Proiectarea unei Baze de Date a

unei competiții amicale europene de fotbal

Realizat de: Iordache Tudor-Dimitrie, grupa 133

**Cuprins**

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

3. Descrierea entităților, incluzând precizarea cheii primare.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3 5.

7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.

8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la

punctul 7.

9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum

5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative;

maxim 30 de înregistrări în fiecare tabel).

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în

ansamblul lor, următoarele elemente:

a) subcereri sincronizate în care intervin cel puțin 3 tabele

b) subcereri nesincronizate în clauza FROM

c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate

(în clauza de HAVING) în care intervin cel puțin 3 tabele (in cadrul aceleiași cereri)

d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)

e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice,

a cel puțin unei expresii CASE

f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Observație: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel

încât cele 5 cereri să le cuprindă pe toate.

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe

vizualizarea respectivă și un exemplu de operație LMD nepermisă.

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care

implementează analiza top-n.

Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.

16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile

operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore

algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.

17. a. Realizarea normalizării BCNF, FN4, FN5.

b. Aplicarea denormalizării, justificând necesitatea acesteia

**1. Descrierea Modelului Real**

În acest proiect, voi prezenta implementarea unei baze de date ce are ca inspirație o competiție amicală de fotbal, la care iau parte cele mai mari echipe de club din fotbalul european. Utilitatea sa constă în ușurința cu care se pot afla informații despre partidele din cadrul competiției, cât și despre componența echipelor participante, din diverse puncte de vedere (la nivel de staff tehnic, administrativ, sau la nivel de jucători) de către pasionați. Se poate spune că reprezintă o fuziune între back-end-urile aplicațiilor Flashscore (<https://www.flashscore.ro>), respectiv Transfermarkt (<https://www.transfermarkt.com>), responsabile pentru furnizarea ultimelor informații din lumea fotbalului, la nivel sportiv și la nivel economic. Partidele introduse in baza de date, ce au ca element central echipele participante (gazde, respectiv oaspeți), prezintă detalii cu privire la scor, brigada de arbitri delegată, numărul de spectatori prezenți. Prin intermediul echipelor, se extrag informații despre lotul de jucători al fiecăreia (cu informații specifice despre aceștia la nivel de carieră), despre campionatul din care fac parte la nivel intern și despre administrarea lor. Ideea provine din pasiunea mea pentru acest sport pe care-l urmăresc în mod regulat și consider că proiectul ”ușurează viața” pasionaților.

**2. Prezentarea Constrângerilor**

- O echipă poate aparține unui singur campionat intern, iar din campionat fac parte mai multe echipe (cel puțin una).

- O echipă poate juca pe un stadion, și stadionul poate aparține unei singure echipe, dar exista și posibilitatea ca o echipă să nu dispună de un stadion

- O echipă poate fi condusă de mai mulți antrenori (așa cum în lumea fotbalului, o echipă are la conducere antrenori principali, respectiv, secundari) – cel puțin unul, iar un antrenor poate conduce o singură echipă.

- O echipă poate avea mai mulți jucători, iar un jucător poate juca la o singura echipă.

- Un jucător poate juca pe mai multe poziții, și pe o poziție pot fi atribuiți mai mulți jucători, dar nu este obligatoriu să existe vreun jucator pe o poziție (așa cum în fotbal sunt unele echipe care nu folosesc anumite poziții, depinzând de formațiile de joc pe care le abordează).

- Un stadion se află într-un oraș, iar într-un oraș se pot găsi mai multe stadioane.

- La o partida pot lua parte exact două echipe, si o echipă poate juca mai multe partide, dar nu este obligatoriu să joace (pot fi unele echipe care la momentul creării bazei de date să nu fi jucat, încă, în cadrul competiției 😊)

- O partidă poate fi arbitrată de mai mulți arbitri (cel puțin unul) – acele brigăzi delegate formate din unul sau mai mulți arbitri în cadrul partidelor – iar un arbitru poate lua parte la mai multe partide, dar nu este obligatoriu să ia parte la una.

**3. Descrierea Entităților**

- Structurile ECHIPE, JUCATORI, POZITII, ANTRENORI, STADIOANE, ORASE, TARI, CAMPIONATE, PARTIDE, ARBITRI

- ECHIPE reprezintă echipele participante la competiție. Fiecare are primary key ”id\_echipa”.

- CAMPIONATE reprezintă campionatele interne din care fac parte, în mod normal, echipele participante la competiție. Fiecare are primary key ”id\_campionat”.

- PARTIDE reprezintă meciurile jucate în cadrul competiției. Fiecare are primary key ”id\_partida”.

- ARBITRI reprezinta arbitrii delegați în cadrul partidelor. Fiecare are primary key ”id\_arbitru”.

- JUCATORI reprezintă jucătorii care fac parte dintr-o echipă. Fiecare are primary key ”id\_jucator”.

- POZITII reprezintă pozițiile pe care joacă jucătorii. Fiecare are primary key ”id\_pozitie”.

- ANTRENORI reprezintă antrenorii echipelor participante. Fiecare are primary key ”id\_antrenor”.

- STADIOANE reprezintă stadioanele pe care echipele obișnuiesc să joace, în campionatul lor intern. Fiecare are primary key ”id\_stadion”.

- ORASE reprezintă orașele în care se află stadioanele. Fiecare are primary key ”id\_oras”.

- TARI reprezintă țările în care se află orașele. Fiecare are primary key ”id\_tara”.

**4. Descrierea Relațiilor**

- ECHIPE(LE) aparțin unor CAMPIONATE (în ce campionat joacă fiecare echipă - relație cu cardinalitate minimă 1:1 si cardinalitate maximă 1:n)

- ECHIPE(LE) joacă pe STADIOANE (pe ce stadion joacă fiecare echipă - cardinalitate 1:1)

- JUCATORI(I) joacă la ECHIPE (la ce echipă joacă fiecare jucator – cardinalitate n:1)

- JUCATORI(I) joacă pe POZITII (pe ce poziție joacă fiecare jucător – relație de tip many to many cu cardinalitate minimă 0:1 si cardinalitate maximă m:n)

- ECHIPE(LE) sunt conduse de ANTRENORI (ce echipă antrenează fiecare antrenor – relație cu cardinalitate minimă 1:1 și cardinalitate maximă 1:m)

- ECHIPE(LE) joacă PARTIDE (ce echipă(gazdă) și echipă(oaspete) joacă partida – cardinalitate minimă 1:0 și cardinalitate maximă 1:n)

- PARTIDE(LE) au ARBITRI (ce arbitru arbitrează fiecare partidă – relație de tip Many to Many cu cardinalitate minimă 0:1 și cardinalitate maximă m:n)

- CAMPIONATE(LE) se află în TARI (cărei îi aparține fiecare campionat – cardinalitate 1:1)

- TARI(LE) au ORASE (în ce țară se află fiecare oraș – relație cu cardinalite minimă 1:1 și cardinalitate maximă m:1)

- STADIOANE(LE) se află în ORASE (în ce oraș se află fiecare stadion – relație cu cardinalitate minimă 0:1 și cardinalitate maximă m:1)

**5. Descrierea Atributelor**

- Tabela ECHIPE are ca atribute:

id\_echipa = primary key, de tip întreg.

id\_campionat = foreign key, cu trimitere la id\_campionat din tabela CAMPIONATE, reprezintă campionatul intern în care joacă fiecare echipă.

id\_stadion = foreign key, cu trimitere la id\_stadion din tabela STADIOANE, reprezintă stadionul pe care joacă, în mod normal, fiecare echipă.

nume\_echipa = variabilă de tip varchar, cu lungime maximă 30 care, în mod evident, reprezintă numele echipei.

- Tabela JUCATORI are ca atribute:

id\_jucator = primary key, de tip întreg.

id\_echipa = foreign key, cu trimitere la id\_echipa din tabela ECHIPE, reprezintă echipa la care este legitimat fiecare jucător.

nume\_de\_familie = variabilă de tip varchar, cu lungime maximă 30 – numele de familie al fiecărui jucător.

prenume = variabilă de tip varchar, cu lungime maximă 30 – prenumele fiecărui jucător.

salariu\_saptamanal = variabilă de tip întreg.

apariții = variabilă de tip întreg, reprezintă numărul de meciuri jucate în cariera fiecărui jucător.

goluri = variabilă de tip întreg, reprezintă numărul de goluri marcate în cariera fiecărui jucător.

assisturi = variabilă de tip întreg, reprezintă numărul de pase de gol date de fiecare jucător în cariera lui.

cota\_de\_piață = variabilă de tip întreg, reprezintă valoarea/prețul de vânzare al fiecărui jucător.

data\_semnare\_contract = variabilă de tip date.

- Tabela POZITII are ca atribute:

id\_pozitie = primary key, de tip întreg.

pozitie = variabilă de tip varchar, cu lungime maximă 30 – numele poziției.

- Tabela ANTRENORI are ca atribute:

id\_antrenor = primary key, de tip întreg.

id\_echipa = foreign key, de tip întreg, cu trimitere la id\_echipa din tabela ECHIPE, reprezintă echipa pe care o antrenează fiecare antrenor.

nume\_de\_familie = variabilă de tip varchar, cu lungime maximă 30.

prenume = variabilă de tip varchar, cu lungime maximă 30.

salariu\_saptamanal = variabilă de tip întreg.

data\_semnare\_contract = variabilă de tip date.

- Tabela STADIOANE are ca atribute:

id\_stadion = primary key, de tip întreg.

id\_oras = foreign key, de tip întreg, cu trimitere la id\_oras din tabela ORASE, reprezintă orașul în care se află fiecare stadion.

nume\_stadion = variabilă de tip varchar, cu lungime maximă 30.

capacitate = variabilă de tip întreg.

- Tabela ORASE are ca atribute:

id\_oras = primary key, de tip întreg.

id\_tara = foreign key, de tip întreg, cu trimitere la id\_tara din tabela TARI, reprezintă țara în care se află fiecare oraș.

nume = variabilă de tip varchar, cu lungime maximă 30.

- Tabela TARI are ca atribute:

id\_tara = primary key, de tip întreg.

nume = variabilă de tip varchar, cu lungime maximă 30.

- Tabela CAMPIONATE are ca atribute:

id\_campionat = primary key, de tip întreg.

id\_tara = foreign key, de tip întreg, cu trimitere la id\_tara din tabela TARI, reprezintă id-ul țării din care provine fiecare campionat.

nume = variabilă de tip varchar, cu lungime maximă 30.

- Tabela PARTIDE are ca atribute:

id\_partida = primary key, de tip întreg.

id\_gazde = foreign key, de tip întreg, cu trimitere la id\_echipa din tabela ECHIPE, reprezintă echipa care găzduiește fiecare partidă.

id\_oaspeti = foreign key, de tip întreg, cu trimitere, de asemenea, la id\_echipa din tabela ECHIPE, reprezintă echipa oaspete din cadrul fiecărei partide.

numar\_de\_spectatori = variabilă de tip întreg, reprezintă spectatorii prezenți la fiecare partidă.

goluri\_gazde = variabilă de tip întreg. (1)

goluri\_oaspeți = variabilă de tip întreg. (2)

În funcție de (1) și (2), se decide câștigătoarea fiecărei partide. În cazul în care cele două variabile coincid, meciul se termină, deci, cu un rezultat de egalitate.

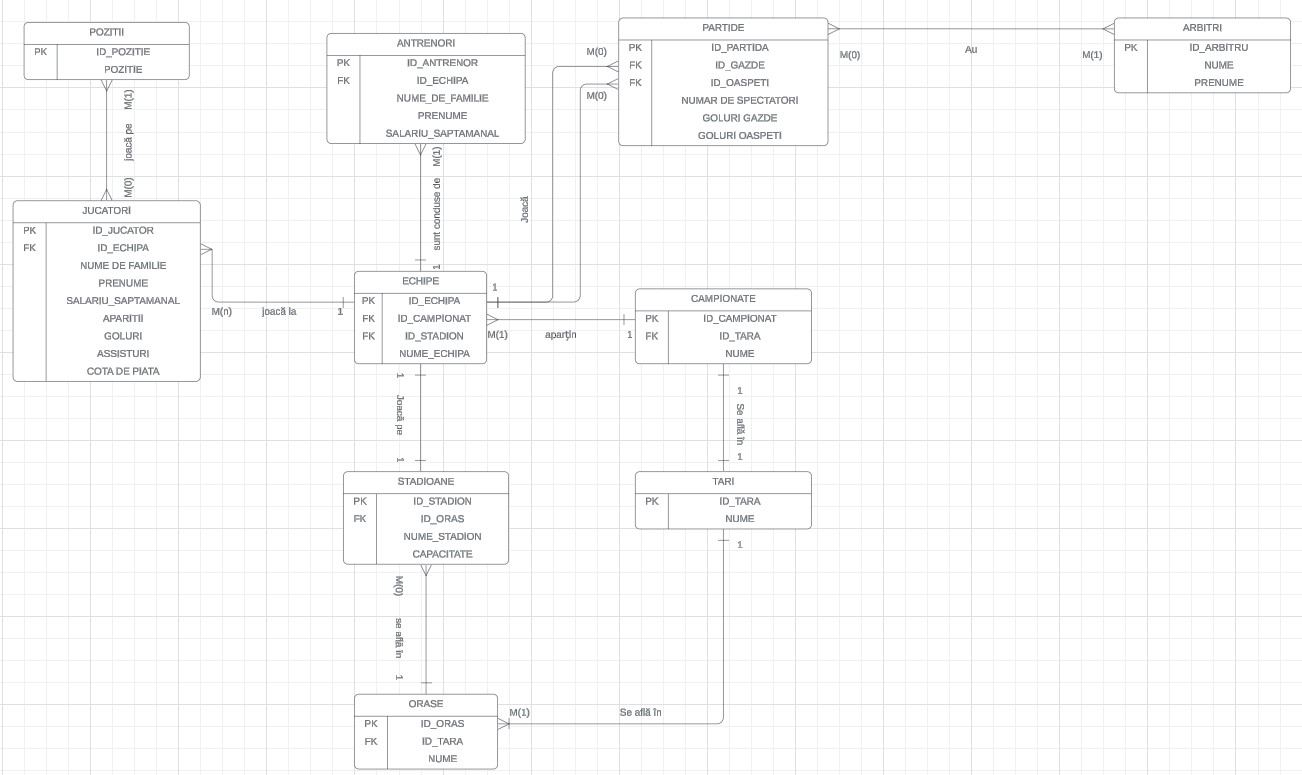
- Tabela ARBITRI are ca atribute:

id\_arbitru = primary key, de tip întreg.

nume = variabilă de tip varchar, cu lungime maximă 30.

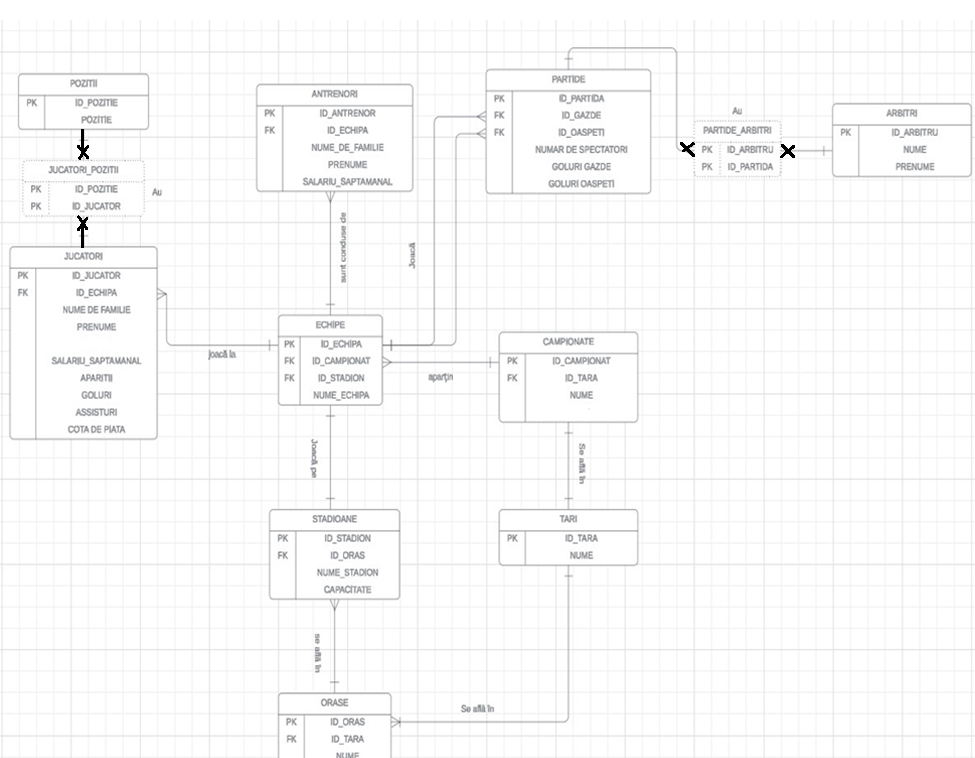
prenume = variabilă de tip varchar, cu lungime maximă 30.

**6. Diagrama Entitate – Relație**



- Notă: în tabelele Jucatori și Antrenori lipsește atributa ”DATA\_SEMNARE\_CONTRACT”, de tip date, prezentă in crearea tabelelor in SQL.

**7. Diagrama Conceptuală**

****

- Notă: în tabelele Jucatori și Antrenori lipsește atributa ”DATA\_SEMNARE\_CONTRACT”, de tip date, prezentă in crearea tabelelor in SQL.

**8. Enumerarea Schemelor Relaționale**

Echipe (id\_echipa pk, id\_campionat fk, id\_stadion fk, nume\_echipa)

Stadioane (id\_stadion pk, id\_oras fk, nume\_stadion, capacitate)

Orase (id\_oras pk, id\_tara fk, nume)

Tari (id\_tara pk, nume)

Campionate (id\_campionat pk, id\_tara fk)

Antrenori (id\_antrenor pk, id\_echipa fk, nume\_de\_familie, prenume, salariu\_saptamanal, data\_semnare\_contract)

Jucatori (id\_jucator pk, id\_echipa fk, nume\_de\_familie, prenume, salariu\_saptamanal, aparitii, goluri, assisturi, cota de piata, data\_semnare\_contract) \*

Pozitii (id\_pozitie fk, pozitie) \*

Jucatori\_Pozitii (id\_pozitie pk, id\_jucator pk) \*

Partide (id\_partida pk, id\_gazde fk, id\_oaspeti fk, numar\_de\_spectatori, goluri\_gazde, goluri\_oaspeti) \*\*

Arbitri (id\_arbitru pk, nume, prenume) \*\*

Partide\_Arbitri (id\_arbitru pk, id\_partida pk) \*\*

**9. Realizarea Normalizării până la forma normală 3 (FN1-FN3)**

„Aflarea in FN1” este garantată de existența cheilor primare în fiecare tabelă a bazei de date. Un exemplu care ar fi contrazis FN1 ar fi fost existența ipotetică a doua partide disputate între aceleași două echipe încheiate la același scor (ex: Tur - Echipa1 0-0 Echipa2, Retur – Echipa1 0-0 Echipa2), neputând fi diferențiate fără id-ul (diferit) al fiecărei partide.

