Postolache Andreea Miruna

Grupa 242

Gestionarea rezultatelor sportivilor dintr-un club la competiții

(această aplicație se face pentru un sport unic)

# 1.Descrierea

Modelul de date va furniza informații despre rezultatele obținute de sportivii unui club la competițiile la care au participat, cât și detalii ce țin de acel sportiv: cine îl antrenează, din ce club face parte și care sunt sponsorii săi. Modelul de date se aplică pentru sporturile individuale și este util pentru federația acelui sport, deoarece aceasta trebuie să aibă o astfel de evidență.

Există cluburi sportive care angajează antrenori pentru ca aceștia să antreneze sportivi.

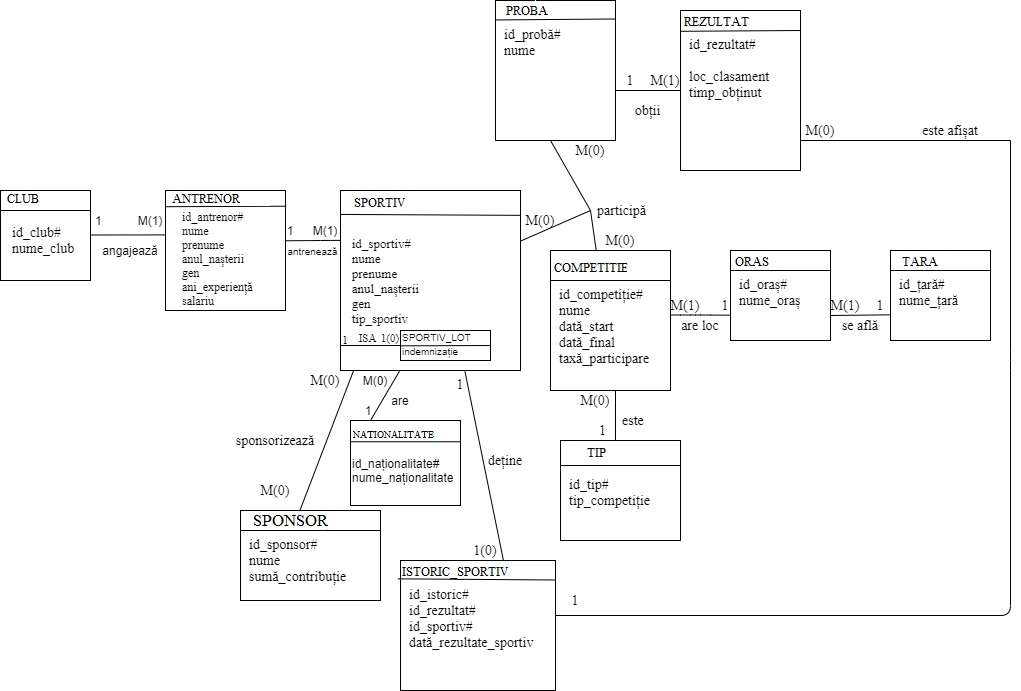
Sportivul poate fi selectat să facă parte din lot. Totodată, acesta poate să fie sponsorizat. Cel mai important este faptul că un sportiv poate să își aleagă la ce competiții vrea să participe, dar și probele din cadrul acelei competiții la care dorește să ia parte.

La fiecare probă din cadrul competiției la care a participat se va reține rezultatul obținut (locul obținut și timpul pe care l-a realizat în acea probă). Aceste rezultate diferă de la probă la probă și de la competiție la competiție. Mai precis, dacă la proba x de la competiția y sportivul a avut un rezultat, la aceeași probă, dar la altă competiție acesta va putea avea un alt rezultat.

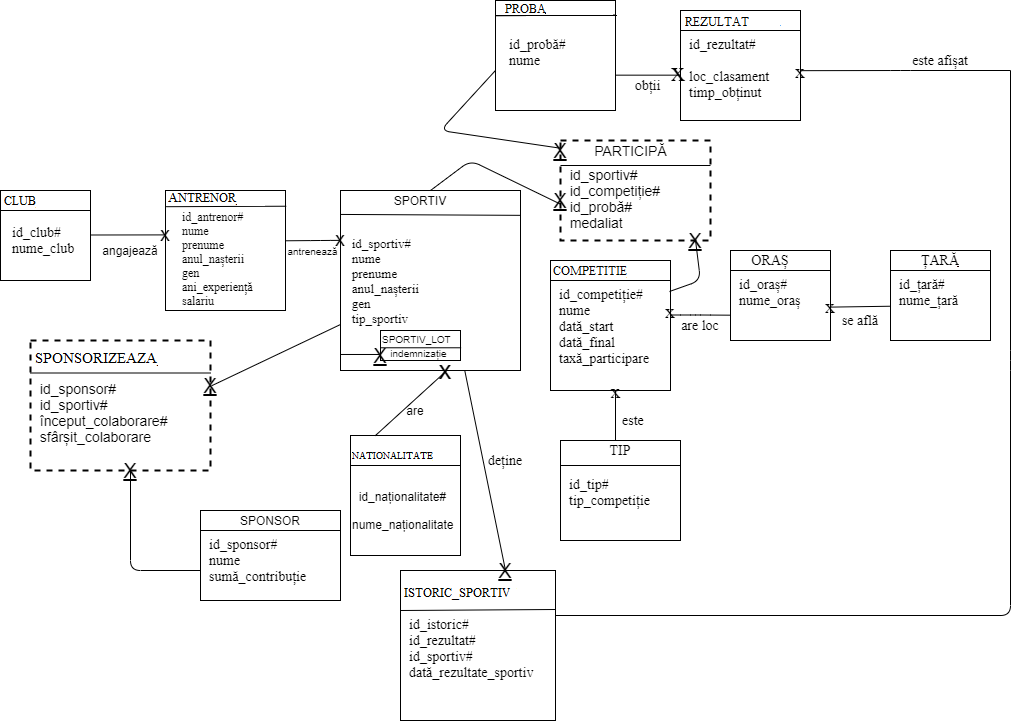
În istoricul sportivului se vor reține rezultatele aferente probelor din fiecare competiție la care a participat (adică să apară competitia cu probele la care a participat și cu rezultatul obținut).

Competițiile la care participă sportivii sunt de mai multe tipuri (naționale, regionale, europene, balcanice, etc.) și acestea au loc în orașele unor țări.

# 2. Diagrama entitate-relație



# 3. Diagrama conceptuală



# 4. Crearea tabelelor

CREATE TABLE CLUB

(id\_club NUMBER(5),

nume\_club VARCHAR2(50) CONSTRAINT nume\_club\_nn NOT NULL,

UNIQUE (nume\_club)

);

ALTER TABLE CLUB

ADD CONSTRAINT id\_club\_pk PRIMARY KEY (id\_club);

CREATE TABLE ANTRENOR

(

id\_antrenor NUMBER(5),

nume VARCHAR2(25) CONSTRAINT nume\_antrenor\_nn NOT NULL,

prenume VARCHAR2(25) CONSTRAINT prenume\_antrenor\_nn NOT NULL,

anul\_nasterii DATE CONSTRAINT nastere\_nn NOT NULL,

gen CHAR(1) CONSTRAINT gen\_nn NOT NULL,

ani\_experienta NUMBER(3),--poate fi null

salariu NUMBER(8,2) CONSTRAINT salariu\_min CHECK (salariu>300.00),

id\_club NUMBER(5),

CONSTRAINT genul CHECK (gen='M' OR gen='F')

);

ALTER TABLE ANTRENOR

ADD CONSTRAINT id\_antrenor\_pk PRIMARY KEY(id\_antrenor);

ALTER TABLE ANTRENOR

ADD CONSTRAINT id\_club\_fk FOREIGN KEY(id\_club) REFERENCES CLUB (id\_club);

ALTER TABLE ANTRENOR

MODIFY id\_club number(5) NOT NULL;

CREATE TABLE NATIONALITATE

(

id\_nationalitate NUMBER(5),

nume\_nationalitate VARCHAR2(15) CONSTRAINT nationalitate\_sportiv\_nn NOT NULL,

UNIQUE (nume\_nationalitate),

CONSTRAINT id\_nationalitate\_pk PRIMARY KEY (id\_nationalitate)

);

CREATE TABLE SPORTIV

(

id\_sportiv NUMBER(5),

gen CHAR(1) CONSTRAINT genul\_sp\_nn NOT NULL,

nume VARCHAR2(25) CONSTRAINT nume\_sportiv\_nn NOT NULL,

prenume VARCHAR2(25) CONSTRAINT prenume\_sportiv\_nn NOT NULL,

anul\_nasterii DATE CONSTRAINT nastere\_sp\_nn NOT NULL,

id\_nationalitate NUMBER(5)CONSTRAINT nationalitate\_sp\_nn NOT NULL,

CONSTRAINT genul\_sp CHECK (gen='M' OR gen='F'),

tip\_sportiv VARCHAR2(25) CONSTRAINT tip\_sportiv\_nn NOT NULL,

indemnizatie NUMBER(5,1),

id\_antrenor NUMBER(5),

CONSTRAINT id\_sportiv\_pk PRIMARY KEY (id\_sportiv),

CONSTRAINT id\_antrenor\_fk FOREIGN KEY(id\_antrenor) REFERENCES ANTRENOR(id\_antrenor),

CONSTRAINT id\_nationalitate\_fk FOREIGN KEY(id\_nationalitate) REFERENCES NATIONALITATE(id\_nationalitate)

);

ALTER TABLE SPORTIV

MODIFY id\_antrenor number(5) NOT NULL;

CREATE TABLE SPONSOR

(id\_sponsor NUMBER(5),

nume VARCHAR2(50) CONSTRAINT nume\_sponsor\_nn NOT NULL,

suma\_contributie NUMBER(5,1) CONSTRAINT suma\_sponsor\_nn NOT NULL,

UNIQUE (nume),

CONSTRAINT id\_sponsor\_pk PRIMARY KEY (id\_sponsor)

);

CREATE TABLE SPONSORIZEAZA

( id\_sponsor NUMBER(5),

id\_sportiv NUMBER(5),

inceput\_colaborare DATE CONSTRAINT inc\_colab\_nn NOT NULL,

sfarsit\_colaborare DATE CONSTRAINT sf\_colab\_nn NOT NULL,

CONSTRAINT datele CHECK (inceput\_colaborare<sfarsit\_colaborare),

CONSTRAINT pk\_id\_sponsor\_id\_sportiv\_colab PRIMARY KEY (id\_sponsor,id\_sportiv,inceput\_colaborare),

CONSTRAINT id\_sponsor\_fk FOREIGN KEY (id\_sponsor) REFERENCES SPONSOR (id\_sponsor),

CONSTRAINT id\_sportiv\_fk FOREIGN KEY (id\_sportiv) REFERENCES SPORTIV (id\_sportiv)

);

CREATE TABLE TIP

(id\_tip NUMBER(5),

tip\_competitie VARCHAR2(20) CONSTRAINT comp\_tip\_nn NOT NULL,

UNIQUE(tip\_competitie)

);

ALTER TABLE TIP

ADD CONSTRAINT id\_tip\_pk PRIMARY KEY (id\_tip);

CREATE TABLE TARA

(id\_tara CHAR(2),

nume\_tara VARCHAR2(25) CONSTRAINT nume\_tara\_nn NOT NULL,

UNIQUE(nume\_tara)

);

ALTER TABLE TARA

ADD CONSTRAINT id\_tara\_pk PRIMARY KEY (id\_tara);

CREATE TABLE ORAS

(id\_oras NUMBER(5),

nume\_oras VARCHAR2(25) CONSTRAINT nume\_oras\_nn NOT NULL,

id\_tara CHAR(2),

CONSTRAINT id\_oras\_pk PRIMARY KEY (id\_oras),

CONSTRAINT id\_tara\_fk FOREIGN KEY (id\_tara) REFERENCES TARA (id\_tara),

UNIQUE (nume\_oras)

);

ALTER TABLE oras

MODIFY id\_tara char(2) NOT NULL;

CREATE TABLE COMPETITIE

(id\_competitie NUMBER(5),

nume VARCHAR2(30) CONSTRAINT nume\_comp\_nn NOT NULL,

data\_start DATE CONSTRAINT data\_start\_nn NOT NULL,

data\_final DATE CONSTRAINT data\_final\_nn NOT NULL,

taxa\_participare NUMBER(4,1) CONSTRAINT taxa\_nn NOT NULL,

id\_oras NUMBER(5),

id\_tip NUMBER(5),

CONSTRAINT id\_competitie\_pk PRIMARY KEY (id\_competitie),

CONSTRAINT id\_tip\_fk FOREIGN KEY (id\_tip) REFERENCES TIP (id\_tip),

CONSTRAINT id\_oras\_fk FOREIGN KEY (id\_oras) REFERENCES ORAS (id\_oras),

CONSTRAINT date\_concurs CHECK (data\_start<data\_final),

UNIQUE (nume)

);

ALTER TABLE competitie

MODIFY id\_oras number(5) NOT NULL;

ALTER TABLE competitie

MODIFY id\_tip number(5) NOT NULL;

CREATE TABLE ISTORIC\_SPORTIV

(

data\_inscriere\_club DATE CONSTRAINT data\_inscriere\_club\_nn NOT NULL,

data\_renuntare\_la\_sport DATE,

data\_rezultate\_sportiv DATE CONSTRAINT data\_sportiv\_rez NOT NULL,

id\_istoric NUMBER(5),

id\_rezultat NUMBER(5),

id\_sportiv NUMBER(5),

CONSTRAINT pk\_istoric\_rezultat\_sportiv primary key(id\_istoric, id\_rezultat,id\_sportiv),

CONSTRAINT id\_sportivul\_fk FOREIGN KEY (id\_sportiv) REFERENCES SPORTIV (id\_sportiv)

);

--după ce am realizat inserările am observat că nu mai este nevoie de două

--coloane (data\_inscriere\_club si data\_renuntare\_la\_sport) din tabelul

--TABLE ISTORIC\_SPORTIV. Acestea au fost eliminate

--după inserările de la exercitiul 5 cu drop.

