-22 - 27-MAY-23

```

Script Output (Bottom Pane):

```

Package PACHET13_SSN compiled
Package Body PACHET13_SSN compiled
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
>>Query Run In:Query Result

```