„FN2” constă în faptul că toate elementele tabelelor sunt dependente funcțional de cheile primare. De asemenea, tabelele asociative sunt alcătuite doar dintr-o cheie primară compusă. Un exemplu care ar fi contrazis forma normală 2 este o greșeală pe care am facut-o chiar eu pe parcursul creării tabelelor, și anume introducerea atributei varchar ”poziție” în tabela Jucatori, care depinde de id\_pozitie (din tabela Pozitie), nu si de id\_jucator.

Normalizarea până la FN3 are loc deoarece toate atributele non-cheie depind doar de cheile primare. Ar fi fost greșit dacă, spre exemplu, în tabela Jucători, pe lângă atributa de tip întreg „Goluri”, as fi avut o atribută numită „Goluri cu capul” ( <= Goluri), care ar fi fost extrasă, ca valoare, din numărul total de goluri (atributa principală), deci ar fi depins de ea.

**10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele**create sequence seq

start with 1

increment by 1

minvalue 1

maxvalue 6;

**11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.**

- Observatie: pe parcursul creării tabelelor, din pricina neatenției, unele tabele au fost alterate.

--exercitiul 11

--Tabela Tari

Create table Tari

(

id\_tara int NOT NULL,

nume varchar2(30) NOT NULL UNIQUE,

primary key(id\_tara)

);

Describe Tari;

--Inserarea datelor in tabela Tari

insert into Tari(id\_tara, nume) values(seq.nextval, 'Anglia');

insert into Tari(id\_tara, nume) values(seq.nextval, 'Italia');

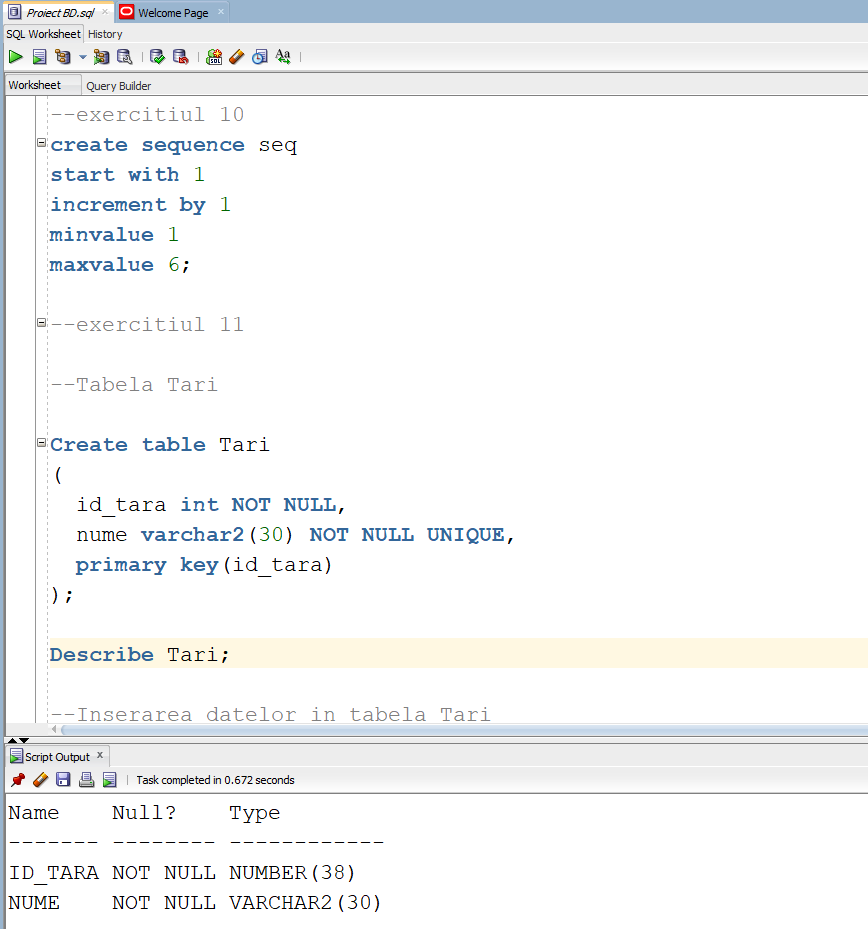
insert into Tari(id\_tara, nume) values(seq.nextval, 'Germania');

insert into Tari(id\_tara, nume) values(seq.nextval, 'Spania');

insert into Tari(id\_tara, nume) values(seq.nextval, 'Franta');

insert into Tari(id\_tara, nume) values(seq.nextval, 'Romania');

select \* from Tari;





--Tabela campionate

Create table Campionate

(

id\_campionat int NOT NULL,

id\_tara int NOT NULL,

nume varchar2(30),

tara varchar2(30),

primary key(id\_campionat),

foreign key(id\_tara) references Tari(id\_tara)

);

Describe Campionate;

select \* from Campionate;

--Inserarea datelor in tabela campionate

insert into Campionate values(10, 1, 'Premier League', 'Anglia');

insert into Campionate values(20, 2, 'Serie A', 'Italia');

insert into Campionate values(30, 3, 'Bundesliga', 'Germania');

insert into Campionate values(40, 4, 'La Liga', 'Spania');

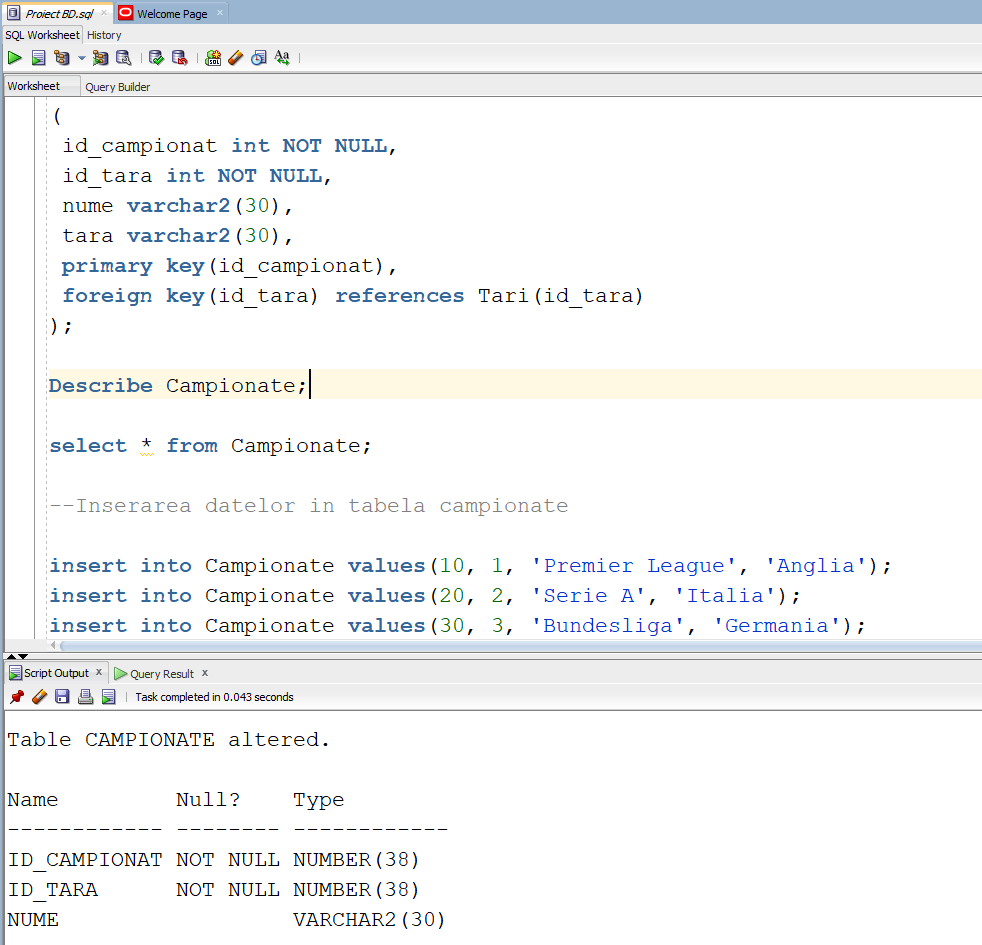
insert into Campionate values(50, 5, 'Ligue 1', 'Franta');

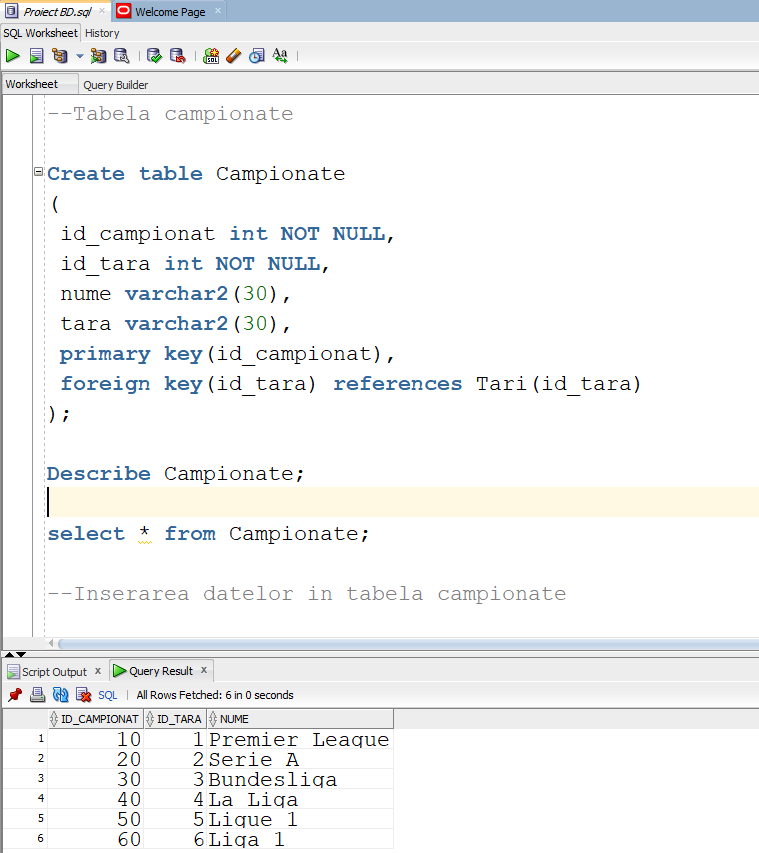
insert into Campionate values(60, 6, 'Liga 1', 'Romania');

alter table Campionate

drop column Tara; - aici am realizat ca variabila tara ar fi incalcat FN2

savepoint taricamp;



****

--Tabela orase

Create table Orase

(

id\_oras int NOT NULL,

id\_tara int NOT NULL,

nume varchar2(30) NOT NULL UNIQUE,

primary key(id\_oras),

foreign key(id\_tara) references Tari(id\_tara)

);

alter table Orase

add tara varchar2(30);

alter table Orase

drop column Tara;

--Inserarea valorilor in tabela Orase

INSERT INTO Orase VALUES (11, 1, 'Londra', 'Anglia');

INSERT INTO Orase VALUES (12, 1, 'Manchester', 'Anglia');

INSERT INTO Orase VALUES (13, 1, 'Birmingham', 'Anglia');

INSERT INTO Orase VALUES (14, 1, 'Liverpool', 'Anglia');

INSERT INTO Orase VALUES (15, 1, 'Leeds', 'Anglia');

INSERT INTO Orase VALUES (21, 2, 'Roma', 'Italia');

INSERT INTO Orase VALUES (22, 2, 'Milano', 'Italia');

INSERT INTO Orase VALUES (23, 2, 'Torino', 'Italia');

INSERT INTO Orase VALUES (24, 2, 'Florenta', 'Italia');

INSERT INTO Orase VALUES (25, 2, 'Napoli', 'Italia');

INSERT INTO Orase VALUES (31, 3, 'Berlin', 'Germania');

INSERT INTO Orase VALUES (32, 3, 'Munchen', 'Germania');

INSERT INTO Orase VALUES (33, 3, 'Hamburg', 'Germania');

INSERT INTO Orase VALUES (34, 3, 'Frankfurt', 'Germania');

INSERT INTO Orase VALUES (35, 3, 'Dortmund', 'Germania');

INSERT INTO Orase VALUES (41, 4, 'Madrid', 'Spania');

INSERT INTO Orase VALUES (42, 4, 'Barcelona', 'Spania');

INSERT INTO Orase VALUES (43, 4, 'Valencia', 'Spania');

INSERT INTO Orase VALUES (44, 4, 'Sevilia', 'Spania');

INSERT INTO Orase VALUES (45, 4, 'Bilbao', 'Spania');

INSERT INTO Orase VALUES (51, 5, 'Paris', 'Franta');

INSERT INTO Orase VALUES (52, 5, 'Marsilia', 'Franta');

INSERT INTO Orase VALUES (53, 5, 'Lyon', 'Franta');

INSERT INTO Orase VALUES (54, 5, 'Bordeaux', 'Franta');

INSERT INTO Orase VALUES (55, 5, 'Nisa', 'Franta');

INSERT INTO Orase VALUES (61, 6, 'Bucuresti', 'Romania');

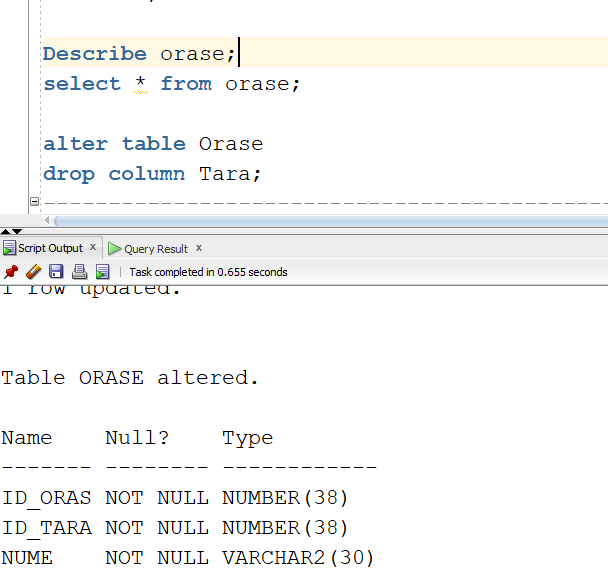
INSERT INTO Orase VALUES (62, 6, 'Cluj', 'Romania');

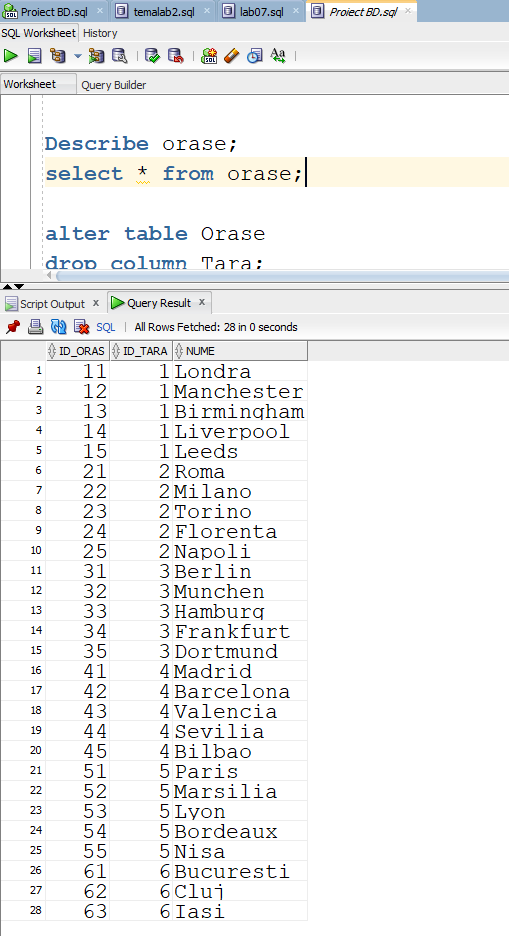
INSERT INTO Orase VALUES (63, 6, 'Iasi', 'Romania');

rollback;

Describe orase;

select \* from orase;





--Tabela stadioane

Create table Stadioane

(

id\_stadion int NOT NULL,

id\_oras int NOT NULL,

nume\_stadion varchar(30),

capacitate int NOT NULL,

primary key(id\_stadion),

foreign key(id\_oras) references Orase(id\_oras)

);

Describe Stadioane;

commit;

--Inserarea datelor in tabela Stadioane

INSERT INTO Stadioane VALUES (220, 22, 'San Siro', 80000);

INSERT INTO Stadioane VALUES (230, 23, 'Juventus Stadium', 41507);

INSERT INTO Stadioane VALUES (610, 61, 'Stefan cel Mare', 15532);

INSERT INTO Stadioane VALUES (510, 51, 'Parc des Princes', 48712);

INSERT INTO Stadioane VALUES (410, 41, 'Camp Nou', 99354);

INSERT INTO Stadioane VALUES (420, 42, 'Santiago Bernabéu', 81044);

INSERT INTO Stadioane VALUES (320, 32, 'Allianz Arena', 75000);

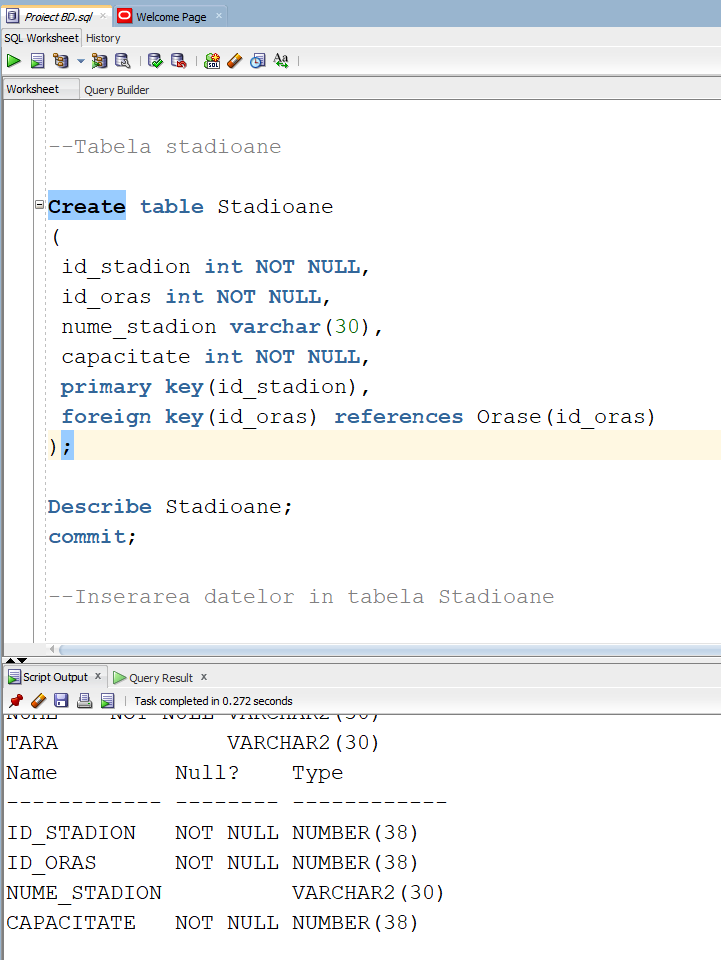
INSERT INTO Stadioane VALUES (350, 35, 'Signal Iduna Park', 81365);