--(codul se găseste înainte de comenzile select pentru toate tabelele)

CREATE TABLE PROBA

(

id\_proba NUMBER(5),

nume VARCHAR2(30) CONSTRAINT nume\_proba\_nn NOT NULL,

CONSTRAINT id\_proba\_pk PRIMARY KEY (id\_proba),

UNIQUE (nume)

);

CREATE TABLE REZULTAT

(

id\_rezultat NUMBER(5),

id\_proba NUMBER(5),

loc\_clasament NUMBER(4) CONSTRAINT loc\_cls\_nn NOT NULL,

timp\_obtinut NUMBER(5) CONSTRAINT timp\_nn NOT NULL,--IN SECUNDE

id\_sportiv NUMBER(5),

id\_istoric NUMBER(5),

CONSTRAINT pk\_rezultat PRIMARY KEY(id\_rezultat),

CONSTRAINT fk\_istoric\_rezultat\_sportiv FOREIGN KEY(id\_istoric, id\_rezultat,id\_sportiv) REFERENCES ISTORIC\_SPORTIV (id\_istoric, id\_rezultat,id\_sportiv),

CONSTRAINT fk\_proba FOREIGN KEY(id\_proba) REFERENCES PROBA(id\_proba)

);

CREATE TABLE PARTICIPA

(

id\_sportiv NUMBER(5),

id\_competitie NUMBER(5),

id\_proba NUMBER(5),

medaliat NUMBER(1) CONSTRAINT medalie\_nn NOT NULL,

CONSTRAINT medaliat\_da\_sau\_nu CHECK (medaliat IN (1,0)),

CONSTRAINT pk\_sportiv\_rezultat\_sportiv PRIMARY KEY (id\_sportiv,id\_competitie,id\_proba),

CONSTRAINT id\_sportivi\_fk FOREIGN KEY (id\_sportiv) REFERENCES SPORTIV (id\_sportiv),

CONSTRAINT id\_compet\_fk FOREIGN KEY (id\_competitie) REFERENCES COMPETITIE(id\_competitie),

CONSTRAINT id\_probele\_fk FOREIGN KEY (id\_proba) REFERENCES PROBA (id\_proba)

);

# 5. Inserări

SELECT \* FROM CLUB;

CREATE SEQUENCE SEQ\_CLUB\_MPO

INCREMENT BY 10

START WITH 70

MAXVALUE 10000

NOCYCLE;

INSERT INTO CLUB

VALUES(SEQ\_CLUB\_MPO.NEXTVAL, 'AQUA TEAM');

INSERT INTO CLUB

VALUES(SEQ\_CLUB\_MPO.NEXTVAL, 'DELFINUL BUCURESTI');

INSERT INTO CLUB

VALUES(SEQ\_CLUB\_MPO.NEXTVAL, 'BUCHAREST SPORT CLUB ELITE');

INSERT INTO CLUB

VALUES(SEQ\_CLUB\_MPO.NEXTVAL, 'STEAUA CSA');

INSERT INTO CLUB

VALUES(SEQ\_CLUB\_MPO.NEXTVAL, 'ATACK TEAM');

COMMIT;

rollback;

SELECT \*FROM CLUB;

CREATE SEQUENCE SEQ\_ANTRENOR\_MPO

INCREMENT BY 10

START WITH 20

MAXVALUE 10000

NOCYCLE;

SELECT \*FROM antrenor;

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'BECHERU','CATALIN',TO\_DATE('10-09-1981','dd-mm-yyyy'),'M',10,500,70);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'BECHERU','IULIA',TO\_DATE('20-12-1984','dd-mm-yyyy'),'F',8,400.54,70);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'ORMENISAN','DANIEL',TO\_DATE('09-06-1988','dd-mm-yyyy'),'M',7,300.3,70);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'DRAGULET','DAN',TO\_DATE('20-09-1966','dd-mm-yyyy'),'M',20,600.45,80);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'GIURCANEANU','MATEI',TO\_DATE('10-09-1963','dd-mm-yyyy'),'M',23,800.5,80);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'GRIGORE','MARIUS',TO\_DATE('25-01-1981','dd-mm-yyyy'),'M',20,700.2,90);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'TOMA','ALEXANDRU',TO\_DATE('10-11-1991','dd-mm-yyyy'),'M',10,350,90);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'BORDEI','DARIA',TO\_DATE('03-09-1976','dd-mm-yyyy'),'F',25,800,100);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'BRAN','IONEL',TO\_DATE('10-08-1977','dd-mm-yyyy'),'M',21,650,100);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'FIRANESCU','SANDRA',TO\_DATE('24-03-1997','dd-mm-yyyy'),'F',4,300.1,110);

INSERT INTO antrenor(id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

VALUES (SEQ\_ANTRENOR\_MPO.nextval,'STAN','DENISA',TO\_DATE('11-07-1980','dd-mm-yyyy'),'F',6,310,110);

COMMIT;

ROLLBACK;

SELECT\*FROM nationalitate;

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (1,'ROMAN');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (2,'SPANIOL');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (3,'FRANCEZ');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (4,'SUEDEZ');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (5,'ARAB');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (6,'UNGUR');

INSERT INTO nationalitate(id\_nationalitate,nume\_nationalitate)

values (7,'ENGLEZ');

COMMIT;

select \*from sportiv;

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (1,'M','MARINCA','RARES',TO\_DATE('26-03-2001','dd-mm-yyyy'),1,'SPORTIV',NULL,20);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (2,'M','POPESCU','ROBERT',TO\_DATE('21-11-2002','dd-mm-yyyy'),1,'SPORTIV LOT',200,30);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (3,'F','STANCA','IULIA',TO\_DATE('28-03-2001','dd-mm-yyyy'),2,'SPORTIV',NULL,40);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (4,'F','BELLU','SARA',TO\_DATE('16-08-2000','dd-mm-yyyy'),3,'SPORTIV LOT',300,50);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (5,'M','GROFF','PHILIPPE',TO\_DATE('10-03-1999','dd-mm-yyyy'),4,'SPORTIV LOT',500,60);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (6,'M','AHMAD','ZAHED',TO\_DATE('22-09-2007','dd-mm-yyyy'),5,'SPORTIV',NULL,70);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (7,'F','BLEJAN','ALEXANDRA',TO\_DATE('12-05-2006','dd-mm-yyyy'),1,'SPORTIV',NULL,80);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (8,'F','BUZEA','ANA',TO\_DATE('21-07-2001','dd-mm-yyyy'),1,'SPORTIV LOT',600.5,90);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (9,'M','COSTROJ','ANTONIO',TO\_DATE('19-07-2002','dd-mm-yyyy'),1,'SPORTIV',NULL,100);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (10,'M','MANAFI','ZOLDA',TO\_DATE('20-02-1998','dd-mm-yyyy'),6,'SPORTIV',NULL,110);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (11,'F','PANA','BIANCA',TO\_DATE('21-02-1990','dd-mm-yyyy'),1,'SPORTIV LOT',600.5,120);

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (12,'M','SIMION','ANDREI',TO\_DATE('20-10-1998','dd-mm-yyyy'),1,'SPORTIV LOT',800,30);

COMMIT;

INSERT INTO SPONSOR(id\_sponsor,nume,suma\_contributie)

values (1,'RED BULL',500.5);

INSERT INTO SPONSOR(id\_sponsor,nume,suma\_contributie)

values (2,'ISOSTAR',600.4);

INSERT INTO SPONSOR(id\_sponsor,nume,suma\_contributie)

values (3,'INFINITY TROPHY',550.1);

INSERT INTO SPONSOR(id\_sponsor,nume,suma\_contributie)

values (4,'CARREFOUR',800.7);

INSERT INTO SPONSOR(id\_sponsor,nume,suma\_contributie)

values (5,'ENEL',1000.9);

SELECT \*FROM SPONSOR;

COMMIT;

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(1,1,to\_date('13-05-2019','dd-mm-yyyy'),to\_date('13-06-2023','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(1,2,to\_date('25-09-2020','dd-mm-yyyy'),to\_date('25-10-2024','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(2,1,to\_date('12-05-2018','dd-mm-yyyy'),to\_date('16-06-2021','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(2,11,to\_date('30-04-2010','dd-mm-yyyy'),to\_date('31-05-2020','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(3,4,to\_date('29-03-2019','dd-mm-yyyy'),to\_date('30-03-2024','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(3,5,to\_date('26-08-2020','dd-mm-yyyy'),to\_date('27-08-2022','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(4,11,to\_date('30-03-2010','dd-mm-yyyy'),to\_date('30-04-2021','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(4,2,to\_date('13-05-2021','dd-mm-yyyy'),to\_date('13-09-2022','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(4,6,to\_date('21-08-2017','dd-mm-yyyy'),to\_date('21-09-2018','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(5,11,to\_date('21-04-2018','dd-mm-yyyy'),to\_date('21-05-2022','dd-mm-yyyy'));

INSERT INTO SPONSORIZEAZA(id\_sponsor,id\_sportiv,inceput\_colaborare,sfarsit\_colaborare)

values(4,6,to\_date('25-09-2018','dd-mm-yyyy'),to\_date('27-09-2019','dd-mm-yyyy'));

COMMIT;

SELECT \*FROM sponsorizeaza;

select \*from sportiv;

select\* from tip;

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(1,'LOCAL');

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(2,'MUNICIPAL');

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(3,'JUDETEAN');

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(4,'EUROPEAN');

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(5,'BALCANIC');

INSERT INTO TIP(id\_tip, tip\_competitie)

VALUES(6,'MONDIAL');

COMMIT;

SELECT\* FROM TARA;

INSERT INTO TARA(id\_tara,nume\_tara)

values('RO','ROMANIA');

INSERT INTO TARA(id\_tara,nume\_tara)

values('PO','POLONIA');

INSERT INTO TARA(id\_tara,nume\_tara)

values('FR','FRANTA');

INSERT INTO TARA(id\_tara,nume\_tara)

values('JP','JAPONIA');

INSERT INTO TARA(id\_tara,nume\_tara)

values('IT','ITALIA');

INSERT INTO TARA(id\_tara,nume\_tara)

values('KO','KOREA');

COMMIT;

SELECT \*FROM ORAS;

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (1,'BUCURESTI','RO');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (2,'CRAIOVA','RO');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (3,'LUBLIN','PO');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (4,'PARIS','FR');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (5,'KYOTO','JP');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (6,'NAPOLI','IT');

INSERT INTO ORAS(id\_oras,nume\_oras,id\_tara)

VALUES (7,'GWANGJU','KO');

COMMIT;

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(1,'CUPA DE TOAMNA 2016',TO\_DATE('05-11-2016','DD-MM-YYYY'),TO\_DATE('06-11-2016','DD-MM-YYYY'),100,1,1);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(2,'CUPA JAKED 2016',TO\_DATE('21-05-2016','DD-MM-YYYY'),TO\_DATE('22-05-2016','DD-MM-YYYY'),200,2,2);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(3,'CUPA ANDREI NICOLESCU EDITIA 1',TO\_DATE('15-10-2017','DD-MM-YYYY'),TO\_DATE('16-10-2017','DD-MM-YYYY'),400.5,6,4);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(4,'CUPA DE IARNA 2018',TO\_DATE('22-11-2018','DD-MM-YYYY'),TO\_DATE('03-12-2018','DD-MM-YYYY'),300,2,3);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(5,'WORLD AQUATICS CHAMPIONSHIP 1',TO\_DATE('12-07-2019','DD-MM-YYYY'),TO\_DATE('28-07-2019','DD-MM-YYYY'),900.5,7,6);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(6,'EUROPEAN GAMES 2021',TO\_DATE('23-06-2021','DD-MM-YYYY'),TO\_DATE('27-06-2021','DD-MM-YYYY'),700,5,4);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(7,'JUNIOR CHAMPIONSHIP 2021',TO\_DATE('31-07-2021','DD-MM-YYYY'),TO\_DATE('10-08-2021','DD-MM-YYYY'),890,3,4);

INSERT INTO COMPETITIE(id\_competitie,nume,data\_start,data\_final,taxa\_participare,id\_oras,id\_tip)

values(8,'FINA SWIMMING WORLD CUP 2021',TO\_DATE('20-08-2021','DD-MM-YYYY'),TO\_DATE('24-08-2021','DD-MM-YYYY'),500,4,6);

COMMIT;

SELECT\*FROM COMPETITIE;

SELECT\*

FROM PROBA;

INSERT INTO PROBA(id\_proba,nume)

values(1,'50 LIBER');

INSERT INTO PROBA(id\_proba,nume)

values(2,'50 SPATE');

INSERT INTO PROBA(id\_proba,nume)

values(3,'100 LIBER');

INSERT INTO PROBA(id\_proba,nume)

values(4,'200 BRAS');

INSERT INTO PROBA(id\_proba,nume)

values(5,'200 FLUTURE');

INSERT INTO PROBA(id\_proba,nume)

values(6,'400 LIBER');

COMMIT;

SELECT \*

FROM PARTICIPA;

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(1,1,1,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(1,4,3,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(2,4,5,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(2,1,1,1);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(5,5,6,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(6,1,2,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(7,1,1,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(8,5,4,1);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(11,2,2,0);