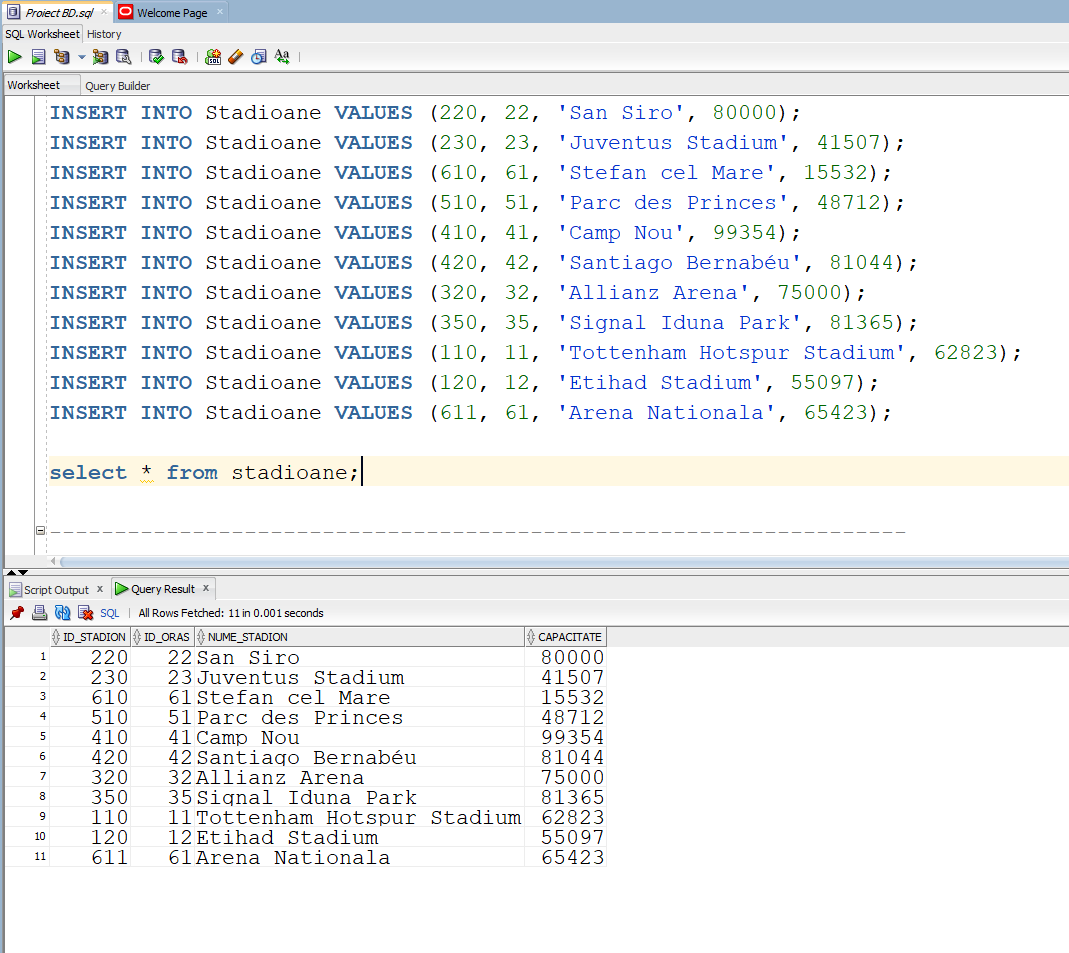
INSERT INTO Stadioane VALUES (110, 11, 'Tottenham Hotspur Stadium', 62823);

INSERT INTO Stadioane VALUES (120, 12, 'Etihad Stadium', 55097);

INSERT INTO Stadioane VALUES (611, 61, 'Arena Nationala', 65423);

select \* from stadioane;





--tabela Echipe

create Table Echipe

(

id\_echipa int NOT NULL,

id\_campionat int NOT NULL,

id\_stadion int NOT NULL,

nume\_echipa varchar(30) NOT NULL UNIQUE,

primary key(id\_echipa),

foreign key(id\_campionat) references Campionate(id\_campionat),

foreign key(id\_stadion) references Stadioane(id\_stadion)

);

Describe Echipe;

select \* from Echipe;

ALTER TABLE ECHIPE

MODIFY ID\_STADION INT NULL;

ALTER TABLE Echipe

ADD campionat VARCHAR2(30);

ALTER TABLE Echipe

ADD tara VARCHAR2(30);

ALTER TABLE Echipe

ADD stadion VARCHAR2(30);

Alter table Echipe

drop column campionat;

Alter table Echipe

drop column Tara;

Alter table Echipe

drop column Stadion;

-- Adaugare echipele în tabelul Echipe, împreun? cu informa?iile suplimentare

INSERT INTO Echipe Values (562, 20, 220, 'Inter Milano', 'Serie A', 'Italia', 'San Siro');

INSERT INTO Echipe Values (134, 20, 230, 'Juventus Torino', 'Serie A', 'Italia', 'Juventus Stadium');

INSERT INTO Echipe Values (789, 60, 610, 'Dinamo Bucuresti', 'Liga 1', 'Romania', 'Stadionul Dinamo');

INSERT INTO Echipe Values (245, 50, 510, 'Paris Saint Germain', 'Ligue 1', 'Franta', 'Parc des Princes');

INSERT INTO Echipe Values (878, 40, 410, 'FC Barcelona', 'La Liga', 'Spania', 'Camp Nou');

INSERT INTO Echipe Values (416, 40, 420, 'Real Madrid', 'La Liga', 'Spania', 'Santiago Bernabéu');

INSERT INTO Echipe Values (721, 30, 320, 'Bayern Munchen', 'Bundesliga', 'Germania', 'Allianz Arena');

INSERT INTO Echipe Values (333, 30, 350, 'Borussia Dortmund', 'Bundesliga', 'Germania', 'Signal Iduna Park');

INSERT INTO Echipe Values (599, 10, 110, 'Tottenham Hotspur', 'Premier League', 'Anglia', 'Tottenham Hotspur Stadium');

INSERT INTO Echipe Values (902, 10, 120, 'Manchester City', 'Premier League', 'Anglia', 'Etihad Stadium');

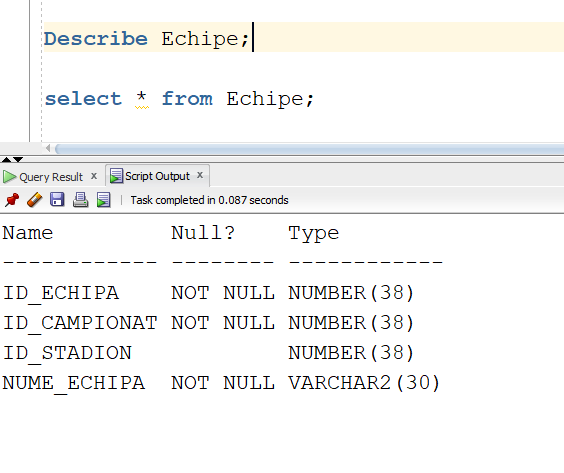
INSERT INTO Echipe Values (734, 60, NULL, 'FCSB', 'Liga 1', 'ROMANIA', '');

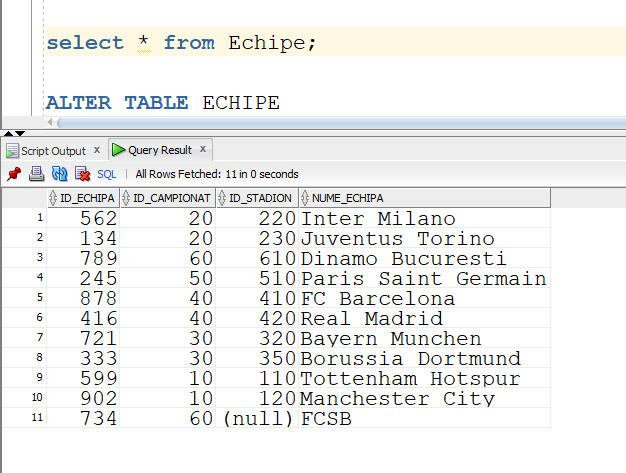
UPDATE Echipe

SET tara = 'Romania'

WHERE tara = 'ROMANIA';

--mic typo





--Crearea tabelei Antrenori

CREATE TABLE Antrenori

(

id\_antrenor int NOT NULL,

id\_echipa int NULL,

nume\_de\_familie varchar2(30),

prenume varchar2(30),

salariu\_saptamanal int NULL,

data\_semnare\_contract DATE NULL

);

ALTER TABLE Antrenori

ADD CONSTRAINT pk\_antrenor\_id PRIMARY KEY (id\_antrenor);

ALTER TABLE Antrenori

ADD CONSTRAINT fk\_echipa\_id FOREIGN KEY (id\_echipa) REFERENCES Echipe(id\_echipa);

Describe Antrenori;

select \* from Antrenori;

--Inserarea valorilor in tabela Antrenori

INSERT INTO Antrenori VALUES (1, 562, 'Inzaghi', 'Simone', 10000, TO\_DATE('2023-06-03', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (2, 134, 'Allegri', 'Massimiliano', 12000, TO\_DATE('2021-05-28', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (3, 789, 'Kopic', 'Zeljko', 8000, TO\_DATE('2023-12-01', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (4, 245, 'Enrique', 'Luis', 11000, TO\_DATE('2023-07-05', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (5, 878, 'Hernandez', 'Xavi', 9000, TO\_DATE('2021-11-06', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (6, 416, 'Ancelotti', 'Carlo', 13000, TO\_DATE('2021-07-01', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (7, 721, 'Tuchel', 'Thomas', 10000, TO\_DATE('2023-03-24', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (8, 333, 'Terzic', 'Edin', 9500, TO\_DATE('2022-07-01', 'YYYY-MM-DD'));

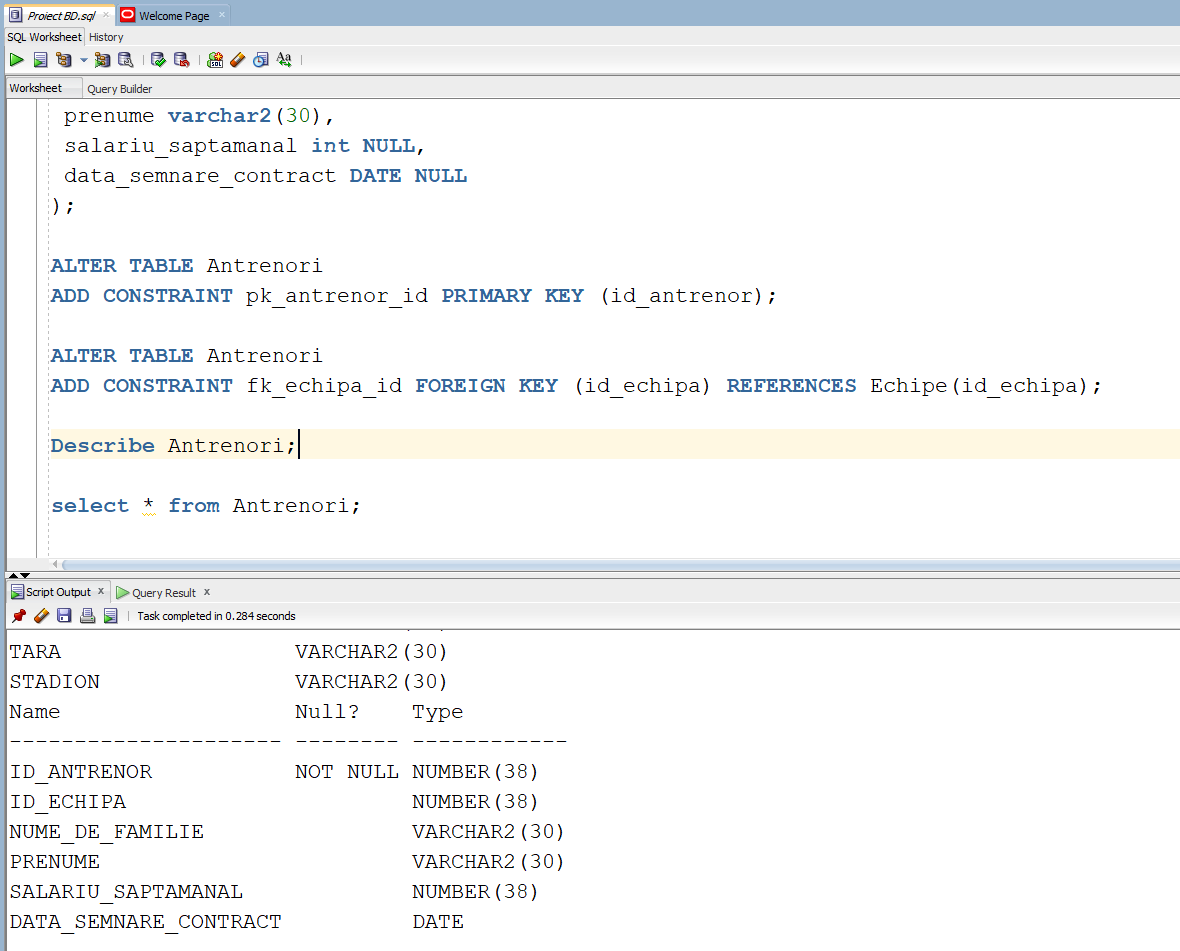
INSERT INTO Antrenori VALUES (9, 599, 'Postecoglou', 'Ange', 10500, TO\_DATE('2023-07-01', 'YYYY-MM-DD'));

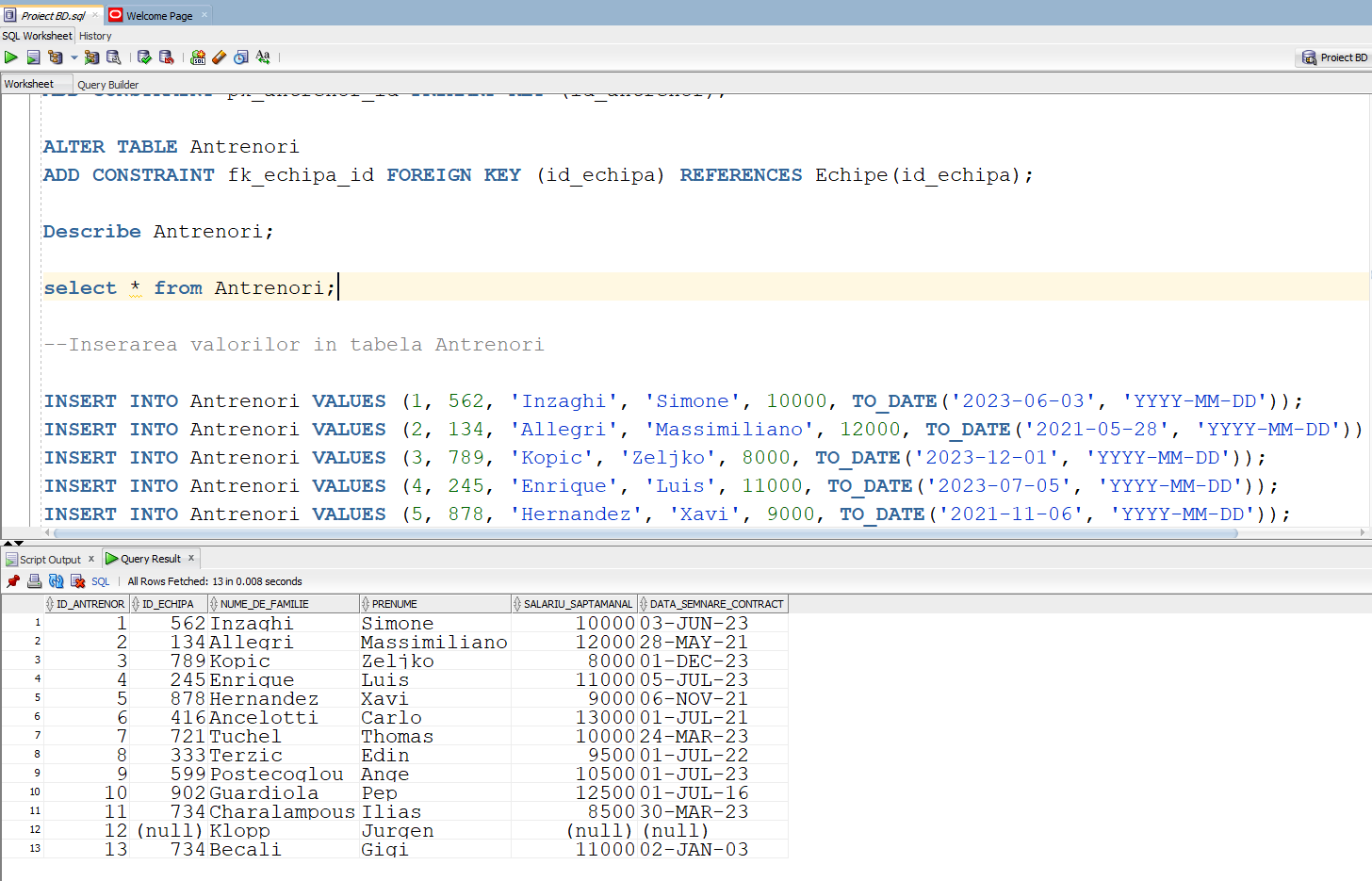
INSERT INTO Antrenori VALUES (10, 902, 'Guardiola', 'Pep', 12500, TO\_DATE('2016-07-01', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (11, 734, 'Charalampous', 'Ilias', 8500, TO\_DATE('2023-03-30', 'YYYY-MM-DD'));

INSERT INTO Antrenori Values (12, NULL, 'Klopp', 'Jurgen', NULL, TO\_DATE('', 'YYYY-MM-DD'));

INSERT INTO Antrenori VALUES (13, 734, 'Becali', 'Gigi', 11000, TO\_DATE('2003-01-02', 'YYYY-MM-DD'));





--crearea tabelei Jucatori

Create Table Jucatori

(

id\_jucator int NOT NULL,

id\_echipa int NOT NULL,

nume\_de\_familie varchar2(30),

prenume varchar2(30),

pozitie varchar2(30),

salariu\_saptamanal int NOT NULL,

aparitii int NULL,

goluri int NULL,

assisturi int NULL,

cota\_de\_piata int NOT NULL,

primary key(id\_jucator),

foreign key(id\_echipa) references Echipe(id\_echipa)

);

alter table Jucatori

drop column pozitie; - aici am realizat ca variabila pozitie incalca FN2

Describe Jucatori;

Select \* from Jucatori;

alter table Jucatori

add data\_semnare\_contract DATE NULL;

--inserarea in tabela

--a jucatorilor de la Inter Milan

INSERT INTO Jucatori VALUES (1, 562, 'Bastoni', 'Alessandro', 'Fundas', 250000, 290, 17, 18, 75000000);

INSERT INTO Jucatori VALUES (2, 562, 'Dumfries', 'Denzel', 'Fundas', 150000, 383, 33, 53, 25000000);

INSERT INTO Jucatori VALUES (3, 562, 'Martinez', 'Lautaro', 'Atacant', 300000, 344, 156, 49, 110000000);

Update Jucatori

Set data\_semnare\_contract = TO\_DATE('2018-07-21', 'YYYY-MM-DD')

where id\_jucator = 1;

Update Jucatori

Set data\_semnare\_contract = TO\_DATE('2020-01-21', 'YYYY-MM-DD')

where id\_jucator = 2;

Update Jucatori

Set data\_semnare\_contract = TO\_DATE('2019-12-21', 'YYYY-MM-DD')

where id\_jucator = 3;

--inserarea in tabela

--a jucatorilor de la Juventus

INSERT INTO Jucatori VALUES

(4, 134, 'Szczesny', 'Wojciech', 'Portar', 175000, 548, NULL, 2, 7000000, TO\_DATE('2015-06-19', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(5, 134, 'Weah', 'Timothy', 'Mijlocas', 95000, 192, 24, 12, 17000000, TO\_DATE('2022-07-28', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(6, 134, 'Vlahovic', 'Dusan', 'Atacant', 250000, 257, 113, 19, 87000000, TO\_DATE('2020-06-19', 'YYYY-MM-DD'));

commit;

--inserarea in tabela

--a jucatorilor de la Dinamo Bucuresti

INSERT INTO Jucatori VALUES

(7, 789, 'Patriche', 'Razvan', 'Fundas', 2000, 100, 2, 12, 350000, TO\_DATE('2022-06-19', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(8, 789, 'Iglesias', 'Dani', 'Mijlocas', 3500, 82, 5, 13, 500000, TO\_DATE('2022-06-28', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(9, 789, 'Politic', 'Dennis', 'Atacant', 250000, 57, 13, 10, 1000000, TO\_DATE('2020-06-19', 'YYYY-MM-DD'));

--inserarea in tabela

--a jucatorilor de la PSG

INSERT INTO Jucatori VALUES

(10, 245, 'Donnarumma', 'Gianluigi', 'Portar', 350000, 400, NULL, 5, 40000000, TO\_DATE('2020-09-14', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(11, 245, 'Mendes', 'Nuno', 'Fundas', 100000, 150, 9, 32, 60000000, TO\_DATE('2021-07-28', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(12, 245, 'Mbappe', 'Kylian', 'Banda', 15000000, 372, 288, 126, 190000000, TO\_DATE('2015-12-08', 'YYYY-MM-DD'));

--inserarea in tabela

--a jucatorilor de la Barcelona 878

INSERT INTO Jucatori VALUES

(13, 878, 'Ter Stegen', 'Marc-Andre', 'Portar', 1350000, 680, 1, 9, 50000000, TO\_DATE('2014-09-15', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(14, 878, 'Torres', 'Ferran', 'Banda', 200000, 163, 19, 42, 50000000, TO\_DATE('2022-08-28', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(15, 878, 'Lewandowski', 'Robert', 'Atacant', 9000000, 770, 567, 153, 15000000, TO\_DATE('2015-11-08', 'YYYY-MM-DD'));

--inserarea in tabela

--a celorlalti jucatori

INSERT INTO Jucatori VALUES

(16, 416, 'Junior', 'Vinicius', 'Banda', 2500000, 348, 109, 112, 120000000, TO\_DATE('2019-12-24', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(17, 721, 'Kimmich', 'Joshua', 'Fundas', 150000, 427, 87, 200, 80000000, TO\_DATE('2018-06-13', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(18, 333, 'Reus', 'Marco', 'Mijlocas', 100000, 789, 135, 350, 10000000, TO\_DATE('2009-12-30', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(19, 599, 'Son', 'Heung-Min', 'Banda', 200000, 570, 225, 113, 70000000, TO\_DATE('2014-08-22', 'YYYY-MM-DD'));

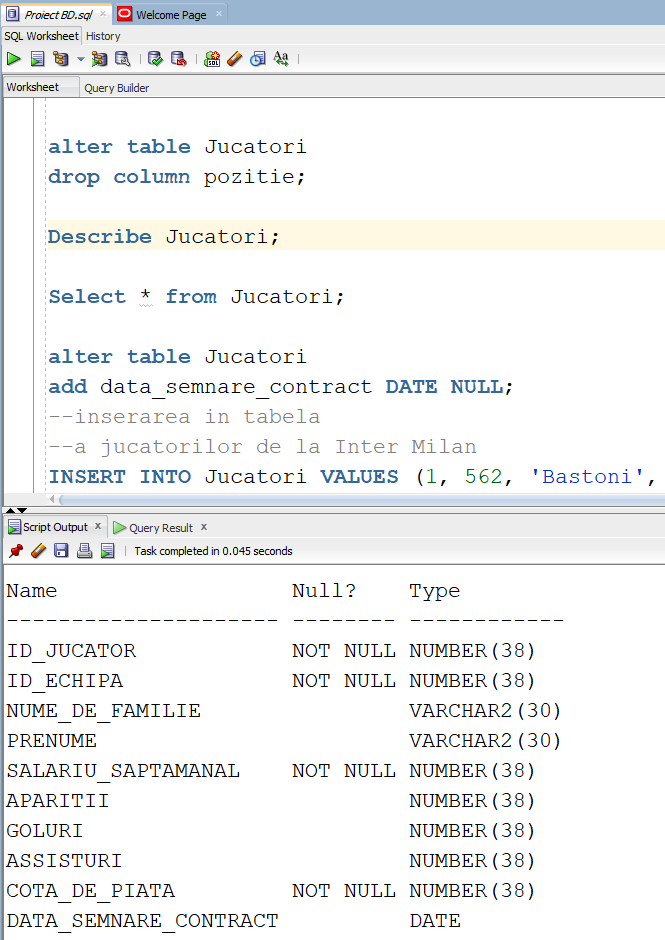
INSERT INTO Jucatori VALUES

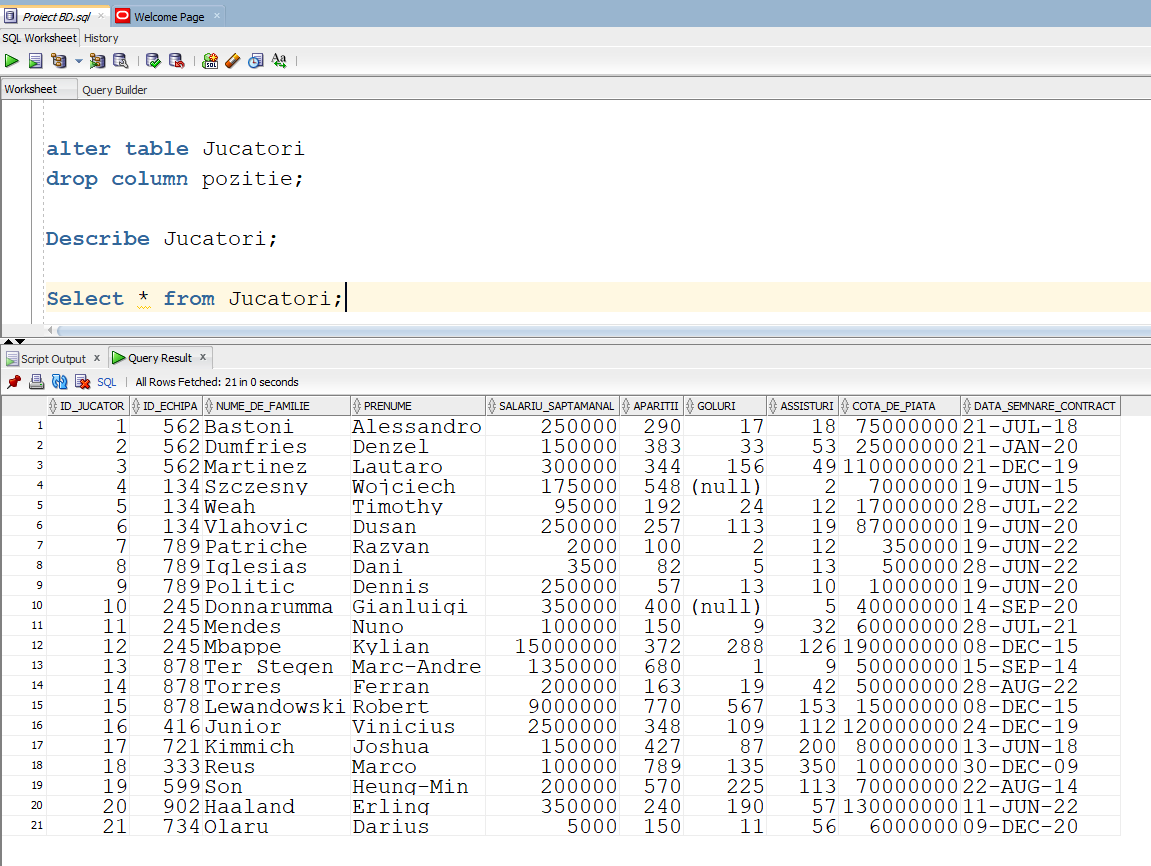
(20, 902, 'Haaland', 'Erling', 'Atacant', 350000, 240, 190, 57, 130000000, TO\_DATE('2022-06-11', 'YYYY-MM-DD'));

INSERT INTO Jucatori VALUES

(21, 734, 'Olaru', 'Darius', 'Mijlocas', 5000, 150, 11, 56, 6000000, TO\_DATE('2020-12-09', 'YYYY-MM-DD'));

commit;





--crearea tabelei Pozitii

Create Table Pozitii

(

id\_pozitie int NOT NULL,

pozitie varchar(20),

primary key (id\_pozitie)

);

Describe Pozitii;

--inserarea valorilor in tabela Pozitii

select \* from Pozitii;

INSERT INTO Pozitii VALUES

(1, 'Portar');

INSERT INTO Pozitii VALUES

(2, 'Fundas');

INSERT INTO Pozitii VALUES

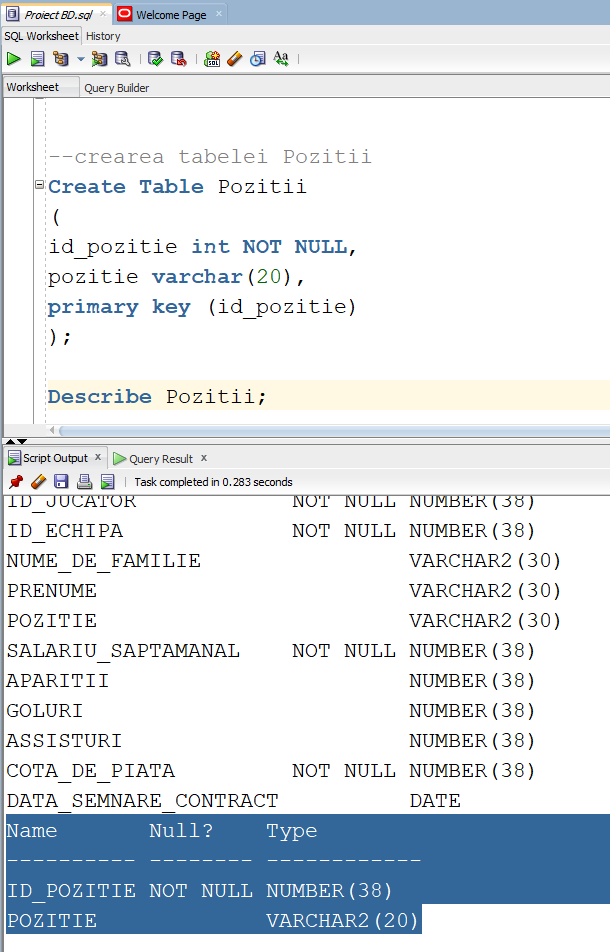
(3, 'Mijlocas');

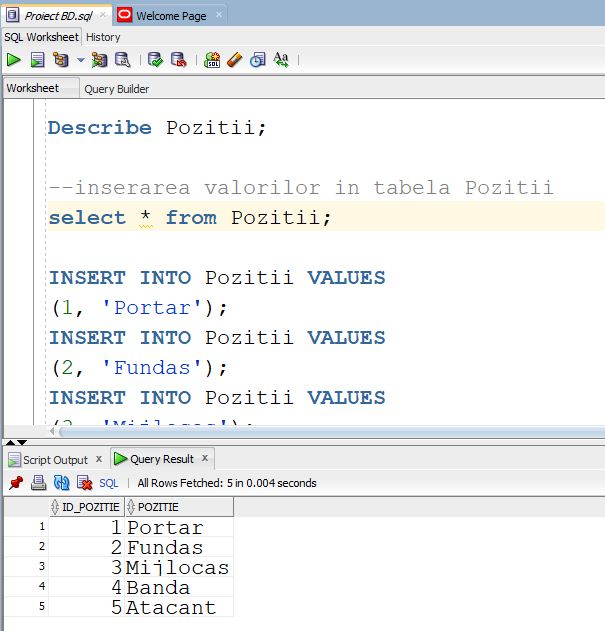
INSERT INTO Pozitii VALUES

(4, 'Banda');

INSERT INTO Pozitii VALUES

(5, 'Atacant');





--crearea tabelei asociative dintre tabela jucatori si tabela pozitii, aflate in relatie Many to Many

CREATE TABLE Jucatori\_Pozitii (

id\_jucator int NOT NULL,

id\_pozitie iNT NOT NULL,

primary key (id\_jucator, id\_pozitie),

foreign key (id\_jucator) references Jucatori(id\_jucator),

foreign key (id\_pozitie) references Pozitii(id\_pozitie)

);

Describe Jucatori\_Pozitii;

Select \* from Jucatori\_Pozitii;

--inserarea valorilor in tabela asociativa Jucatori\_Pozitii

Insert INTO Jucatori\_Pozitii VALUES (1, 2);

Insert INTO Jucatori\_Pozitii VALUES (2, 2);

Insert INTO Jucatori\_Pozitii VALUES (3, 4);

Insert INTO Jucatori\_Pozitii VALUES (3, 5);

Insert INTO Jucatori\_Pozitii VALUES (4, 1);

Insert INTO Jucatori\_Pozitii VALUES (5, 3);

Insert INTO Jucatori\_Pozitii VALUES (5, 4);

Insert INTO Jucatori\_Pozitii VALUES (6, 5);

Insert INTO Jucatori\_Pozitii VALUES (7, 2);

Insert INTO Jucatori\_Pozitii VALUES (8, 3);

Insert INTO Jucatori\_Pozitii VALUES (9, 4);

Insert INTO Jucatori\_Pozitii VALUES (9, 5);

Insert INTO Jucatori\_Pozitii VALUES (10, 1);

Insert INTO Jucatori\_Pozitii VALUES (11, 2);

Insert INTO Jucatori\_Pozitii VALUES (12, 4);

Insert INTO Jucatori\_Pozitii VALUES (12, 5);

Insert INTO Jucatori\_Pozitii VALUES (13, 1);

Insert INTO Jucatori\_Pozitii VALUES (14, 4);

Insert INTO Jucatori\_Pozitii VALUES (14, 5);

Insert INTO Jucatori\_Pozitii VALUES (15, 5);

Insert INTO Jucatori\_Pozitii VALUES (16, 4);

Insert INTO Jucatori\_Pozitii VALUES (16, 5);

Insert INTO Jucatori\_Pozitii VALUES (17, 2);

Insert INTO Jucatori\_Pozitii VALUES (17, 3);

Insert INTO Jucatori\_Pozitii VALUES (18, 3);

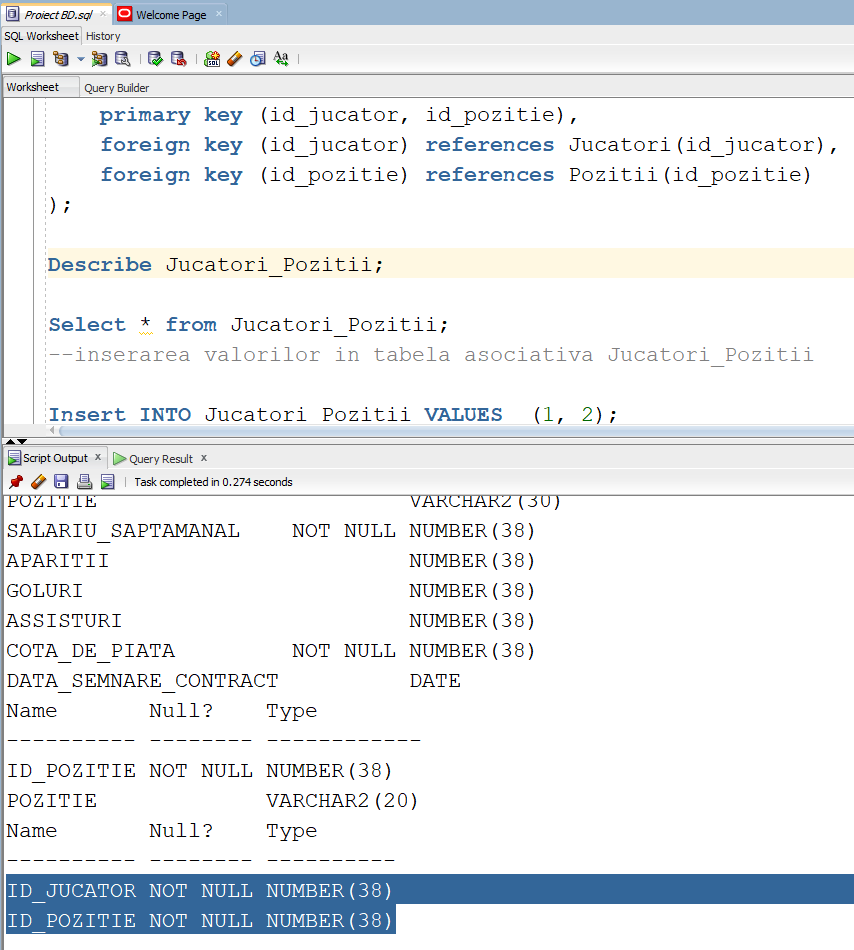
Insert INTO Jucatori\_Pozitii VALUES (18, 4);