INSERT INTO PARTICIPA(id\_sportiv,id\_competitie,id\_proba,medaliat)

values(11,3,5,1);

commit;

select \*FROM ISTORIC\_SPORTIV;

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('01-01-2006','DD-MM-YYYY'),NULL,TO\_DATE('05-11-2016','DD-MM-YYYY'),1,1,1);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('01-01-2006','DD-MM-YYYY'),NULL,TO\_DATE('25-11-2018','DD-MM-YYYY'),1,2,1);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('21-11-2007','DD-MM-YYYY'),NULL,TO\_DATE('26-11-2018','DD-MM-YYYY'),2,3,2);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('21-11-2007','DD-MM-YYYY'),NULL,TO\_DATE('05-11-2016','DD-MM-YYYY'),2,4,2);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('22-03-2008','DD-MM-YYYY'),NULL,TO\_DATE('14-07-2019','DD-MM-YYYY'),5,5,5);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('21-11-2011','DD-MM-YYYY'),TO\_DATE('14-08-2020','DD-MM-YYYY'),TO\_DATE('06-11-2016','DD-MM-YYYY'),6,7,6);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('14-05-2012','DD-MM-YYYY'),NULL,TO\_DATE('05-11-2016','DD-MM-YYYY'),7,10,7);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('19-08-2007','DD-MM-YYYY'),NULL,TO\_DATE('14-07-2019','DD-MM-YYYY'),8,8,8);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('22-11-1997','DD-MM-YYYY'),NULL,TO\_DATE('21-05-2016','DD-MM-YYYY'),11,6,11);

INSERT INTO ISTORIC\_SPORTIV(data\_inscriere\_club,data\_renuntare\_la\_sport,data\_rezultate\_sportiv,id\_istoric,id\_rezultat,id\_sportiv)

values(TO\_DATE('22-11-1997','DD-MM-YYYY'),NULL,TO\_DATE('15-10-2017','DD-MM-YYYY'),11,9,11);

COMMIT;

select\*from rezultat;

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(1,1,4,30,1,1);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(2,3,10,70,1,1);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(3,5,9,180,2,2);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(4,1,2,26,2,2);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(5,6,20,420,5,5);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(6,2,8,43,11,11);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(7,2,60,70,6,6);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(8,4,3,170,8,8);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(9,5,3,175,11,11);

INSERT INTO REZULTAT(id\_rezultat,id\_proba,loc\_clasament,timp\_obtinut,id\_sportiv,id\_istoric)

VALUES(10,1,100,62,7,7);

COMMIT;

ALTER TABLE ISTORIC\_SPORTIV

DROP COLUMN data\_inscriere\_club;

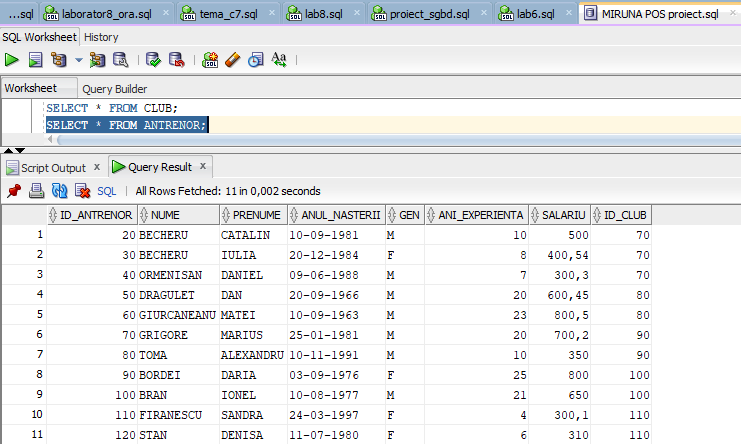
ALTER TABLE ISTORIC\_SPORTIV

DROP COLUMN data\_renuntare\_la\_sport;

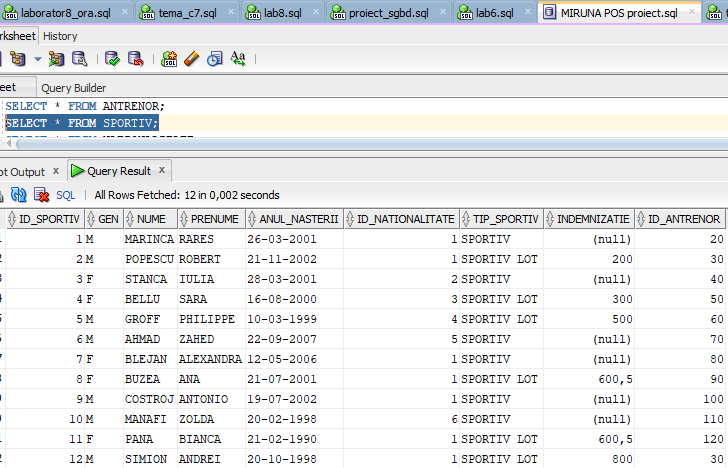
SELECT \* FROM CLUB;



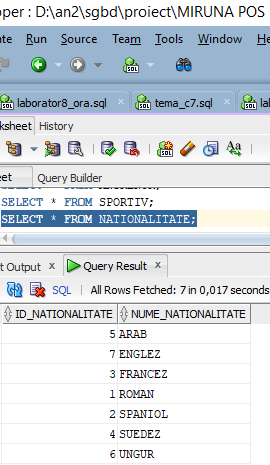
SELECT \* FROM ANTRENOR;



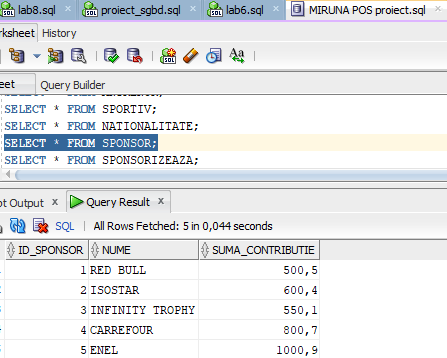
SELECT \* FROM SPORTIV;



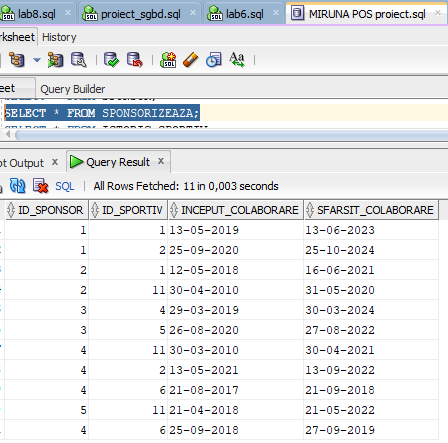
SELECT \* FROM NATIONALITATE;



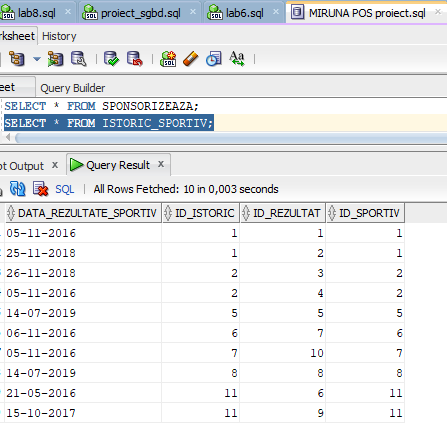
SELECT \* FROM SPONSOR;



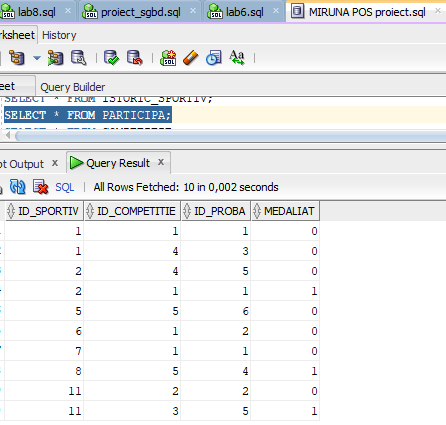
SELECT \* FROM SPONSORIZEAZA;



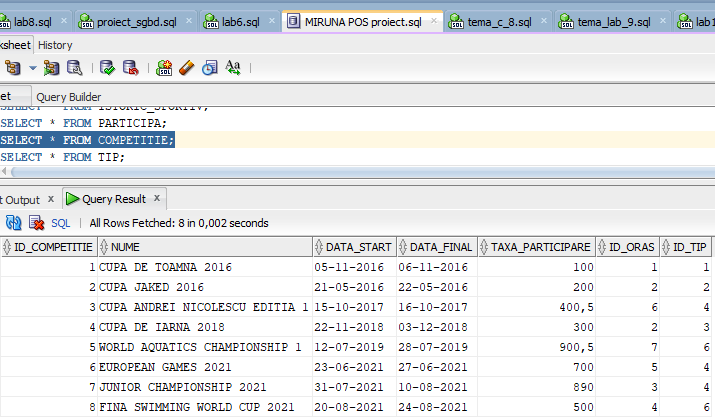
SELECT \* FROM ISTORIC\_SPORTIV;



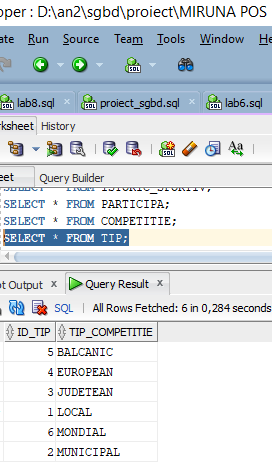
SELECT \* FROM PARTICIPA;



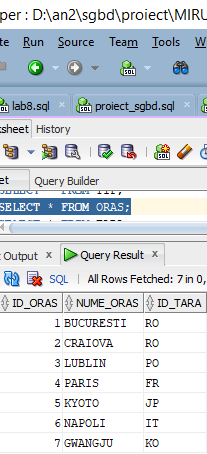
SELECT \* FROM COMPETITIE;



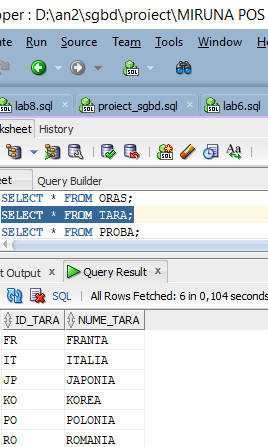
SELECT \* FROM TIP;



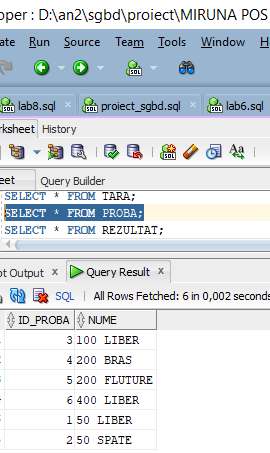
SELECT \* FROM ORAS;



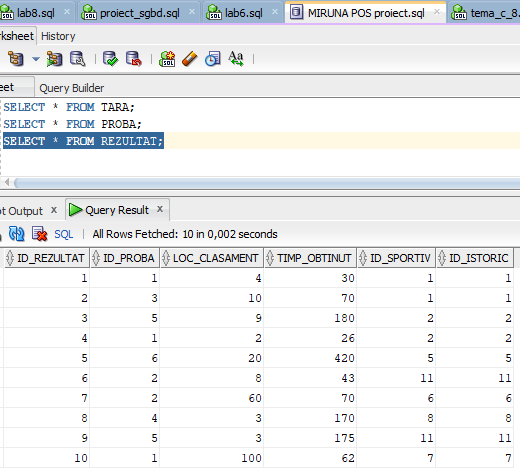
SELECT \* FROM TARA;



SELECT \* FROM PROBA;



SELECT \* FROM REZULTAT;



# 6.

/\*

**CERINTA**: Sa se afiseze pentru fiecare sportiv (care nu face parte din lot;

nu se iau in considerare sportivii care nu au participat la nicio competitie)

din clubul AQUA TEAM si Bucharest Sport Club Elite

numele competitiilor la care a participat si care au avut loc cel mult acum 6 ani.

Sa se mentina in colectii separate pentru fiecare club: numele competitiei,numele,prenumele sportivilor

(avand in vedere indicatiile de mai sus) si taxa de participare la acea competitie.

Mai mult, se doreste sa se afle cat a cheltuit fiecare club in parte pentru participarea

sportivilor (care nu fac parte din lot, plus mentiunile de mai sus) la aceste competitii.

\*/

----------------------------------

--REZOLVAREA

/\*

Pasii rezolvarii:

In tipul de date record se vor pune informatiile cerute.

Se vor construi un **tablou indexat**, respectiv **unul imbricat** pentru a tine informatiile separat pentru cele doua cluburi

Se face join intre 5 tabele pentru a lua informatiile necesare

Datele vor fi pastrate in tablourile precizate mai sus

Se parcurg tablourile si se afiseaza mesajele cerute si totdata se calculeaza

cat a cheltuit fiecare club in parte pentru participarea sportivilor care nu sunt in lot) la competitiile respective.