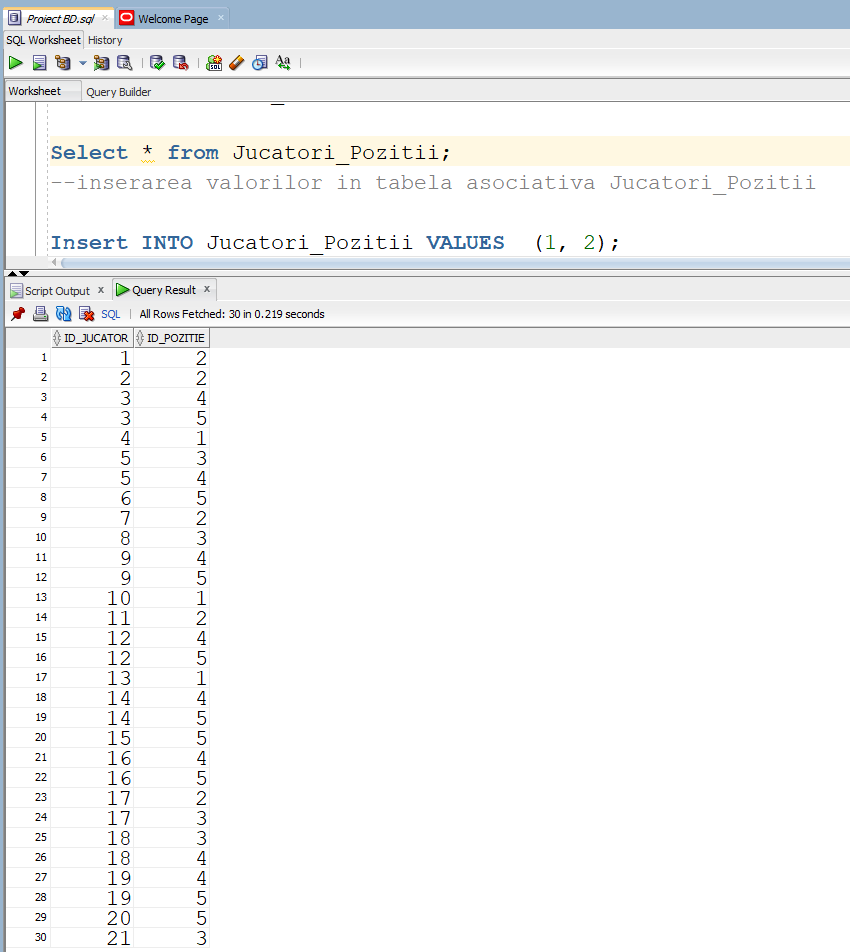
Insert INTO Jucatori\_Pozitii VALUES (19, 4);

Insert INTO Jucatori\_Pozitii VALUES (19, 5);

Insert INTO Jucatori\_Pozitii VALUES (20, 5);

Insert INTO Jucatori\_Pozitii VALUES (21, 3);





--crearea tabelei Partide

Create Table Partide

(

id\_partida int NOT NULL,

id\_gazde int NOT NULL,

id\_oaspeti int NOT NULL,

numar\_spectatori int NOT NULL,

goluri\_gazde int NULL,

goluri\_oaspeti int NULL,

primary key(id\_partida),

foreign key(id\_gazde) references Echipe(id\_echipa),

foreign key(id\_oaspeti) references Echipe(id\_echipa)

);

Describe Partide;

select \* from Partide;

--inserarea valorilor in tabela Partide

INSERT INTO Partide VALUES (1, 562, 134, 50000, 2, 1);

INSERT INTO Partide VALUES (2, 789, 245, 30000, 1, 3);

INSERT INTO Partide VALUES (3, 878, 416, 80000, 1, 1);

INSERT INTO Partide VALUES (4, 721, 333, 75000, 4, 2);

INSERT INTO Partide VALUES (5, 599, 902, 62000, 2, 3);

INSERT INTO Partide VALUES (6, 134, 789, 40000, 1, 0);

INSERT INTO Partide VALUES (7, 245, 878, 50000, 3, 2);

INSERT INTO Partide VALUES (8, 416, 721, 70000, 2, 2);

INSERT INTO Partide VALUES (9, 333, 599, 40000, 1, 3);

INSERT INTO Partide VALUES (10, 902, 562, 58000, 2, 2);

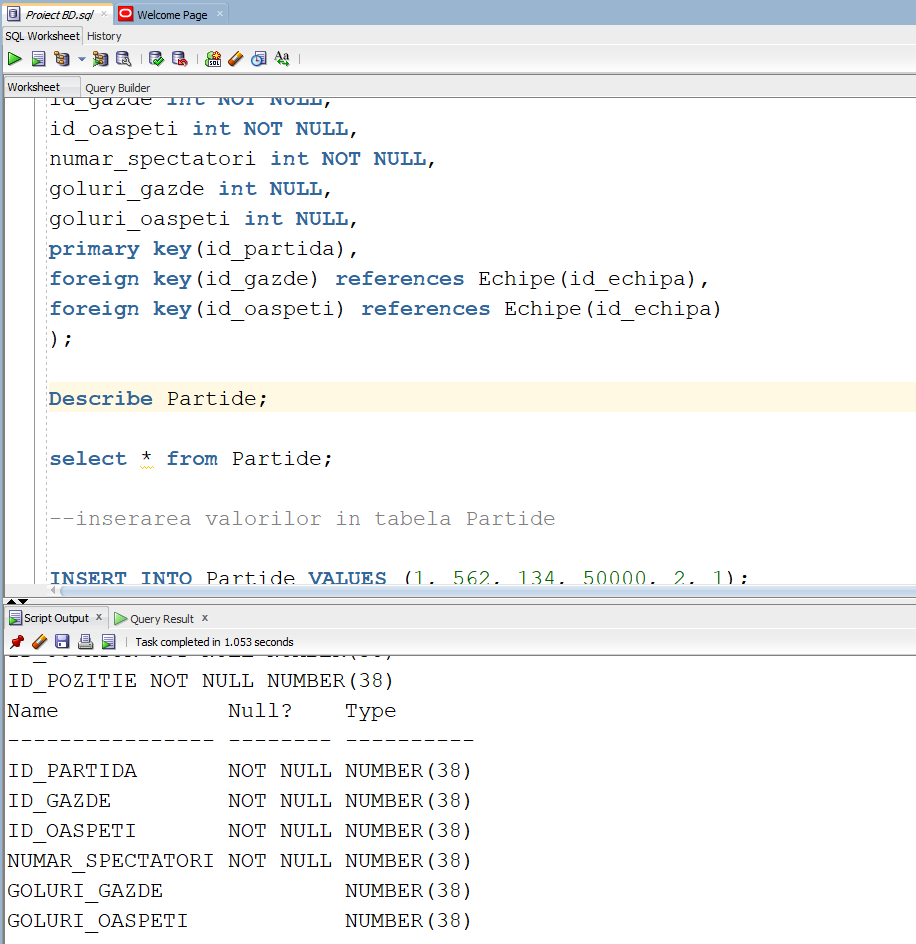
INSERT INTO Partide VALUES (11, 134, 878, 45000, 0, 1);

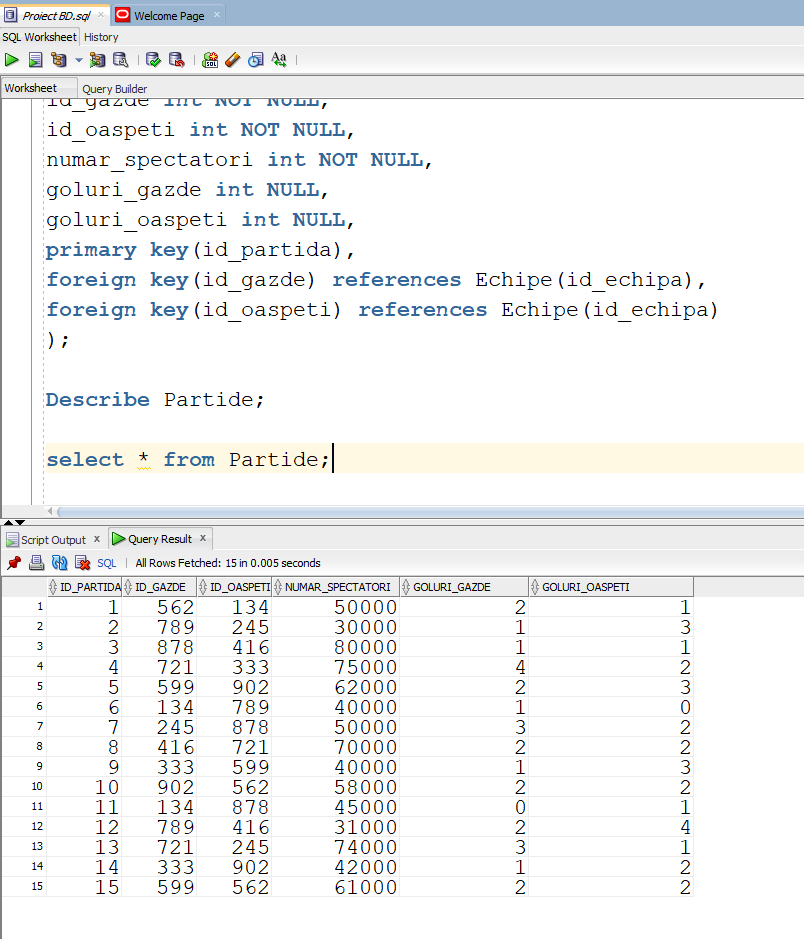
INSERT INTO Partide VALUES (12, 789, 416, 31000, 2, 4);

INSERT INTO Partide VALUES (13, 721, 245, 74000, 3, 1);

INSERT INTO Partide VALUES (14, 333, 902, 42000, 1, 2);

INSERT INTO Partide VALUES (15, 599, 562, 61000, 2, 2);





--crearea tabelei Arbitri

Create Table Arbitri

(

id\_arbitru int NOT NULL,

nume varchar(30),

prenume varchar(30)

);

Describe arbitri;

Select \* from arbitri;

ALTER TABLE Arbitri

ADD CONSTRAINT pk\_arbitru\_id PRIMARY KEY (id\_arbitru);

--inserarea valorilor in tabela Arbitri

INSERT INTO Arbitri VALUES (1, 'Popescu', 'Ion');

INSERT INTO Arbitri VALUES (2, 'Ionescu', 'Mihai');

INSERT INTO Arbitri VALUES (3, 'Vasilescu', 'Andrei');

INSERT INTO Arbitri VALUES (4, 'Georgescu', 'Alexandru');

INSERT INTO Arbitri VALUES (5, 'Dumitrescu', 'Florin');

INSERT INTO Arbitri VALUES (6, 'Marinescu', 'Cristian');

INSERT INTO Arbitri VALUES (7, 'Stanescu', 'Gabriel');

INSERT INTO Arbitri VALUES (8, 'Radu', 'Stefan');

INSERT INTO Arbitri VALUES (9, 'Serban', 'Daniel');

INSERT INTO Arbitri VALUES (10, 'Petrescu', 'Radu');

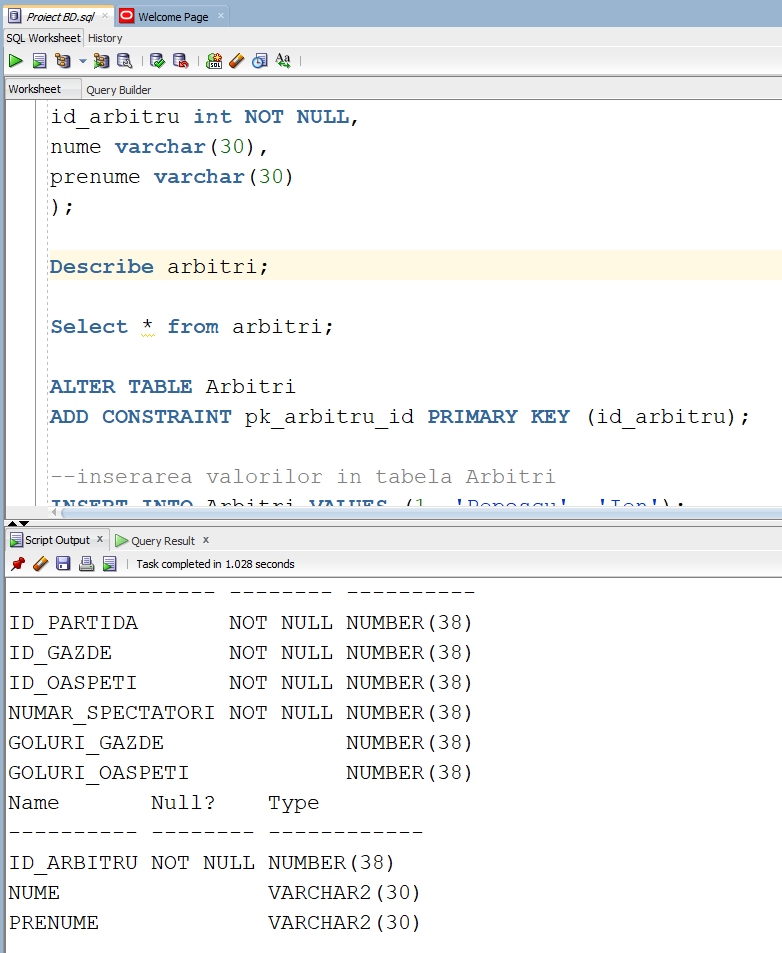
INSERT INTO Arbitri VALUES (11, 'Nistor', 'Alin');

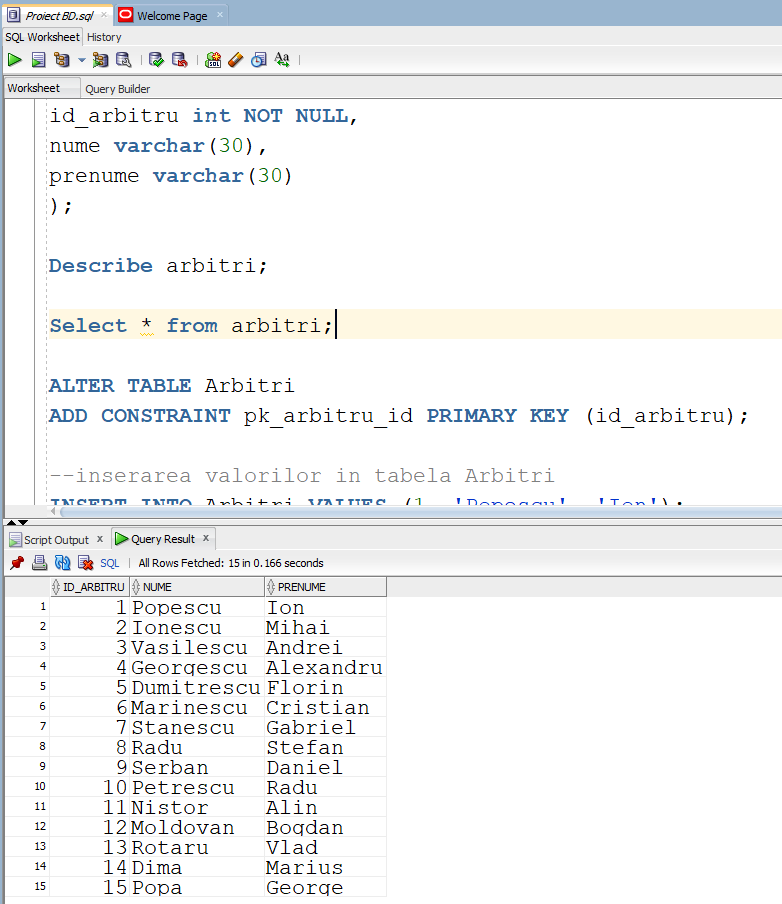
INSERT INTO Arbitri VALUES (12, 'Moldovan', 'Bogdan');

INSERT INTO Arbitri VALUES (13, 'Rotaru', 'Vlad');

INSERT INTO Arbitri VALUES (14, 'Dima', 'Marius');

INSERT INTO Arbitri VALUES (15, 'Popa', 'George');





--crearea tabelei asociative dintre partide si arbitri, aflate in relatie Many to Many

CREATE TABLE Partide\_Arbitri (

id\_partida int NOT NULL,

id\_arbitru int NOT NULL,

primary key (id\_partida, id\_arbitru),

foreign key (id\_partida) references Partide(id\_partida),

foreign key (id\_arbitru) references Arbitri(id\_arbitru)

);

Describe Partide\_Arbitri;

select \* from Partide\_Arbitri;

--inserarea valorilor in tabela asociativa

INSERT INTO Partide\_Arbitri VALUES (1, 1);

INSERT INTO Partide\_Arbitri VALUES (1, 2);

INSERT INTO Partide\_Arbitri VALUES (2, 3);

INSERT INTO Partide\_Arbitri VALUES (2, 4);

INSERT INTO Partide\_Arbitri VALUES (3, 5);

INSERT INTO Partide\_Arbitri VALUES (3, 6);

INSERT INTO Partide\_Arbitri VALUES (4, 7);

INSERT INTO Partide\_Arbitri VALUES (4, 8);

INSERT INTO Partide\_Arbitri VALUES (5, 9);

INSERT INTO Partide\_Arbitri VALUES (5, 10);

INSERT INTO Partide\_Arbitri VALUES (6, 11);

INSERT INTO Partide\_Arbitri VALUES (6, 12);

INSERT INTO Partide\_Arbitri VALUES (7, 13);

INSERT INTO Partide\_Arbitri VALUES (7, 14);

INSERT INTO Partide\_Arbitri VALUES (8, 15);

INSERT INTO Partide\_Arbitri VALUES (8, 1);

INSERT INTO Partide\_Arbitri VALUES (9, 2);

INSERT INTO Partide\_Arbitri VALUES (9, 3);

INSERT INTO Partide\_Arbitri VALUES (10, 4);

INSERT INTO Partide\_Arbitri VALUES (10, 5);

INSERT INTO Partide\_Arbitri VALUES (11, 6);

INSERT INTO Partide\_Arbitri VALUES (11, 7);

INSERT INTO Partide\_Arbitri VALUES (12, 8);

INSERT INTO Partide\_Arbitri VALUES (12, 9);

INSERT INTO Partide\_Arbitri VALUES (13, 10);

INSERT INTO Partide\_Arbitri VALUES (13, 11);

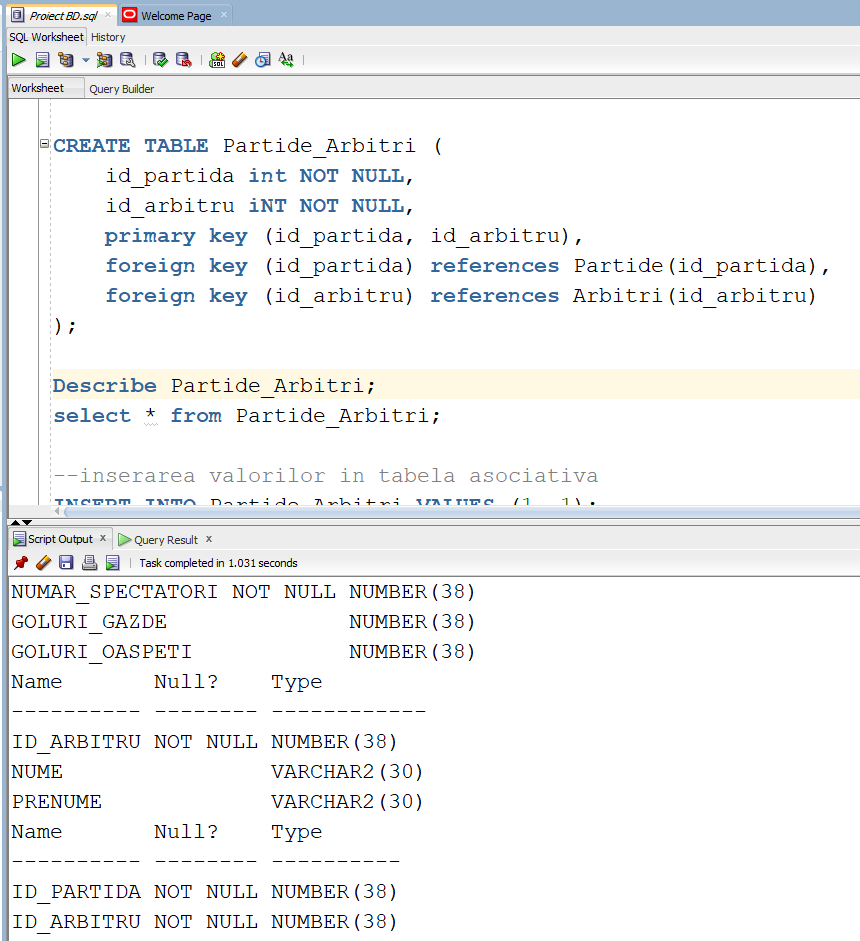
INSERT INTO Partide\_Arbitri VALUES (14, 12);

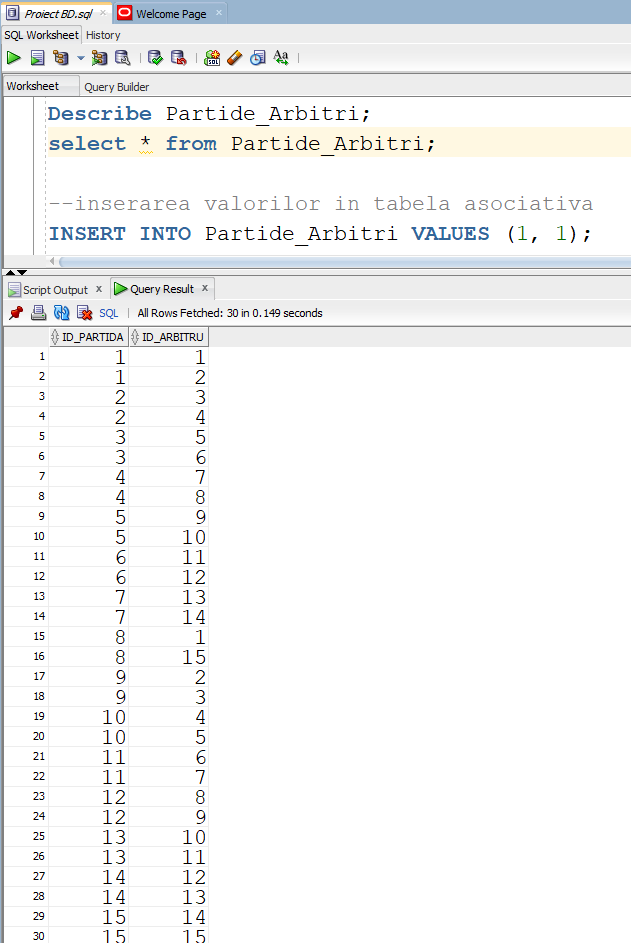
INSERT INTO Partide\_Arbitri VALUES (14, 13);

INSERT INTO Partide\_Arbitri VALUES (15, 14);

INSERT INTO Partide\_Arbitri VALUES (15, 15);

commit;





**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe**

--a) subcerere sincronizata

--Afisati numele echipelor care au castigat macar o partida din postura de gazde

--la care a luat parte din brigada de arbitri Popescu Ion

select e.nume\_echipa

from Echipe e

where e.id\_echipa in

(select id\_gazde from Partide p, Partide\_Arbitri pa, Arbitri a

where p.id\_gazde = e.id\_echipa

and p.id\_partida = pa.id\_partida

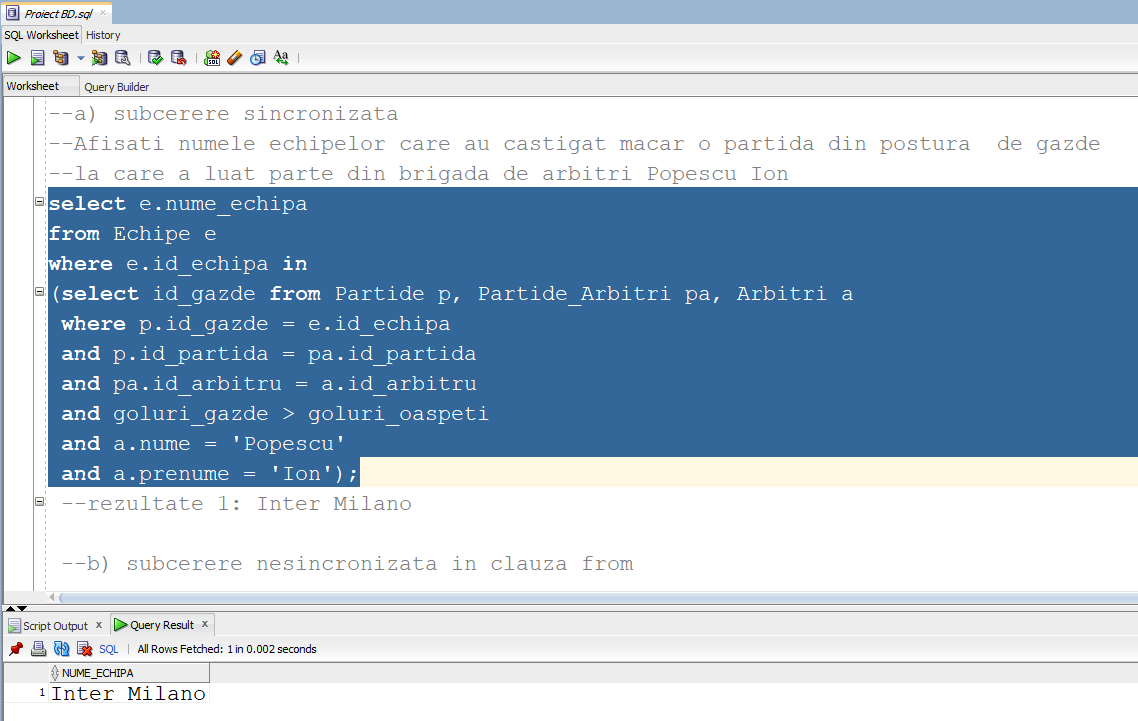
and pa.id\_arbitru = a.id\_arbitru

and goluri\_gazde > goluri\_oaspeti

and a.nume = 'Popescu'

and a.prenume = 'Ion');

--rezultate 1: Inter Milano

****

--b) subcerere nesincronizata in clauza from

--Afisati tipul partidelor ("cu numar mic de spectatori" <=50000,

--50000 < "cu numar mediu de spectatori" < 65000

--65000 <= "cu numar mare de spectatori

--in care echipa gazda a inscris 2 goluri,

--si numele echipei gazde

SELECT

partide2.id\_partida, e.nume\_echipa as "Echipa Gazda",

DECODE(SIGN(numar\_spectatori - 50000),

-1, 'cu numar mic de spectatori',

SIGN(numar\_spectatori - 65000),

'cu numar mediu de spectatori',

'cu numar mare de spectatori') AS tip\_partida

FROM

(SELECT p.numar\_spectatori, p.id\_gazde, p.id\_partida

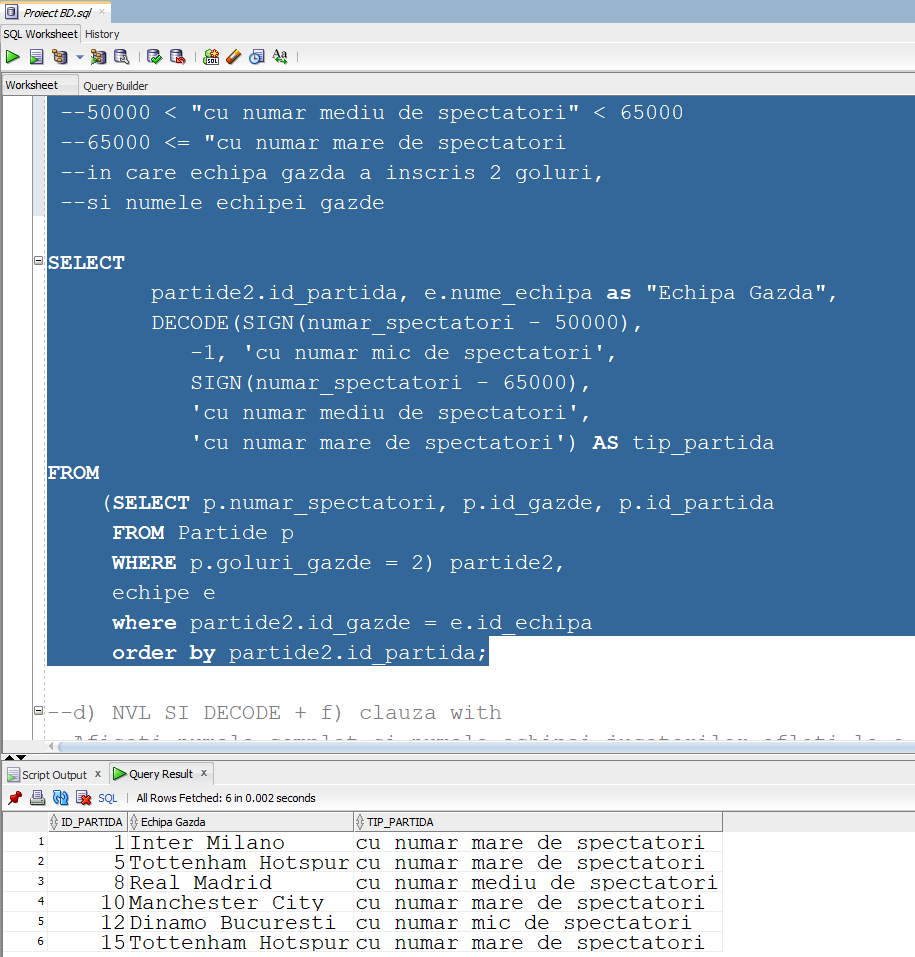
FROM Partide p

WHERE p.goluri\_gazde = 2) partide2,

echipe e

where partide2.id\_gazde = e.id\_echipa

order by partide2.id\_partida;



--d) NVL SI DECODE + f) clauza with

--Afisati numele complet si numele echipei jucatorilor aflati la o echipa antrenata de

--macar un antrenor platit peste media salariala a tuturor antrenorilor

--De asemenea, pentru fiecare jucator sa se afiseze daca a inscris mai multe sau mai putine goluri

--decat media tuturor jucatorilor inscrisi in baza de date.

WITH MedieDeGoluri AS (

SELECT ROUND(AVG(goluri)) AS medie FROM Jucatori

),

MedieSalariu AS (

SELECT ROUND(AVG(salariu\_saptamanal)) as salariu FROM Antrenori

)

--select \* from MedieDeGoluri;

--select \* from MedieSalariu;

SELECT j.prenume || ' ' || j.nume\_de\_familie as "Nume complet", e.nume\_echipa,

DECODE(SIGN(NVL(goluri, 0) - (select medie from MedieDeGoluri)),

-1, 'mai putine goluri decat media',

'mai multe goluri decat media') AS "A inscris"

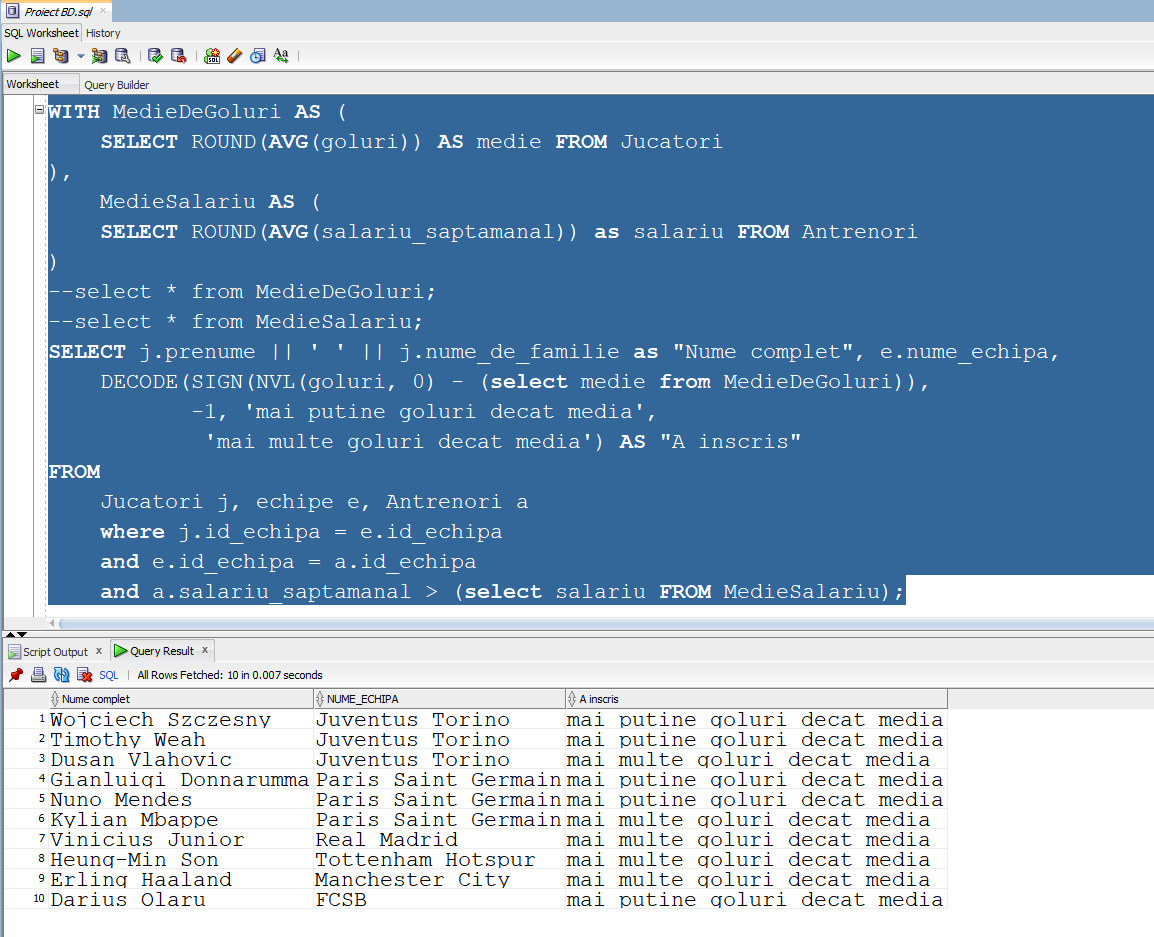
FROM

Jucatori j, echipe e, Antrenori a

where j.id\_echipa = e.id\_echipa

and e.id\_echipa = a.id\_echipa

and a.salariu\_saptamanal > (select salariu FROM MedieSalariu);



--e) Afisati daca exista sau nu macar un jucator care joaca pe o pozitie care se termina in `as`

--in cazul de fata: fundas/mijlocas, care si-a semnat contractul pe timpul verii

--la o echipa al carei antrenor se afla in functie mai tarziu de 1 ianuarie 2021.

Select CASE

when count(distinct j.id\_jucator) > 0 then 'exista'

else 'nu exista'

end as "Exista" --functia CASE

from Jucatori\_Pozitii jp, Jucatori j, Pozitii p

where jp.id\_jucator = j.id\_jucator

and jp.id\_pozitie = p.id\_pozitie

and upper(p.pozitie) like '%AS' --functiile 1-2 pe siruri de caractere

and j.id\_jucator in

(select j2.id\_jucator

from Jucatori j2, Echipe e, Antrenori a

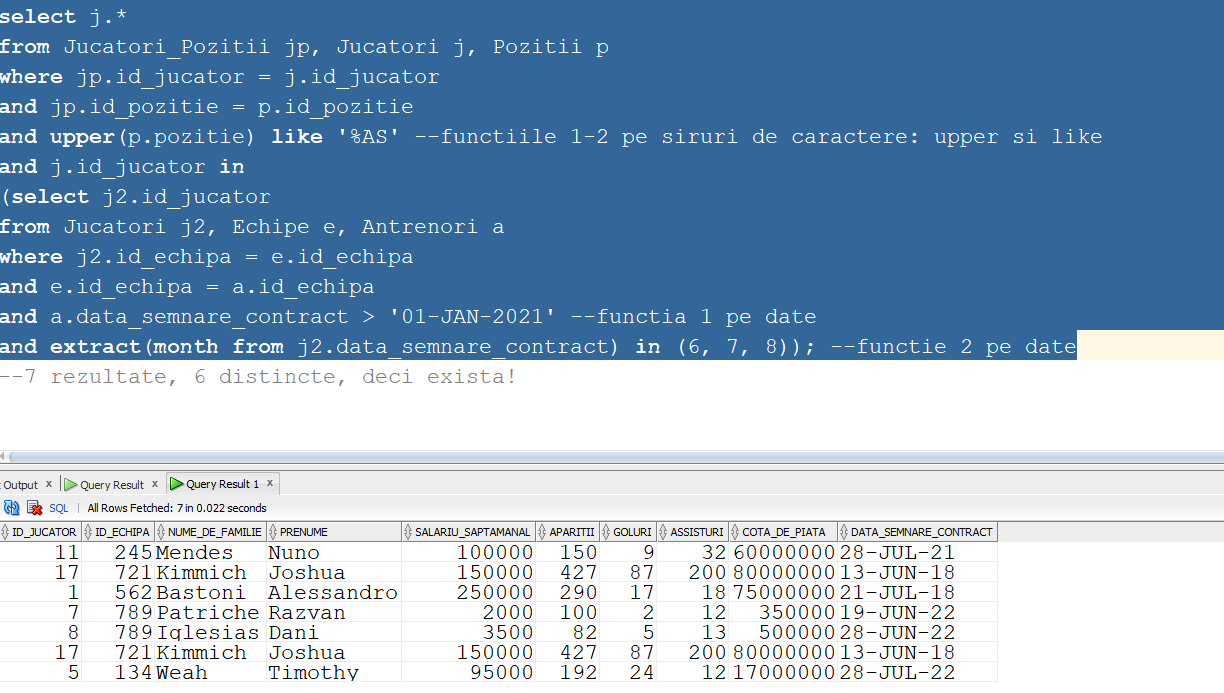
where j2.id\_echipa = e.id\_echipa

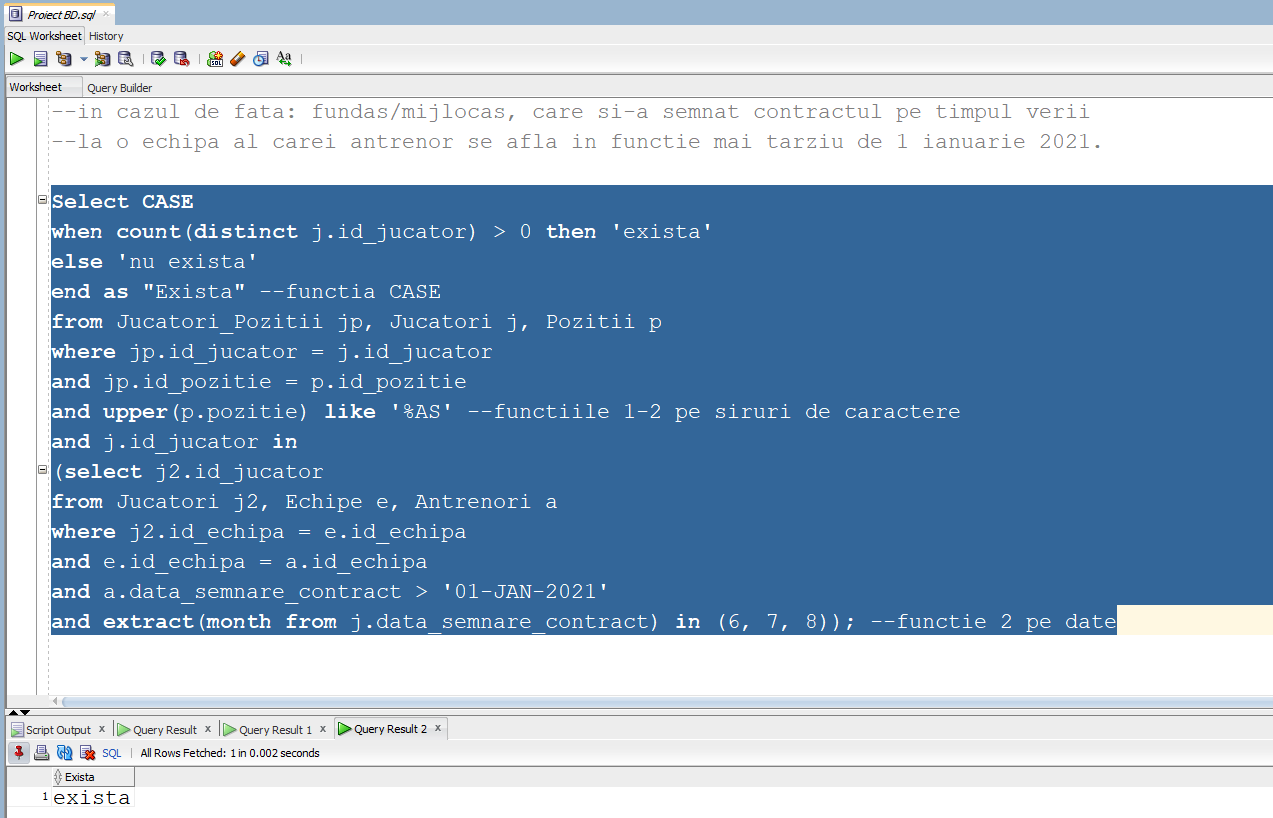
and e.id\_echipa = a.id\_echipa

and a.data\_semnare\_contract > '01-JAN-2021'