\*/

/

--------

create or replace procedure competitie\_cheltuiala\_2

is

TYPE tip\_record IS RECORD

(nume\_comp COMPETITIE.NUME%TYPE,

nume\_sp SPORTIV.NUME%TYPE,

prenume\_sp SPORTIV.PRENUME%TYPE,

taxa COMPETITIE.TAXA\_PARTICIPARE%TYPE,

id\_sp SPORTIV.ID\_SPORTIV%TYPE);

type tablou\_indexat is table of tip\_record index by pls\_integer;

type tablou\_imbricat is table of tip\_record;

t\_indexat\_aq tablou\_indexat;

t\_imbricat\_bsc tablou\_imbricat:= tablou\_imbricat();

suma\_bsc number:=0;

suma\_aq number:=0;

begin

SELECT distinct(COMPETITIE.NUME),SPORTIV.NUME,SPORTIV.PRENUME,COMPETITIE.TAXA\_PARTICIPARE,SPORTIV.ID\_SPORTIV

bulk collect INTO t\_indexat\_aq

FROM SPORTIV

JOIN ANTRENOR ON SPORTIV.ID\_ANTRENOR = ANTRENOR.ID\_ANTRENOR

JOIN CLUB ON ANTRENOR.ID\_CLUB = CLUB.ID\_CLUB

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

WHERE NUME\_CLUB = 'AQUA TEAM'

AND TIP\_SPORTIV <> 'SPORTIV LOT'

AND MONTHS\_BETWEEN(sysdate,COMPETITIE.DATA\_FINAL)<=72

order by sportiv.id\_sportiv;--6 ANI

SELECT distinct(COMPETITIE.NUME),SPORTIV.NUME,SPORTIV.PRENUME,COMPETITIE.TAXA\_PARTICIPARE,SPORTIV.ID\_SPORTIV

bulk collect INTO t\_imbricat\_bsc

FROM SPORTIV

JOIN ANTRENOR ON SPORTIV.ID\_ANTRENOR = ANTRENOR.ID\_ANTRENOR

JOIN CLUB ON ANTRENOR.ID\_CLUB = CLUB.ID\_CLUB

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

WHERE NUME\_CLUB ='BUCHAREST SPORT CLUB ELITE'

AND TIP\_SPORTIV <> 'SPORTIV LOT'

AND MONTHS\_BETWEEN(sysdate,COMPETITIE.DATA\_FINAL)<=72 --6 ANI

order by sportiv.id\_sportiv;

DBMS\_OUTPUT.PUT\_LINE('CLUBUL AQUA TEAM');

DBMS\_OUTPUT.NEW\_LINE();

if t\_indexat\_aq.count >0 then

for i in t\_indexat\_aq.FIRST..t\_indexat\_aq.LAST LOOP

if t\_indexat\_aq.exists(i) then

suma\_aq:=suma\_aq+t\_indexat\_aq(i).taxa;

if i = t\_indexat\_aq.FIRST then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_sp||' '||t\_indexat\_aq(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

end if;

if i>=t\_indexat\_aq.FIRST+1 and t\_indexat\_aq(i).id\_sp <> t\_indexat\_aq(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_sp||' '||t\_indexat\_aq(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

elsif i>=t\_indexat\_aq.FIRST+1 and t\_indexat\_aq(i).id\_sp = t\_indexat\_aq(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

end if;

end if;

END LOOP;

end if;

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('Clubul AQUA TEAM a cheltuit: '||suma\_aq||' lei');

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('CLUBUL BUCHAREST SPORT CLUB ELITE');

DBMS\_OUTPUT.NEW\_LINE();

if t\_imbricat\_bsc.count>0 then

for i in t\_imbricat\_bsc.FIRST..t\_imbricat\_bsc.LAST LOOP

if t\_imbricat\_bsc.exists(i) then

suma\_bsc:=suma\_bsc+t\_imbricat\_bsc(i).taxa;

if i = t\_imbricat\_bsc.FIRST then --primul rand din colectie

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_sp||' '||t\_imbricat\_bsc(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

end if;

if i>=t\_imbricat\_bsc.FIRST+1 and t\_imbricat\_bsc(i).id\_sp <> t\_imbricat\_bsc(i-1).id\_sp

--restul randurilor (daca am trecut la un alt sportiv)

then

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_sp||' '||t\_imbricat\_bsc(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

elsif i>=t\_imbricat\_bsc.FIRST+1 and t\_imbricat\_bsc(i).id\_sp = t\_imbricat\_bsc(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

end if;

end if;

END LOOP;

end if;

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('Clubul BUCHAREST SPORT CLUB ELITE a cheltuit: '||suma\_bsc||' lei');

end;

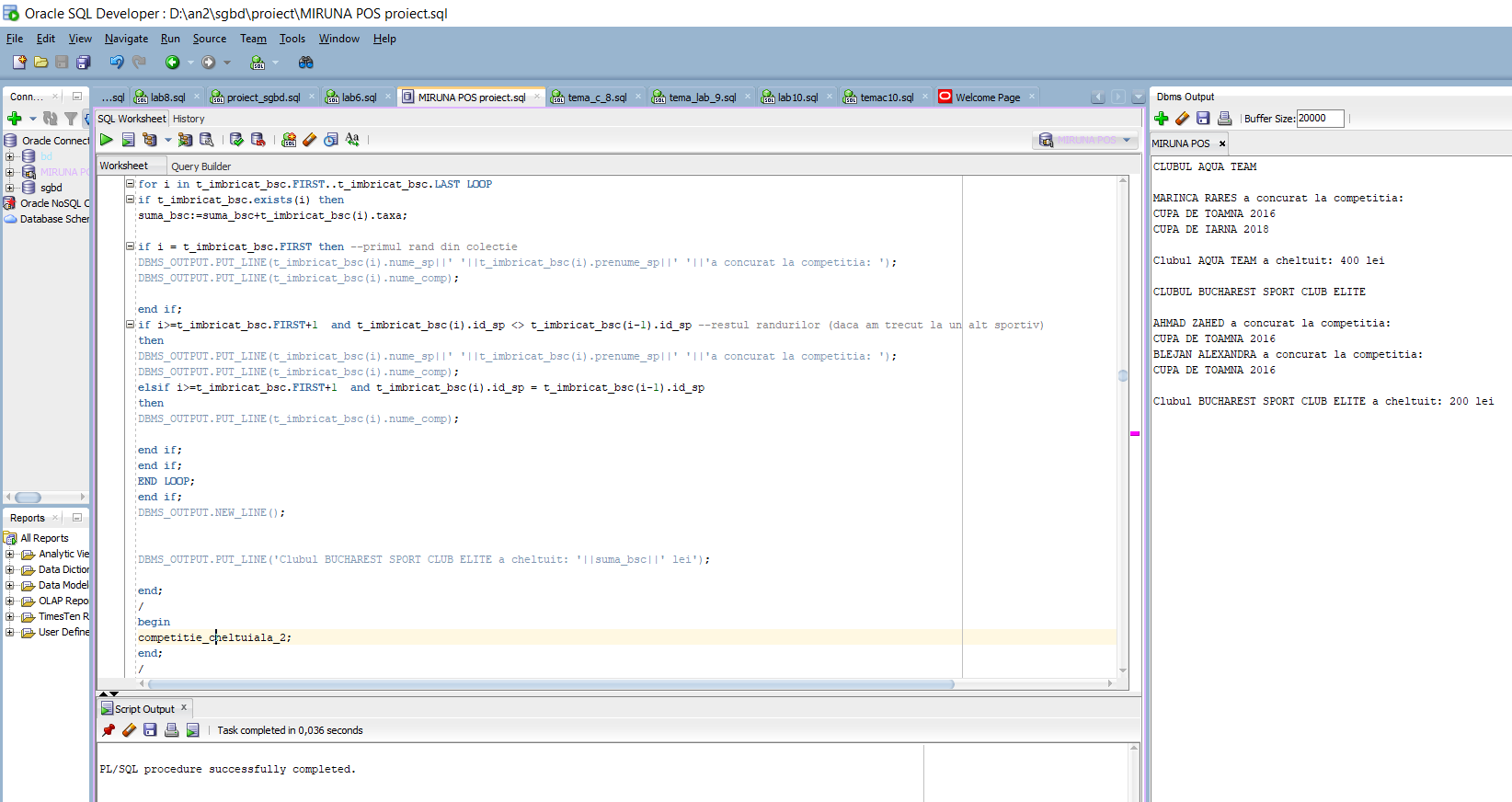
/

begin

competitie\_cheltuiala\_2;

end;

/



# 7.

/\*

**CERINTA:**

Pentru fiecare club care are cel putin un antrenor care la randul sau are cel putin un sportiv care este sponsorizat

in prezent si care a fost medaliat la cel putin o competitie

sa se afiseze numele clubului si sa se obtina lista antrenorilor (nume,prenume,salariu) care lucreaza in clubul respectiv.

Mai mult, sa se realizeze un numar de ordine pentru fiecare antrenor care se va reseta pentru fiecare club care respecta

cerintele.

Pentru fiecare club care indeplineste conditia sa se afiseze numarul de antrenori,

valoarea lunara a veniturilor antrenorilor, valoarea medie a veniturilor antrenorilor

A se tine cont de faptul ca un club trebuie sa aiba cel putin un antrenor.

\*/

----------------------

--REZOLVARE **(ciclu cursor cu subcerere)**

/\*

Pasii rezolvarii

Pentru a obtine cluburile cerute s-au facut doi pasi:

1. S-a vazut pentru ce antrenori (id\_antrenor) sportivii antrenati de acestia sunt sponsorizati si in data curenta

si au si cel putin o medalie. Astfel s-au facut join-uri intre 4 tabele.

2. S-a selectat numele clubului si id-ul pentru cluburile care au id\_urile antrenorilor descoperite la primul pas

pentru aceasta s-a facut un join intre 2 tabele (club\_mpo si antrenor\_mpo)

se parcurge cursorul in care se afla date despre cluburile care indeplinesc conditia

se afiseaza numele

si se parcurge cursorul in care se afla date despre toti antrenorii cluburilor care indeplinesc conditia

se afiseaza numele acestora, se calculeaza veniturile lunare si medii, dar si numarul de antrenori din clubul respectiv

si se afiseaza.

\*/

-------------------------------------

create or replace procedure club\_antr\_sp

is

i number;

venit\_lunar number;

BEGIN

for v\_club in (select club.nume\_club, club.id\_club

from club

join antrenor on club.id\_club = antrenor.id\_club

where antrenor.id\_antrenor in ( select antrenor.id\_antrenor

from antrenor

join sportiv on antrenor.id\_antrenor= sportiv.id\_antrenor

join sponsorizeaza on sportiv.id\_sportiv = sponsorizeaza.id\_sportiv

join participa on sportiv.id\_sportiv = participa.id\_sportiv

where sponsorizeaza.inceput\_colaborare <=sysdate and sponsorizeaza.sfarsit\_colaborare>=sysdate and medaliat=1

group by antrenor.id\_antrenor)

)

loop

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE('Clubul '||v\_club.nume\_club);

DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------------------------------------------');

i:=0;

venit\_lunar:=0;

for v\_antr in ( select nume,prenume,salariu

from antrenor

where id\_club = v\_club.id\_club

)

loop

i:=i+1;

venit\_lunar:=venit\_lunar+ v\_antr.salariu;

DBMS\_OUTPUT.PUT\_LINE(i||'. '||v\_antr.nume||' '||v\_antr.prenume);

end loop;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE('Numarul antrenorilor este de: '||i);

DBMS\_OUTPUT.PUT\_LINE('Valoarea lunara a veniturilor antrenorilor este de: '||venit\_lunar);

if i<>0 then

DBMS\_OUTPUT.PUT\_LINE('Valoarea medie a veniturilor antrenorilor este de: '||venit\_lunar/i);

end if;

end loop;

end;

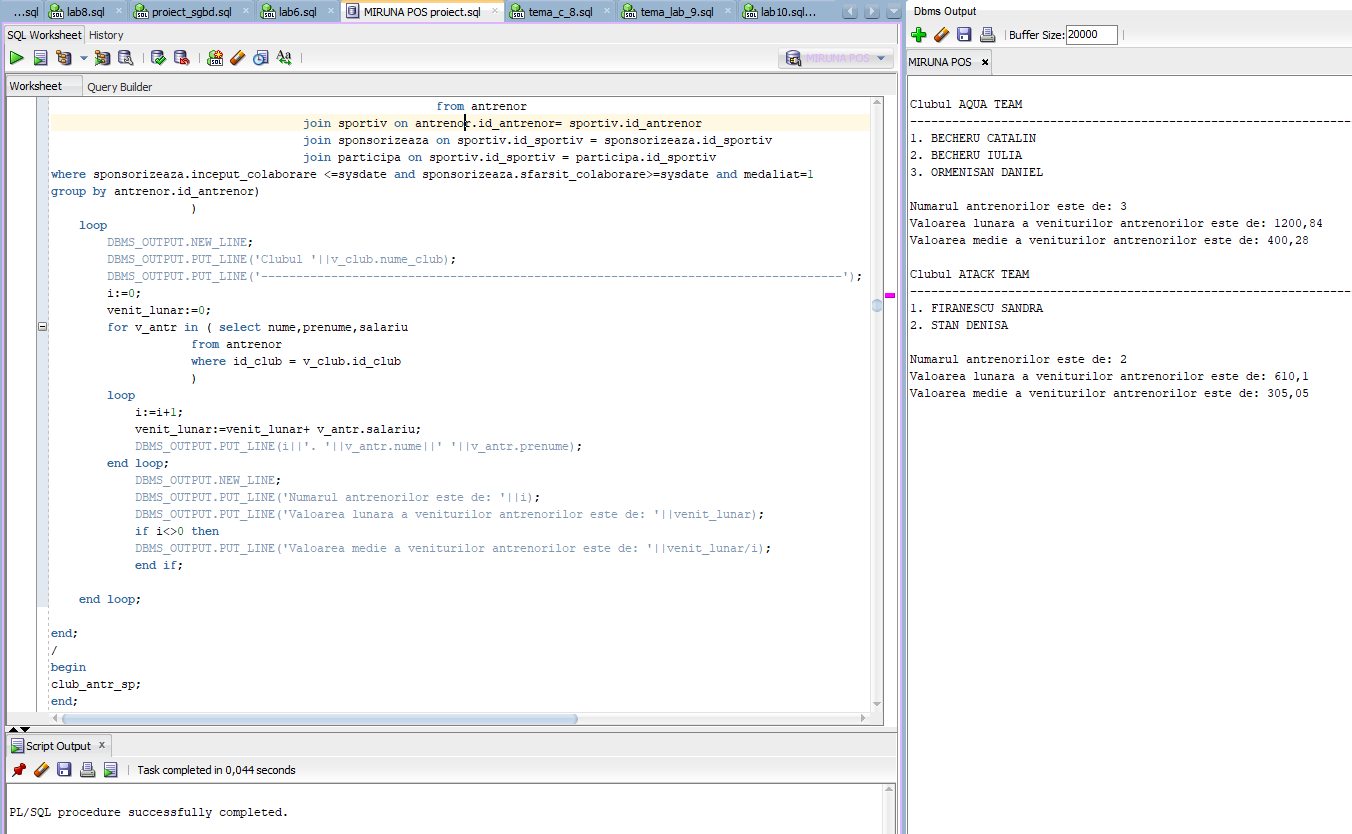
/

begin

club\_antr\_sp;

end;

/



# 8.

/\***CERINTA**:

Pentru o competitie al carei nume este dat ca parametru sa se afieseze numele ei si

sa se determine cat la suta din numarul de sportivi participanti

sunt cei care nu fac parte din lot. Tratati cazul in care competitia data ca parametru nu exista, respectiv cazul in care competitia exista, dar nu are niciun participant.