and extract(month from j.data\_semnare\_contract) in (6, 7, 8)); --functie 2 pe date

--verificarea subcererii, pentru a ne asigura daca exista sau nu, urmand ca mai apoi sa verificam case-ul





--c) Sa se afiseze echipele englezesti care depasesc media din punct de vedere al

--numarului total de goluri inscrise (sa fie afisata si aceasta valoare)

SELECT e.id\_echipa, e.nume\_echipa, SUM(j.goluri) AS goluri\_totale

FROM Jucatori j, Echipe e, Campionate c, Tari t

where j.id\_echipa = e.id\_echipa

and e.id\_campionat = c.id\_campionat

and c.id\_tara = t.id\_tara

and t.nume = 'Anglia'

GROUP BY e.id\_echipa, e.nume\_echipa

HAVING SUM(j.goluri) > (

SELECT AVG(goluri\_per\_echipa)

FROM (

SELECT j2.id\_echipa, SUM(j2.goluri) AS goluri\_per\_echipa

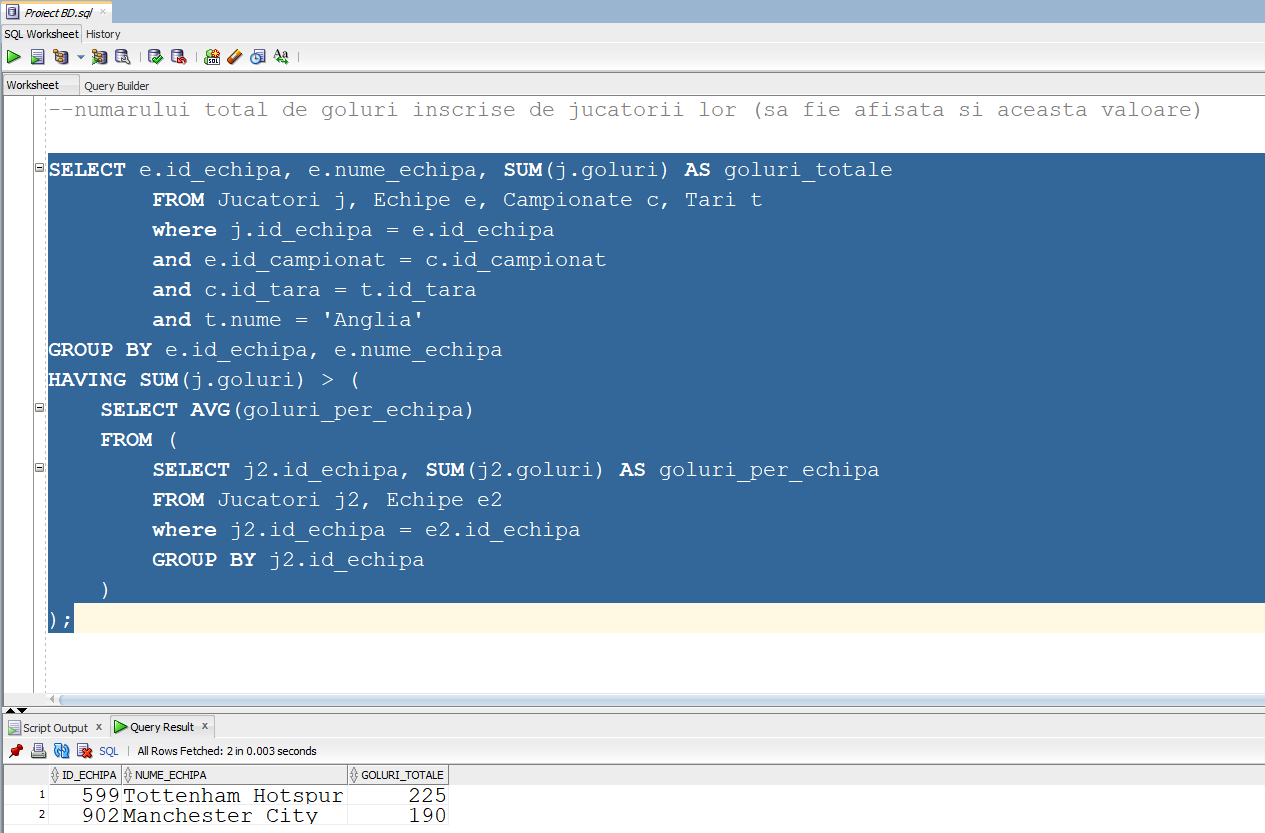
FROM Jucatori j2, Echipe e2

where j2.id\_echipa = e2.id\_echipa

GROUP BY j2.id\_echipa

)

);



**13. Implementarea a 3 operații de actualizare**

**și de suprimare a datelor utilizând subcereri**

--a)

--Nota: În semn de recunoștință, vom mări salariile jucătorilor care au marcat peste 500 de goluri în cariera lor cu 10 procente.

Update jucatori

SET salariu\_saptamanal = salariu\_saptamanal \* 1.10

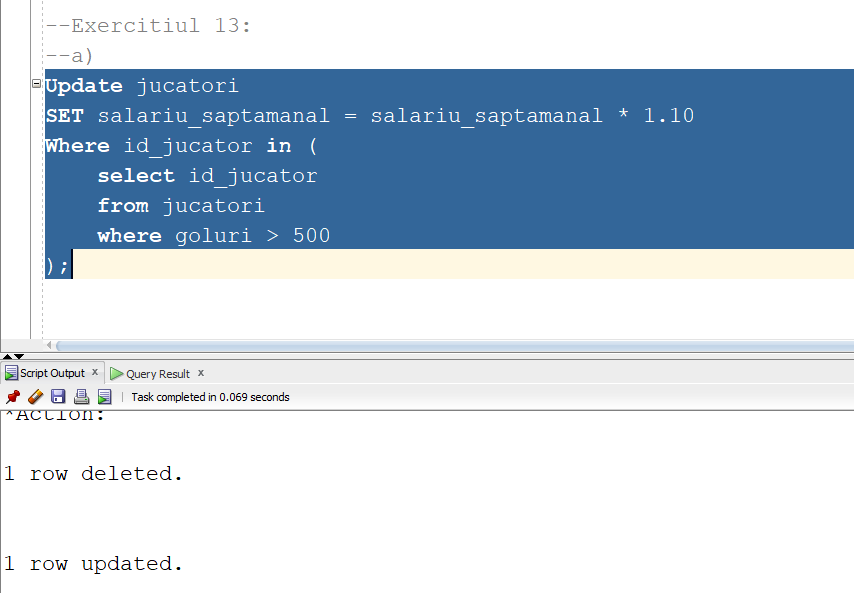
Where id\_jucator in (

select id\_jucator

from jucatori

where goluri > 500

);



--Doar jucatorul Robert Lewandowski are peste 500 goluri, de aici si 1 row updated.

--b)

INSERT INTO Jucatori VALUES

(22, 599, 'Andrei', 'Andreius', 35000000, 450, 126, 456, 236000000, TO\_DATE('2020-05-30', 'YYYY-MM-DD'));

select \* from Jucatori;

--Nota: Să presupunem că echipa cu id-ul 599 a semnat cu jucătorul Andrei Andreius, însă salariul său săptămânal este prea mare, așa că au ajuns la un acord comun, micșorându-și pretențiile salariale.

Update jucatori

Set salariu\_saptamanal = 3500000

where id\_jucator in

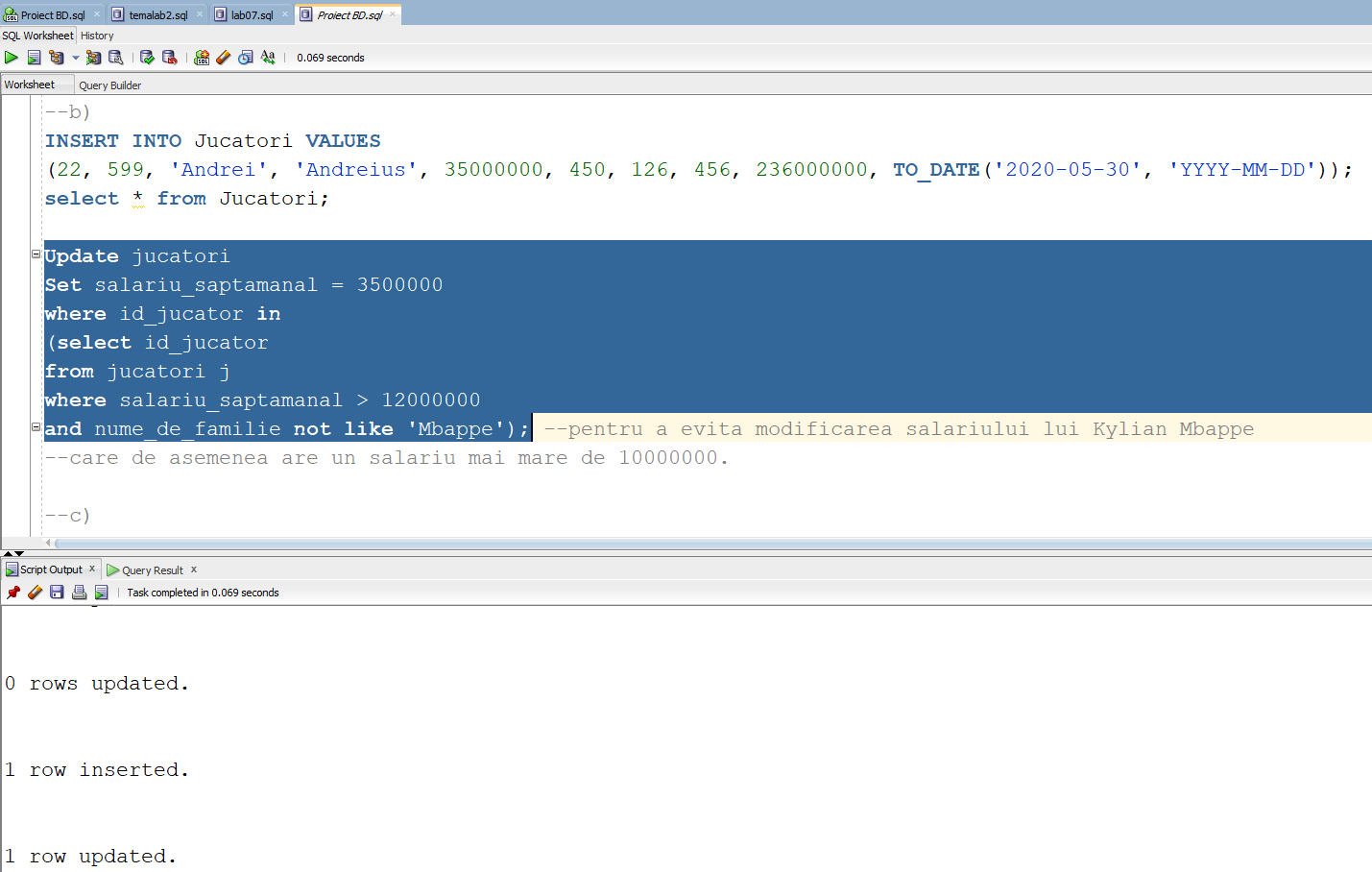
(select id\_jucator

from jucatori j

where salariu\_saptamanal > 12000000

and nume\_de\_familie not like 'Mbappe'); --pentru a evita modificarea salariului lui Kylian Mbappe

--care de asemenea are un salariu mai mare de 12000000.



--c)

INSERT INTO Jucatori VALUES

(22, 599, 'Andrei', 'Andreius', 35000000, 450, 126, 456, 236000000, TO\_DATE('2020-05-30', 'YYYY-MM-DD'));

select \* from Jucatori;

Delete from jucatori

where id\_jucator in (select id\_jucator

from jucatori j, echipe e, stadioane s, orase o

where j.id\_echipa = e.id\_echipa

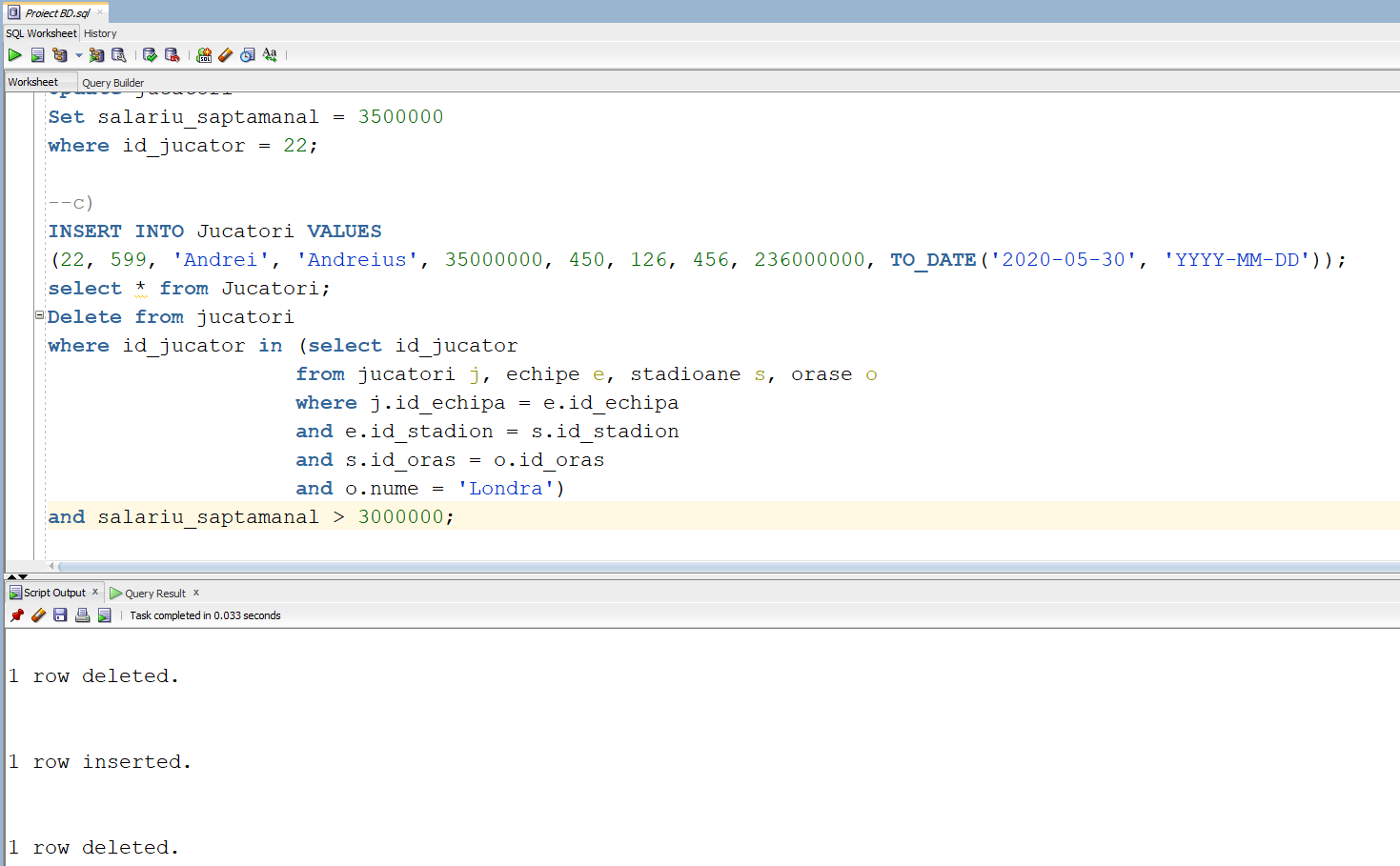
and e.id\_stadion = s.id\_stadion

and s.id\_oras = o.id\_oras

and o.nume = 'Londra')

and salariu\_saptamanal > 3000000;

--Nota: Rezilierea contractului a fost, până la urmă, inevitabilă, dar delete-ul va avea loc cu ajutorulu unei cereri cu join-uri (”Să se șteargă jucătorul care se află la o echipă din Londra și care are salariul săptămânal peste 3000000).



**14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe**

**vizualizarea respectivă și un exemplu de operație LMD nepermisă.**

--exercitiul 14.1: O vizualizare care afișează detalii despre jucători, despre echipa de club, stadion și antrenor.