NOTA: Nu trebuie tratat cazul in care exista mai multe competitii cu acelasi nume pentru ca, in momentul crearii tabelului s-a pus constrangerea de UNIQUE(nume).

\*/

--------------------------------

--REZOLVAREA

/\*

Pasii rezolvarii

Se afiseaza numele competitiei

In cazul in care competita are un id printre cele din tabela COMPETITIE atunci, competitia exista,

altfel intoarce eroare NO\_DATA\_FOUND

Se numara cati sportivi au participat la competita cu numele dat

cu un join pe 2 tabele

Daca numarul aflat este diferit de 0 se va afla numarul sportivilor care nu fac parte din lot si participa la acea competite

cu un join pe 3 tabele

Se returneaza procentul dorit

Altfel, daca numarul este egal cu zero se va arunca o eroare

\*/

------------------------------

CREATE OR REPLACE FUNCTION NR\_SPORTIVI\_NU\_LOT\_COMP(V\_NUME COMPETITIE.NUME%TYPE)

RETURN NUMBER

IS

V\_ID\_COMP COMPETITIE.ID\_COMPETITIE%TYPE;

V\_ID\_C COMPETITIE.ID\_COMPETITIE%TYPE;

V\_NR\_TOTAL\_SP NUMBER;

V\_NR\_SP\_NU\_LOT NUMBER;

NU\_PARTICIPANTI EXCEPTION;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(V\_NUME);

SELECT ID\_COMPETITIE

INTO V\_ID\_COMP

FROM COMPETITIE

WHERE COMPETITIE.NUME = V\_NUME; --NO DATA FOUND CAND INTRODUC UN NUME CARE NU EXISTA

SELECT COUNT(competitie.id\_competitie)--INTOARCE 0 PENTRU COMPETITIILE CARE EXISTA --DAR CARE NU AU PARTICIPANTI

INTO V\_NR\_TOTAL\_SP

FROM COMPETITIE

FULL OUTER JOIN PARTICIPA ON COMPETITIE.ID\_COMPETITIE= PARTICIPA.ID\_COMPETITIE

WHERE ID\_SPORTIV IS NOT NULL AND competitie.id\_competitie= V\_ID\_COMP;

IF V\_NR\_TOTAL\_SP <>0 THEN

SELECT COUNT(COMPETITIE.ID\_COMPETITIE)

INTO V\_NR\_SP\_NU\_LOT

FROM COMPETITIE

FULL OUTER JOIN PARTICIPA ON COMPETITIE.ID\_COMPETITIE= PARTICIPA.ID\_COMPETITIE

JOIN SPORTIV ON PARTICIPA.ID\_SPORTIV= SPORTIV.ID\_SPORTIV

WHERE SPORTIV.TIP\_SPORTIV <> 'SPORTIV LOT' AND PARTICIPA.ID\_SPORTIV IS NOT NULL

AND competitie.id\_competitie= V\_ID\_COMP;

RETURN (V\_NR\_SP\_NU\_LOT\*100)/v\_nr\_total\_sp;

ELSE

RAISE NU\_PARTICIPANTI;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000,'Nu exista nicio competite cu numele dat.');

WHEN NU\_PARTICIPANTI THEN

RAISE\_APPLICATION\_ERROR(-20001,'Aceasta competite nu are participanti');

WHEN OTHERS THEN

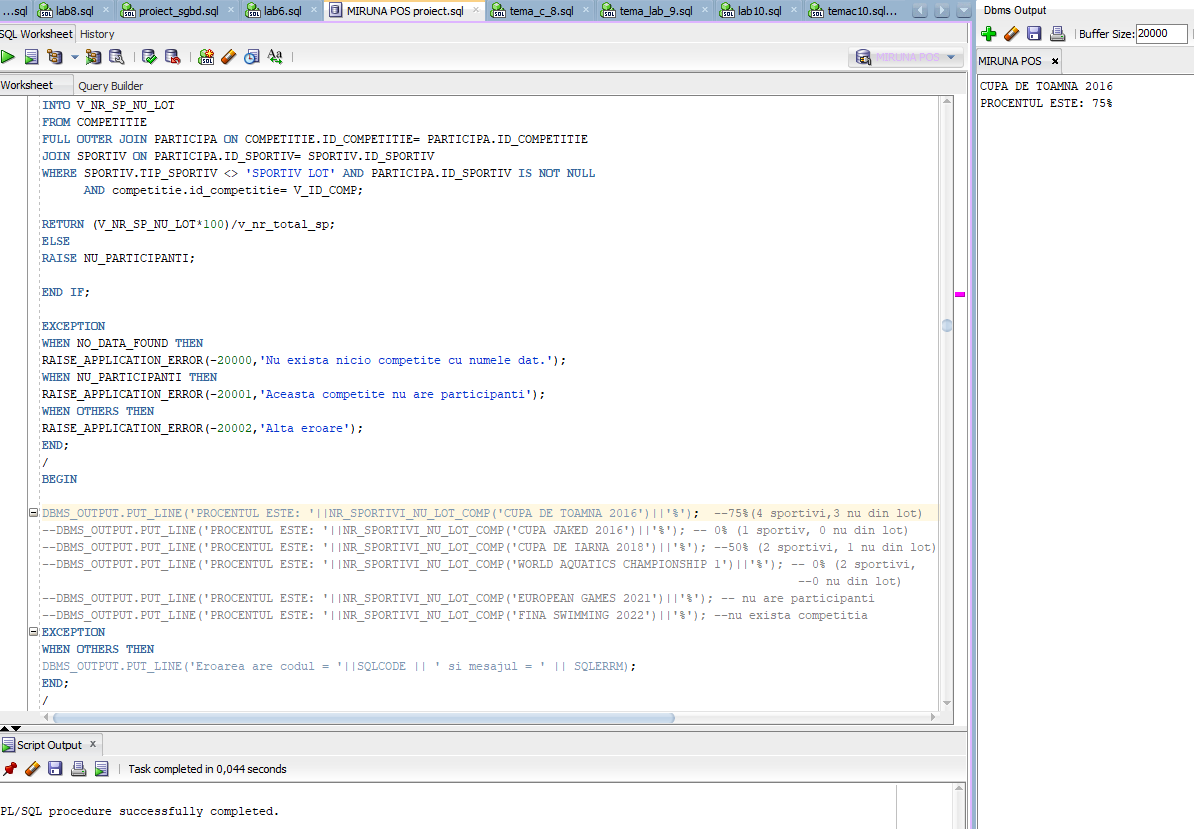
RAISE\_APPLICATION\_ERROR(-20002,'Alta eroare');

END;

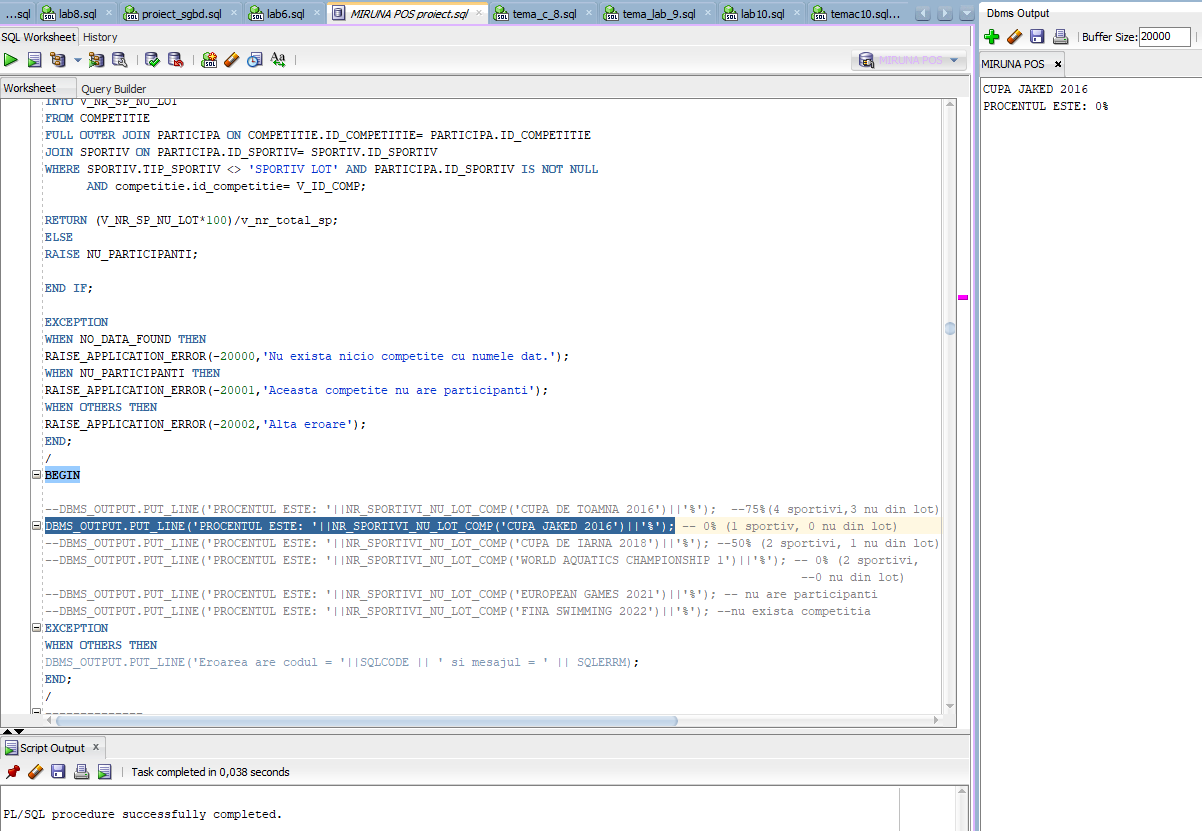
/

BEGIN

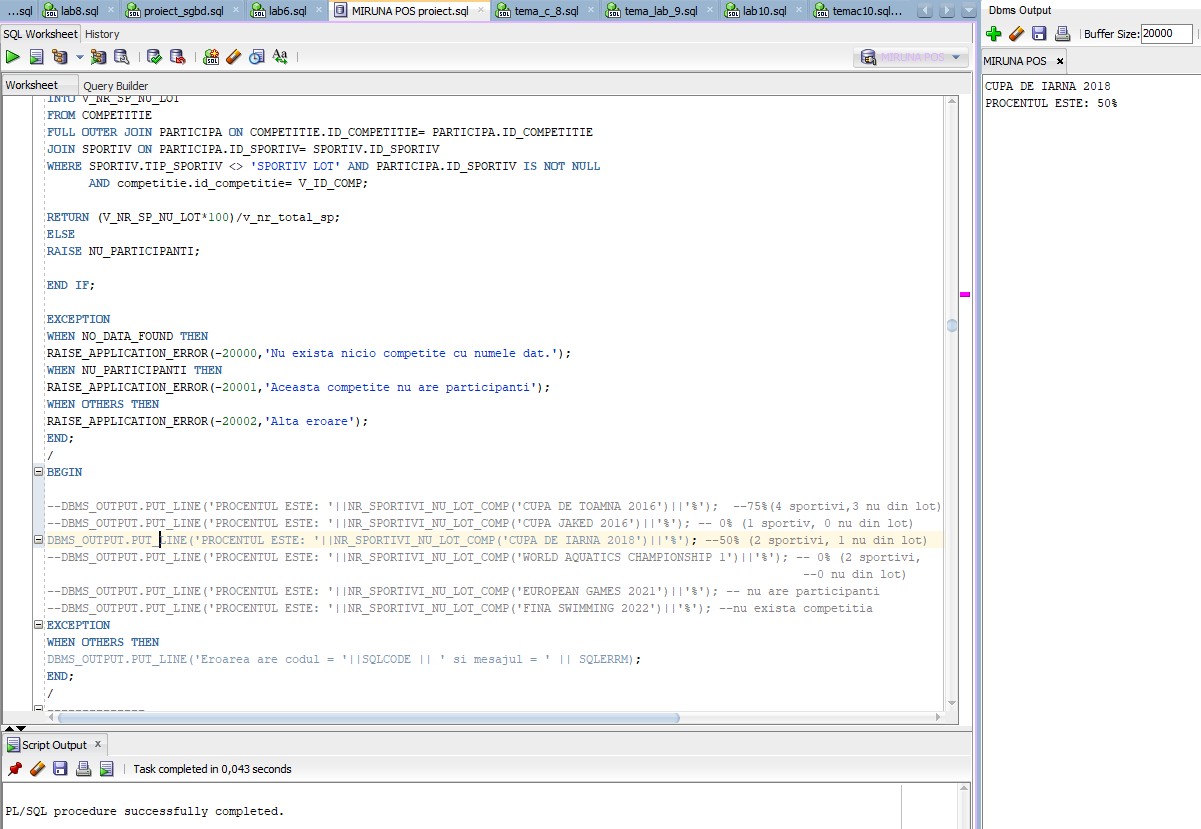
--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA DE TOAMNA 2016')||'%'); --75%(4 sportivi,3 nu din lot)



--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA JAKED 2016')||'%'); -- 0% (1 sportiv, 0 nu din lot)

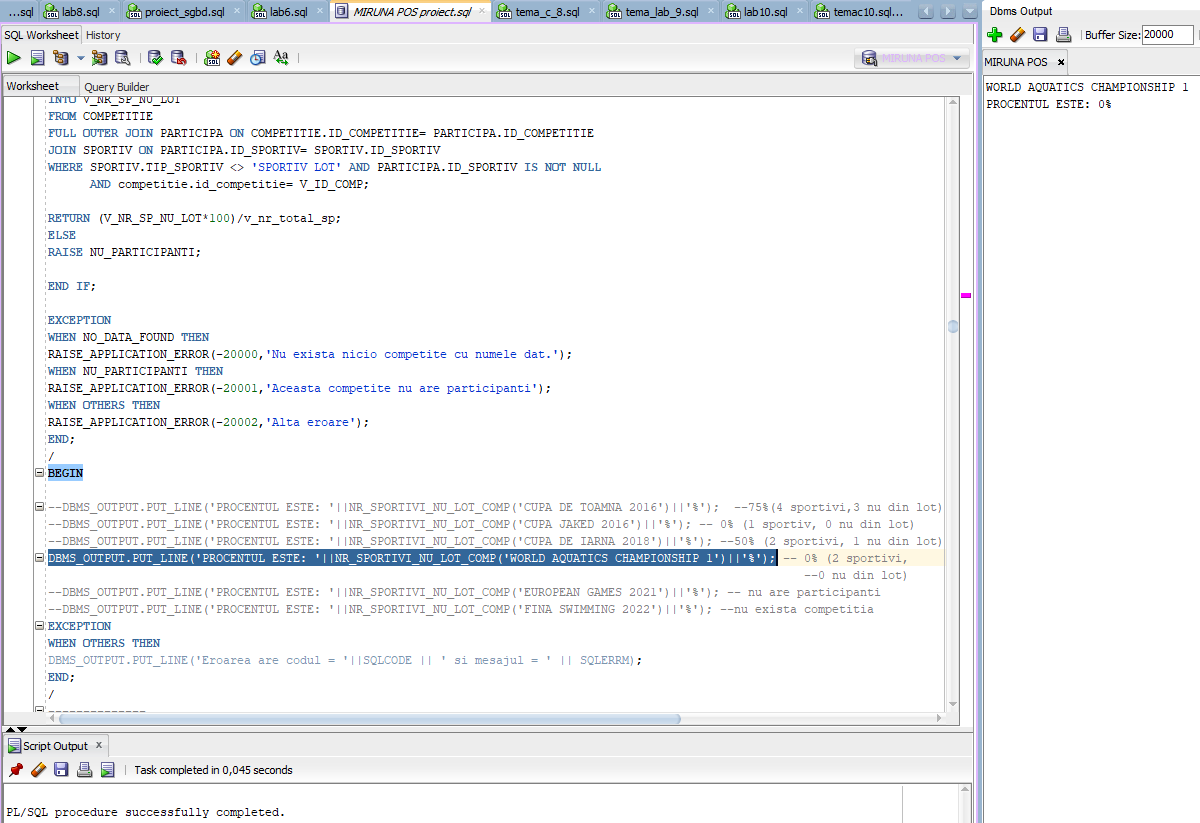


--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA DE IARNA 2018')||'%'); --50% (2 sportivi, 1 nu din lot)

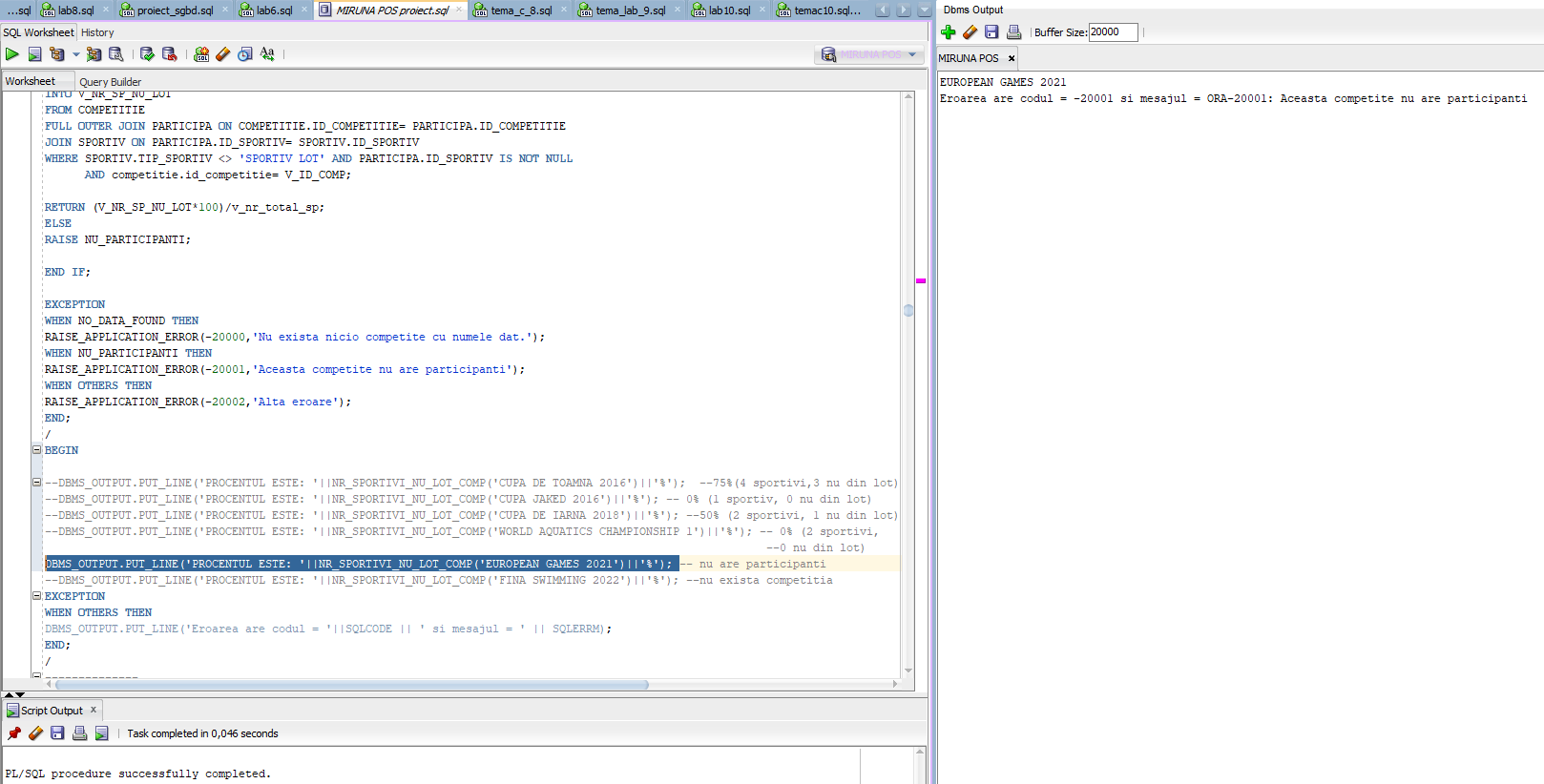


--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('WORLD AQUATICS CHAMPIONSHIP 1')||'%'); -- 0% (2 sportivi,

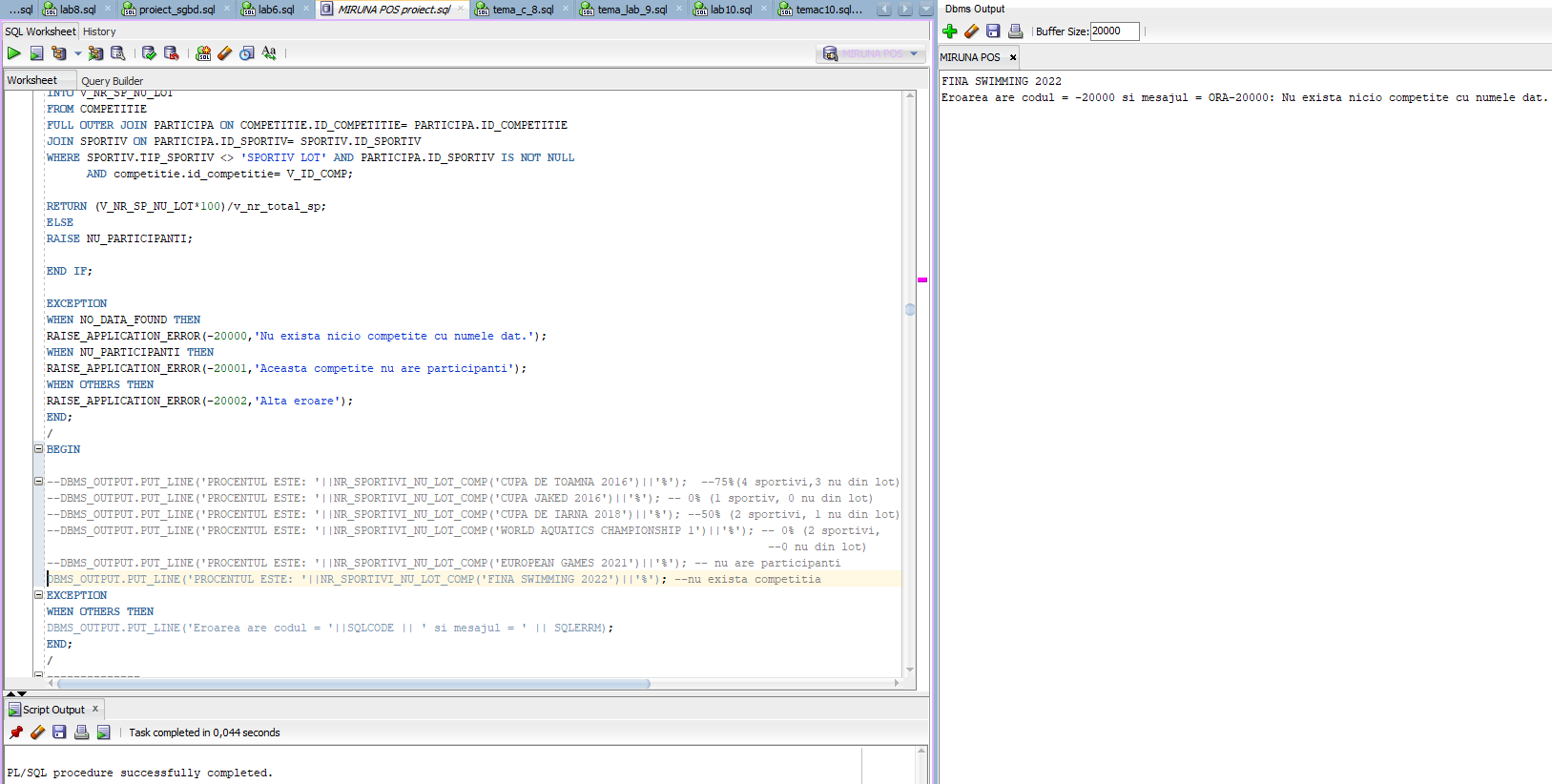
--0 nu din lot)



--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('EUROPEAN GAMES 2021')||'%'); -- nu are participanti



DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||NR\_SPORTIVI\_NU\_LOT\_COMP('FINA SWIMMING 2022')||'%'); --nu exista competitia



EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

END;

/

# 9.

/\***CERINTA:**

Pentru un sportiv al carui nume este dat ca parametru sa se afiseze numele competitiilor la care a participat in Romania care incep primavara sau toamna si in cadrul carora a participat la probe scurte (50 m sau 100 m) , numele probei, dar si locul obtinut in clasament.

Mai mult,ca **cerinta suplimentara**, pentru acest sportiv (dar acum se tine cont de toate competitiile la care a participat) se doreste sa se afle in perioada caror sponsorizari acesta nu a reusit sa obtina nicio medalie sau acesta nu a participat la nicio competitie. Sa se afiseze intervalul de sponsorizare, numele sponsorului si un mesaj corespunzator pentru cazul in care nu a participat la nicio competitie in acel interval,respectiv ca nu a obtinut nicio medalie.

Sa se trateze cazurile in care sportivul cu numele dat nu exista, cat si cazul in care sunt mai multi sportivi au acelasi nume (pentru a obtine aceasta eroare doar pentru acest exercitiu se va insera un sportiv cu numele: BELLU SARA iar dupa rulare se va da ROLLBACK).