Create view Jucatori\_Detalii as

select distinct

j.id\_jucator,

j.nume\_de\_familie NUME,

j.prenume PRENUME,

e.nume\_echipa ECHIPA,

s.nume\_stadion NUME\_STADION,

a.nume\_de\_familie || ' ' || a.prenume NUME\_ANTRENOR

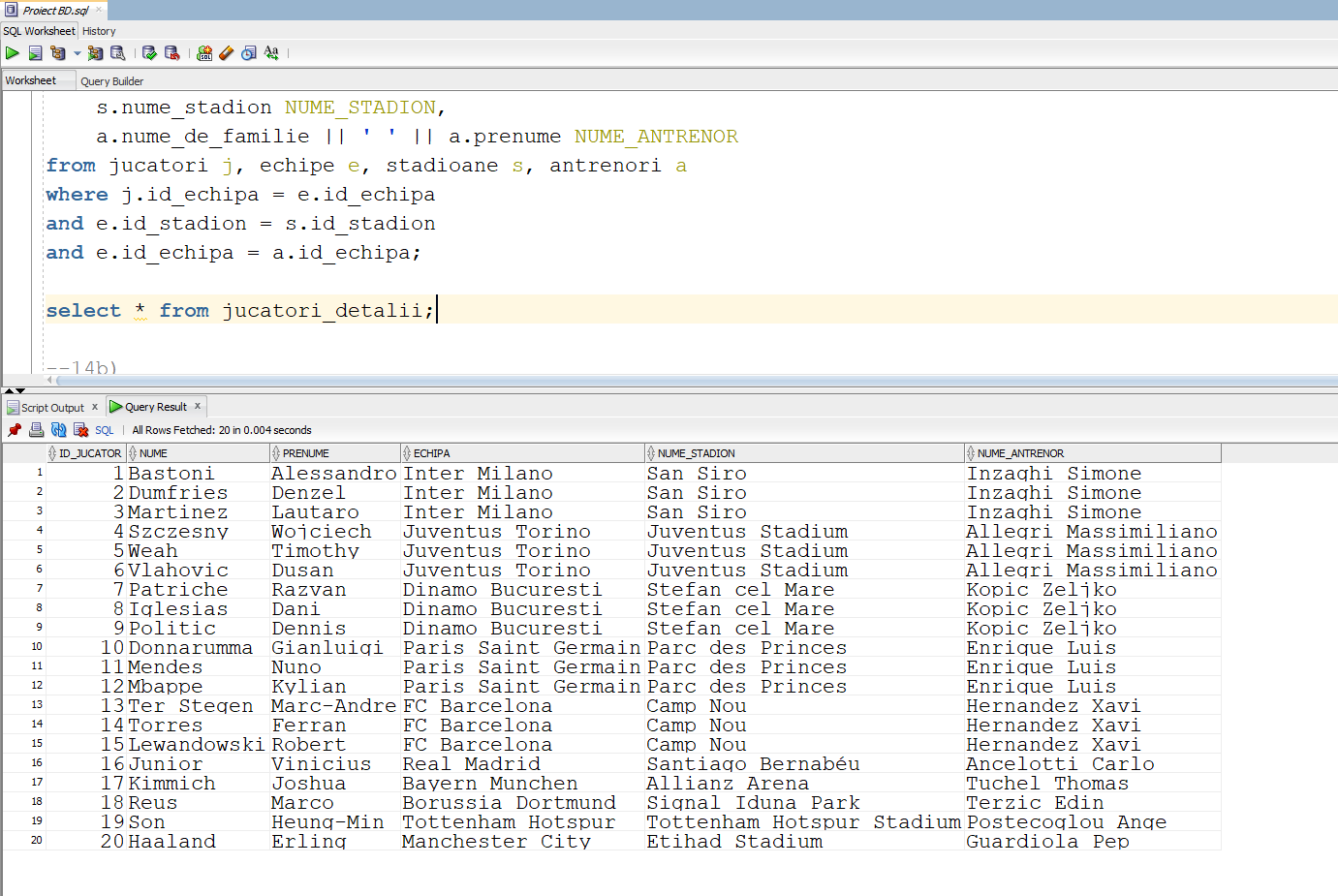
from jucatori j, echipe e, stadioane s, antrenori a

where j.id\_echipa = e.id\_echipa

and e.id\_stadion = s.id\_stadion

and e.id\_echipa = a.id\_echipa;

select \* from jucatori\_detalii;



--exercițiul 14.2: Sa se afiseze toti jucatorii antrenati de Luis Enrique sau de Simone Inzaghi

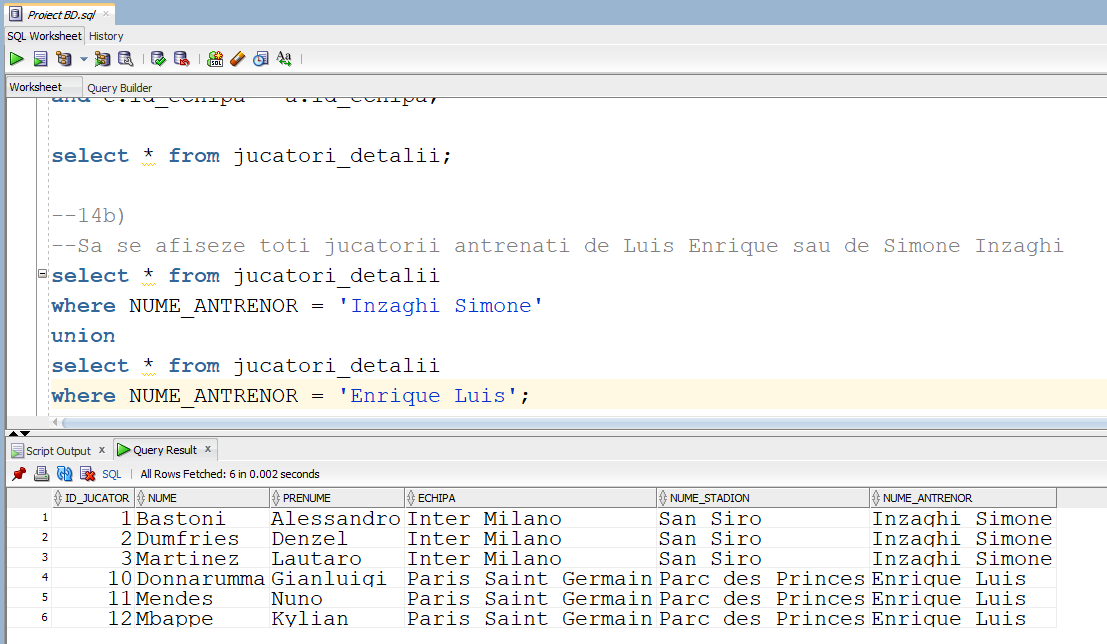
select \* from jucatori\_detalii

where NUME\_ANTRENOR = 'Inzaghi Simone'

union

select \* from jucatori\_detalii

where NUME\_ANTRENOR = 'Enrique Luis';

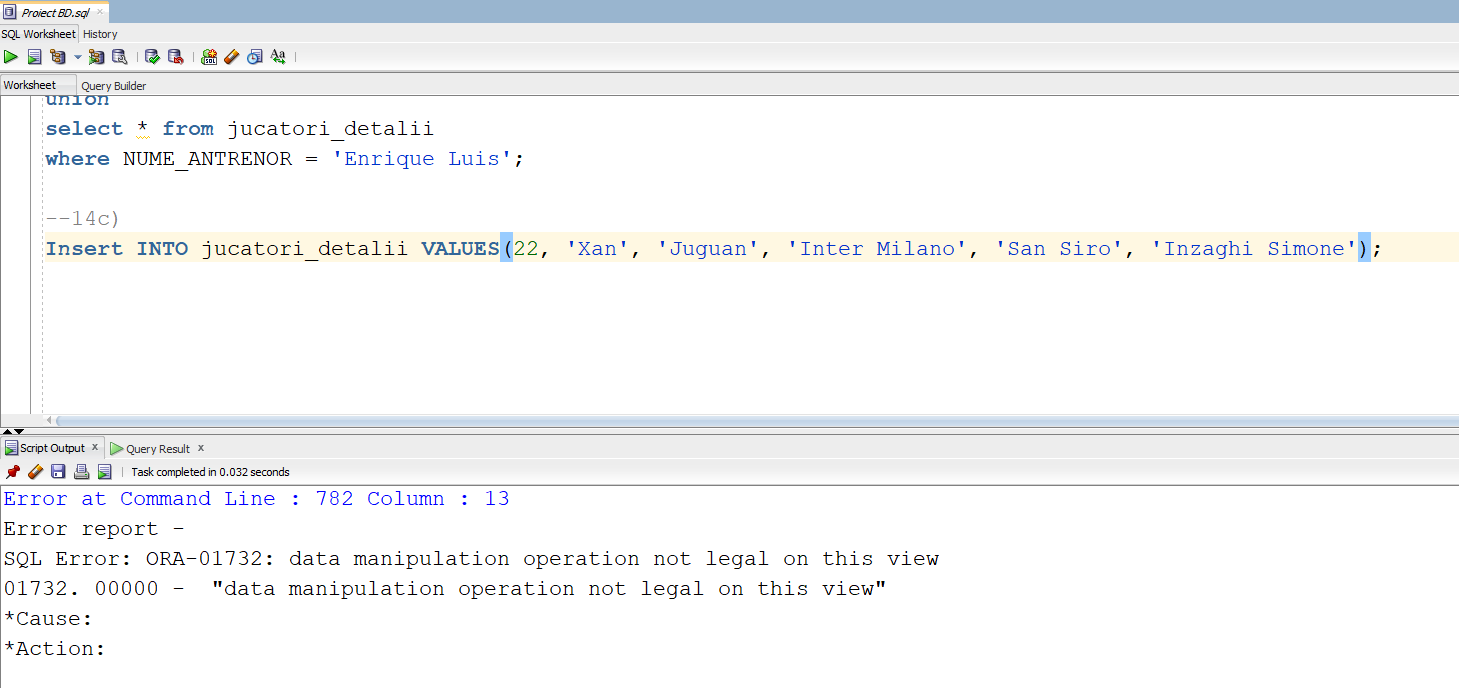


--exercițiul 14.3:

Insert INTO jucatori\_detalii VALUES(22, 'Xan', 'Juguan', 'Inter Milano', 'San Siro', 'Inzaghi Simone');

--Eroare ORA-01732

Insert-ul nu este permis în vizualizare, deoarece aceasta doar preia atribute din entitățile bazei de date, neputând însă să le și modifice în înteriorul ei.



**15. Formulați în limbaj natural și implementați în SQL 3 cereri complexe**

--a) afisati jucatorii echipelor care se afla in campionatele ”Premier League” sau ”Bundesliga”

--si care joaca pe stadioane cu o capacitate mai mare de 65000 de oameni si au manageri

--cu salarii saptamanale de peste 9500

Select

j.nume\_de\_familie || ' ' || j.prenume AS "NUME COMPLET"

from

Jucatori j

*full outer join* Echipe e ON j.id\_echipa = e.id\_echipa

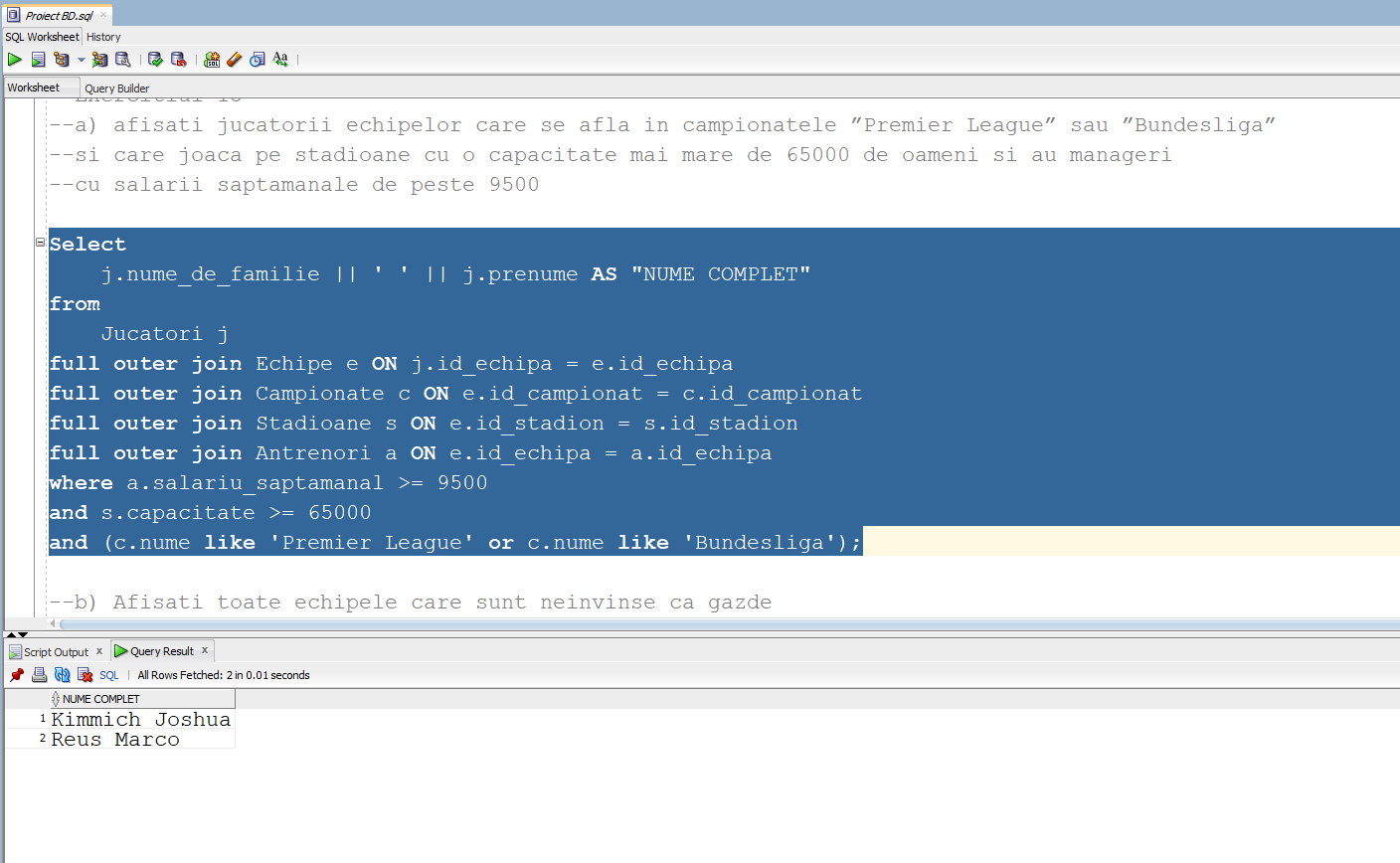
*full outer join* Campionate c ON e.id\_campionat = c.id\_campionat

*full outer join* Stadioane s ON e.id\_stadion = s.id\_stadion

*full outer join* Antrenori a ON e.id\_echipa = a.id\_echipa

where a.salariu\_saptamanal >= 9500

and s.capacitate >= 65000

and (c.nume like 'Premier League' or c.nume like 'Bundesliga'); 

--b) Afisati toate echipele care sunt neinvinse ca gazde

--(au castigat sau au facut cel putin egal in toate meciurile ca gazde)

SELECT e.nume\_echipa

FROM echipe e, partide p

where e.id\_echipa = p.id\_gazde

and *NOT EXISTS* (

SELECT 1

FROM partide p1

WHERE p1.id\_gazde = e.id\_echipa

AND *NOT EXISTS* (

SELECT 1

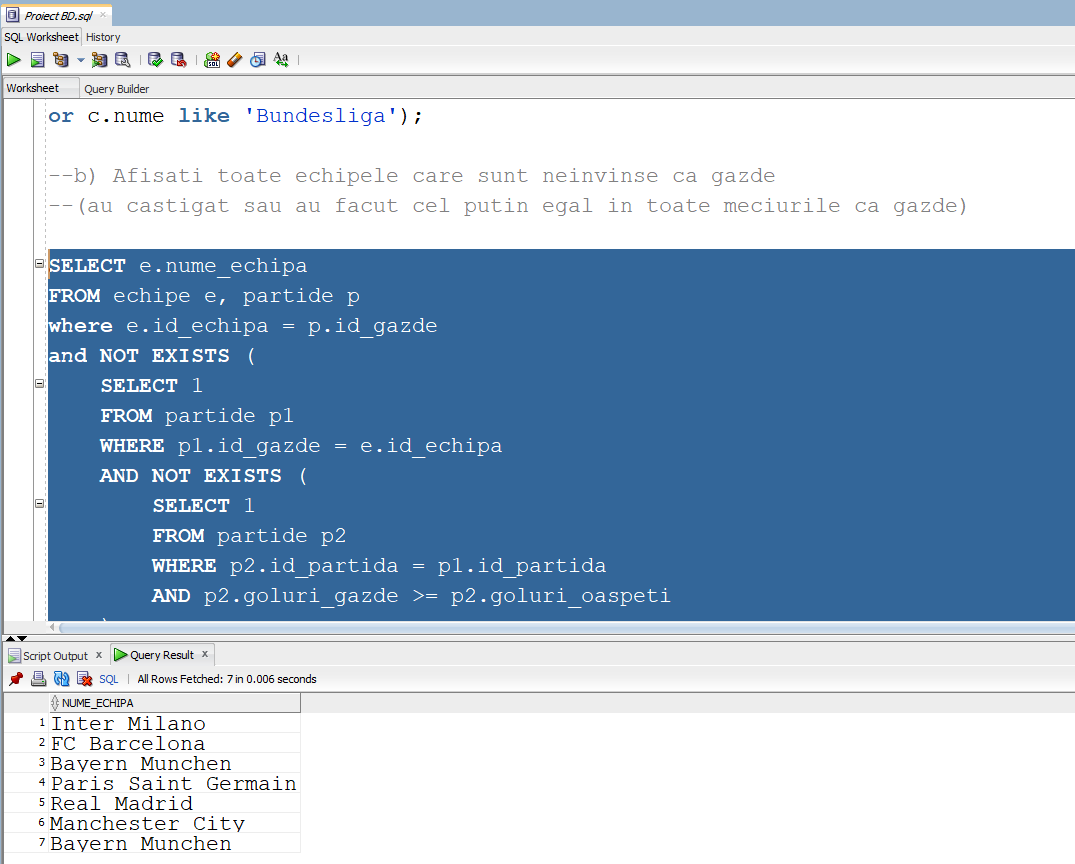
FROM partide p2

WHERE p2.id\_partida = p1.id\_partida

AND p2.goluri\_gazde >= p2.goluri\_oaspeti

)

);



--c) afisati echipele (primele 3) cu cei mai multi jucatori care au inscris peste 15 goluri

--de-a lungul carierei

SELECT \*

FROM (SELECT e.nume\_echipa, COUNT(j.id\_jucator) AS numar\_jucatori

FROM echipe e, jucatori j

where e.id\_echipa = j.id\_echipa

and j.goluri > 15

GROUP BY e.nume\_echipa

ORDER BY numar\_jucatori DESC) ordonati\_desc

WHERE ROWNUM <= 3;