\*/

--REZOLVAREA

/\*

PASII REZOLVARII:

1. Pentru a obtine informatiile cerute in primul paragraf se va face **un join pe 8 tabele**

(se va face join si cu istoricul sportivului deoarece acolo se trece mai intai id-ul rezultatului)

De asemenea, joinul cu tabelul REZULTAT se face dupa 3 coloane: id\_proba, id\_sportiv si id\_rezultat

Se pun conditiile in where pentru a obtine competitiile dorite

Datele din select sunt tinute intr-un tablou indexat

In cazul in care tabloul indexat nu va avea valori inserate se va afisa un mesaj corespunzator

2. Pentru rezolvarea celui de-al doilea paragraf se vor folosi doua tabele indexate. In unul din ele vor fi stocate date cu privire la intervalele de sponsorizare ale sportivului si numele sponsorului (join intre 3 tabele),iar in celalalt date cu privire la intervalul de desfasurare al competitiei si daca sportivul a fost sau nu medaliat.

Pentru a verifica daca in intervalul de sponsorizare sportivul nu a obtinut nicio medalie sau nu a participat la niciun concurs se va parcurge mai intai tabloul indexat pentru sponsorizari, si pentru fiecare sponsorizare se va parcurge tabloul indexat pentru competitii. Se va calcula numarul de competitii la care a participat sportivul pe parcursul unui interval de sponsorizare, iar apoi se va verifica la cate din competitii a fost medaliat.

Daca sportivul nu a participat la nicio competitie in intervalul sponsorizarii respective se va afisa un mesaj, iar daca nu a fost deloc medaliat se va afisa un mesaj corespunzator.

Mai mult, daca sportivul nu a fost niciodata sponsorizat se va afisa un mesaj, la fel si pentru cazul in care nu a participat niciodata la o competitie.

3. Exceptiile NO\_DATA\_FOUND si TOO\_MANY\_ROWS vor fi tratate in blocul exception

\*/

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (13,'F','BELLU','SARA',TO\_DATE('16-10-2001','dd-mm-yyyy'),1,'SPORTIV LOT',301,50);

ROLLBACK;

/

create or replace procedure sportiv\_competitii\_sponsor(v\_nume\_sp SPORTIV.NUME%TYPE,v\_pren\_sp SPORTIV.PRENUME%TYPE)

is

v\_id SPORTIV.ID\_SPORTIV%TYPE;

TYPE tip\_record\_cpr IS RECORD

(nume\_comp COMPETITIE.NUME%TYPE,

nume\_proba PROBA.NUME%TYPE,

loc\_clasam REZULTAT.LOC\_CLASAMENT%TYPE);

type tablou\_indexat\_cpr is table of tip\_record\_cpr index by pls\_integer;

t\_indexat\_cpr tablou\_indexat\_cpr;

--pentru partea a doua a exercitiului

TYPE tip\_record\_spons IS RECORD

(spons\_inc SPONSORIZEAZA.INCEPUT\_COLABORARE%TYPE,

spons\_sf SPONSORIZEAZA.SFARSIT\_COLABORARE%TYPE,

spons\_nume SPONSOR.NUME%TYPE);

TYPE tip\_record\_comp IS RECORD

(medalie PARTICIPA.MEDALIAT%TYPE,

comp\_inc COMPETITIE.DATA\_START%TYPE,

comp\_sf COMPETITIE.DATA\_FINAL%TYPE);

type tablou\_indexat is table of tip\_record\_spons index by pls\_integer;

t\_indexat\_sponsor tablou\_indexat;

type tablou\_indexat\_comp is TABLE OF tip\_record\_comp index by pls\_integer;

t\_indexat\_comp tablou\_indexat\_comp;

nr\_medaliati number;

numar\_comp\_int number; --numarul de competitii la care a participat sportivul in intervalul in care a fost sponsorizat

BEGIN

SELECT ID\_SPORTIV

INTO v\_id

FROM SPORTIV

WHERE NUME = v\_nume\_sp AND PRENUME = v\_pren\_sp;

SELECT competitie.nume,proba.nume,rezultat.loc\_clasament

BULK COLLECT INTO t\_indexat\_cpr

FROM SPORTIV

--SUNT LUATI IN CONSIDERARE DOAR SPORTIVII CARE PARTICIPA LA COMPETITIILE DIN ROMANIA

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

JOIN ORAS ON COMPETITIE.ID\_ORAS = ORAS.ID\_ORAS

JOIN TARA ON ORAS.ID\_TARA = TARA.ID\_TARA

join istoric\_sportiv on sportiv.id\_sportiv = istoric\_sportiv.id\_sportiv

--mai intai datele sunt trecute in istoric si dupa in rezultat

JOIN PROBA ON PARTICIPA.ID\_PROBA = PROBA.ID\_PROBA

JOIN REZULTAT ON PROBA.ID\_PROBA = REZULTAT.ID\_PROBA

AND SPORTIV.ID\_SPORTIV = REZULTAT.ID\_SPORTIV

and istoric\_sportiv.id\_rezultat= rezultat.id\_rezultat

WHERE EXTRACT(MONTH FROM COMPETITIE.DATA\_START) IN (3,4,5,9,10,11)

AND TARA.NUME\_TARA = 'ROMANIA'

AND (PROBA.NUME LIKE '100%' OR PROBA.NUME LIKE '50%')

and SPORTIV.ID\_SPORTIV=v\_id;

IF t\_indexat\_cpr.COUNT <>0 THEN

for i in t\_indexat\_cpr.first .. t\_indexat\_cpr.last

loop

dbms\_output.put\_line(t\_indexat\_cpr(i).nume\_comp||' proba: '||t\_indexat\_cpr(i).nume\_proba||' locul: '||t\_indexat\_cpr(i).loc\_clasam);

end loop;

ELSE

DBMS\_OUTPUT.PUT\_LINE('ACEST SPORTIV NU PARTICIPA LA PROBE SCURTE DIN CADRUL COMPETITIILOR DIN ROMANIA');

END IF;

--**partea a doua a exercitiului**

select sponsorizeaza.inceput\_colaborare,sponsorizeaza.sfarsit\_colaborare,SPONSOR.NUME

BULK COLLECT into t\_indexat\_sponsor

from sportiv

join sponsorizeaza on sportiv.id\_sportiv = sponsorizeaza.id\_sportiv

join sponsor on sponsorizeaza.id\_sponsor = sponsor.id\_sponsor

where sportiv.id\_sportiv =v\_id;

SELECT PARTICIPA.MEDALIAT, COMPETITIE.DATA\_START,COMPETITIE.DATA\_FINAL

BULK COLLECT INTO t\_indexat\_comp

FROM SPORTIV

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

where sportiv.id\_sportiv =v\_id;

IF t\_indexat\_sponsor.COUNT <>0 THEN

for i in t\_indexat\_sponsor.FIRST..t\_indexat\_sponsor.LAST

loop --aici intru pe un interval de sponsorizare

IF t\_indexat\_comp.COUNT <>0 THEN

nr\_medaliati:=0;

numar\_comp\_int:=0;

FOR j IN t\_indexat\_comp.FIRST..t\_indexat\_comp.LAST

loop

if t\_indexat\_comp(j).comp\_inc >= t\_indexat\_sponsor(i).spons\_inc

and t\_indexat\_comp(j).comp\_sf <= t\_indexat\_sponsor(i).spons\_sf

then

numar\_comp\_int:=numar\_comp\_int+1; --numarul de competitii la care a participat sportivul in intervalul in care a fost sponsorizat

end if;

if t\_indexat\_comp(j).comp\_inc >= t\_indexat\_sponsor(i).spons\_inc

and t\_indexat\_comp(j).comp\_sf <= t\_indexat\_sponsor(i).spons\_sf

and t\_indexat\_comp(j).medalie =1

then

nr\_medaliati:=nr\_medaliati+1;

end if;

end loop;

if numar\_comp\_int = 0 then

DBMS\_OUTPUT.PUT\_LINE('In intervalul de sponsorizare: '||t\_indexat\_sponsor(i).spons\_inc||' '||t\_indexat\_sponsor(i).spons\_sf||

' cu sponsorul: '||t\_indexat\_sponsor(i).spons\_nume||' sportivul nu a participat la nicio competitie');

elsif nr\_medaliati = 0 then

DBMS\_OUTPUT.PUT\_LINE('In intervalul de sponsorizare: '||t\_indexat\_sponsor(i).spons\_inc||' '||t\_indexat\_sponsor(i).spons\_sf||

' cu sponsorul: '||t\_indexat\_sponsor(i).spons\_nume||' sportivul nu a reusit sa obtina nicio medalie din '||numar\_comp\_int||' competitii la care a participat');

end if;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Sportivul nu a participat la nicio competitie pana acum');

END IF;

end loop;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Sportivul nu a fost si nici nu este sponsorizat');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000,'NU EXISTA SPORTIVUL');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001,' SUNT MAI MULTI SPORTIVI CU ACELASI NUME');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002,'ALTA EROARE');

END;

/

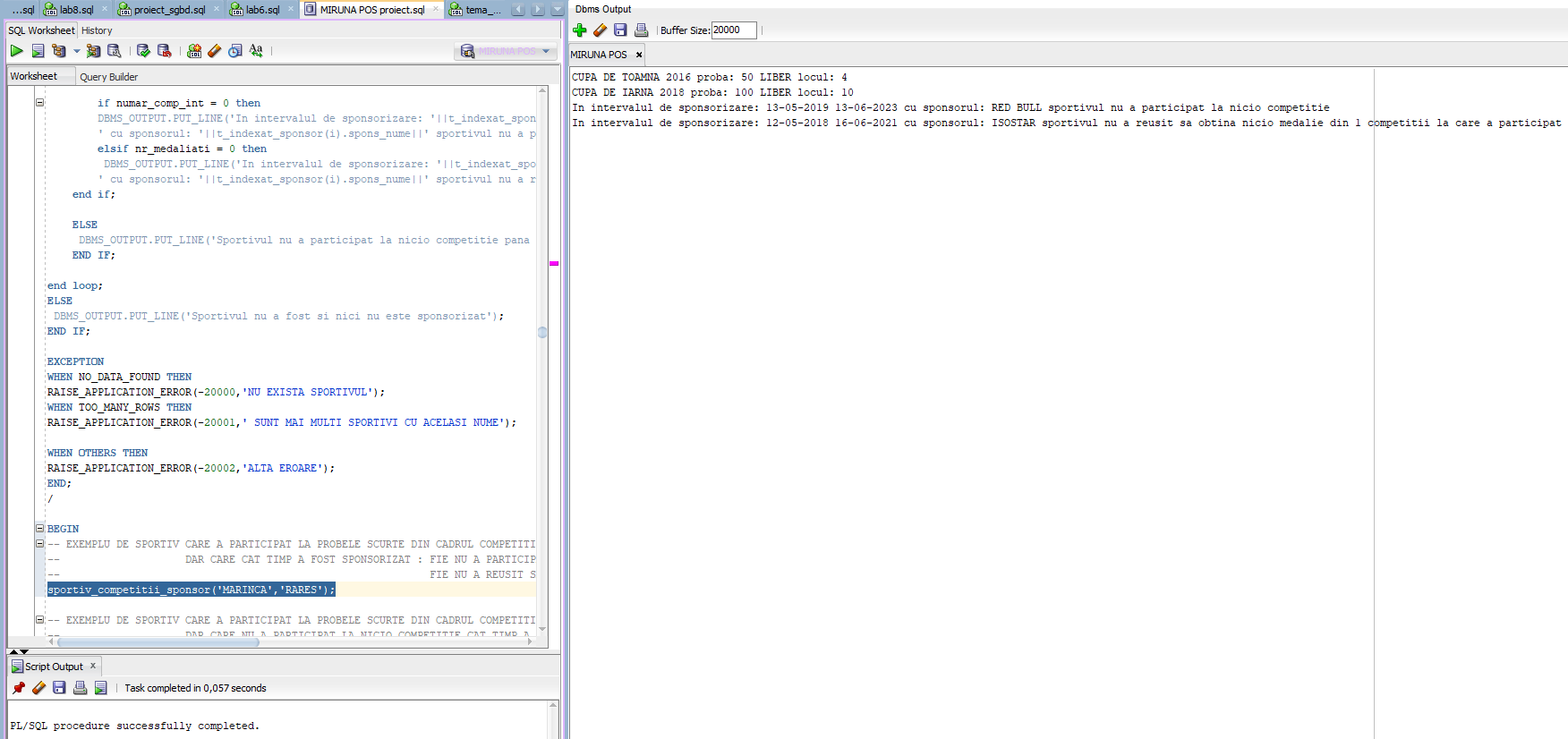
BEGIN

-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE CAT TIMP A FOST SPONSORIZAT : FIE NU A PARTICIPAT LA NICIO COMPETITIE

-- FIE NU A REUSIT SA OBTINA NICIO MEDALIE

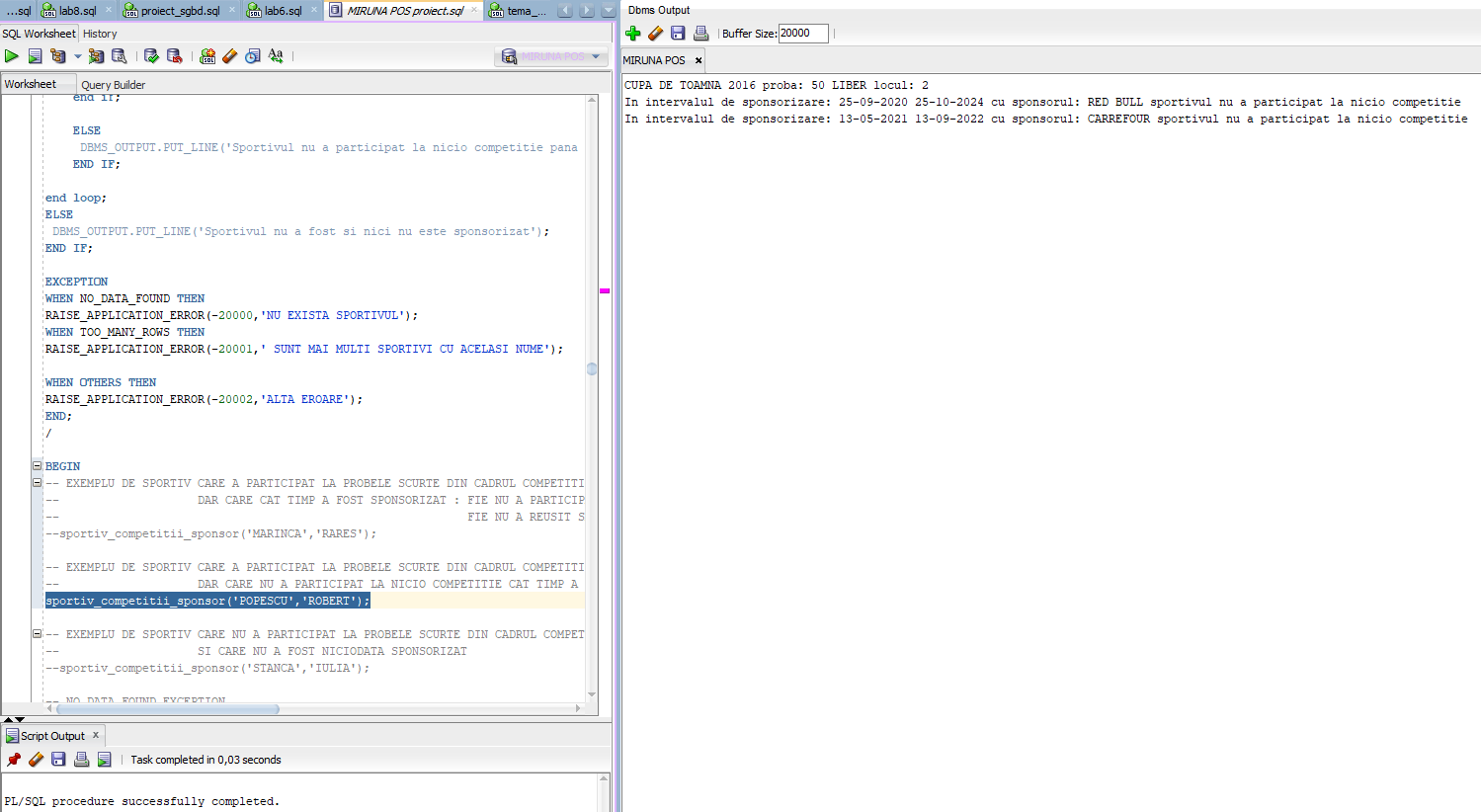
--sportiv\_competitii\_sponsor('MARINCA','RARES');



-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE NU A PARTICIPAT LA NICIO COMPETITIE CAT TIMP A FOST SPOSORIZAT

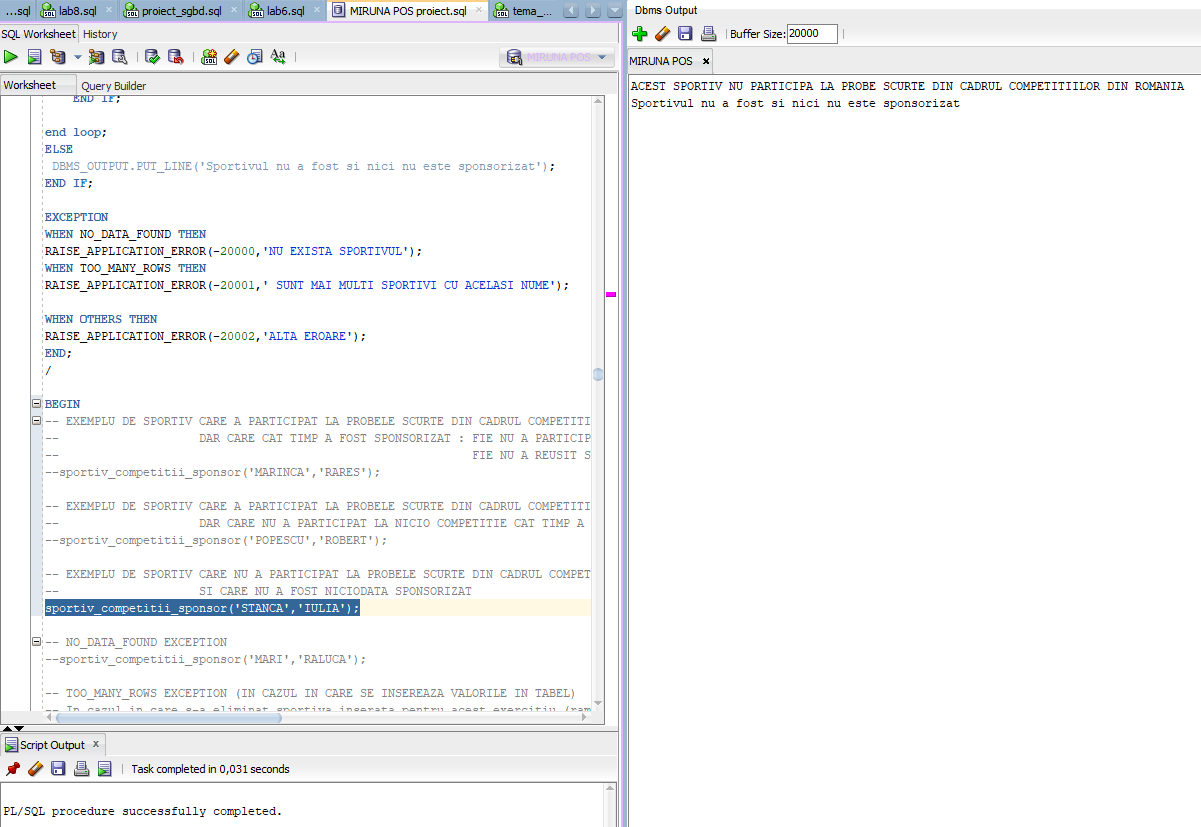
--sportiv\_competitii\_sponsor('POPESCU','ROBERT');



-- EXEMPLU DE SPORTIV CARE NU A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

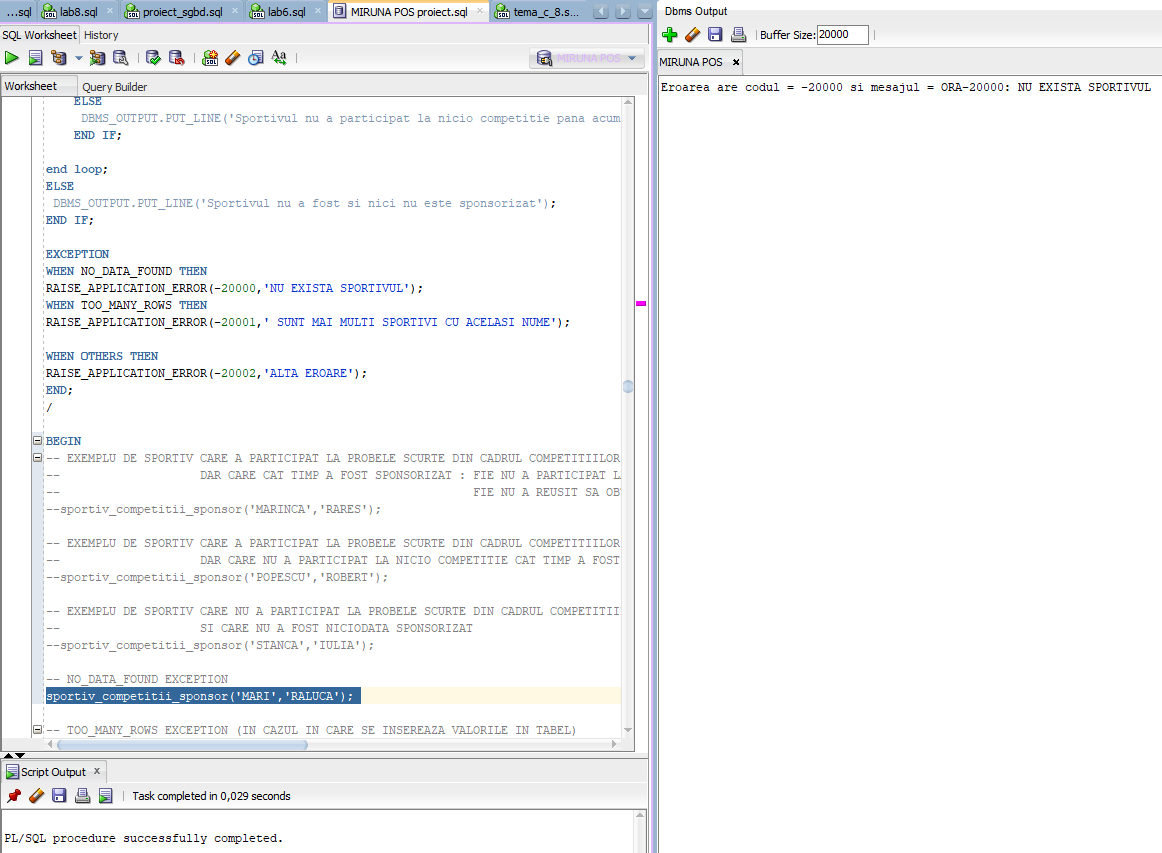
-- SI CARE NU A FOST NICIODATA SPONSORIZAT

--sportiv\_competitii\_sponsor('STANCA','IULIA');

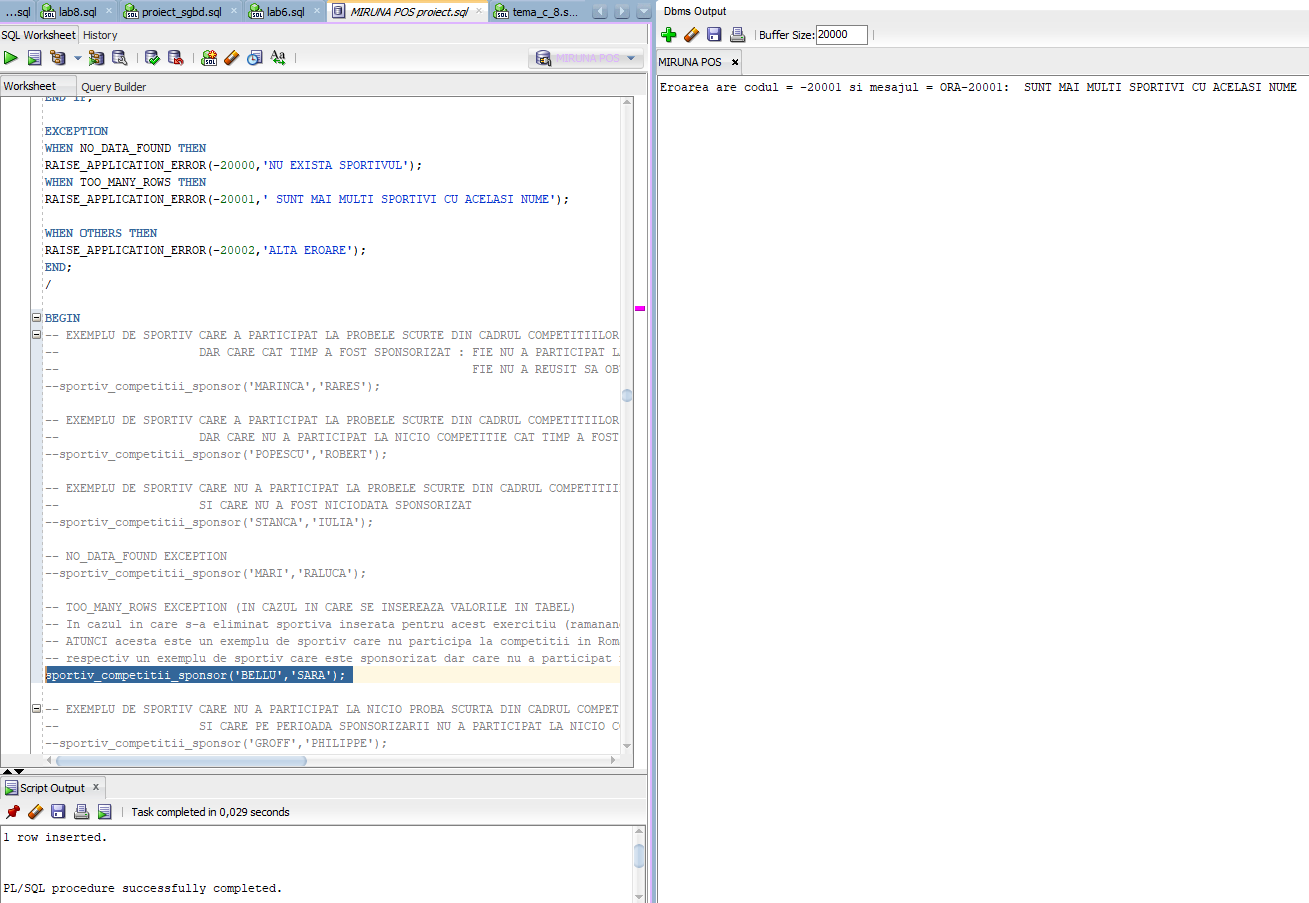


-- NO\_DATA\_FOUND EXCEPTION

--sportiv\_competitii\_sponsor('MARI','RALUCA');



-- TOO\_MANY\_ROWS EXCEPTION (IN CAZUL IN CARE SE INSEREAZA VALORILE IN TABEL)

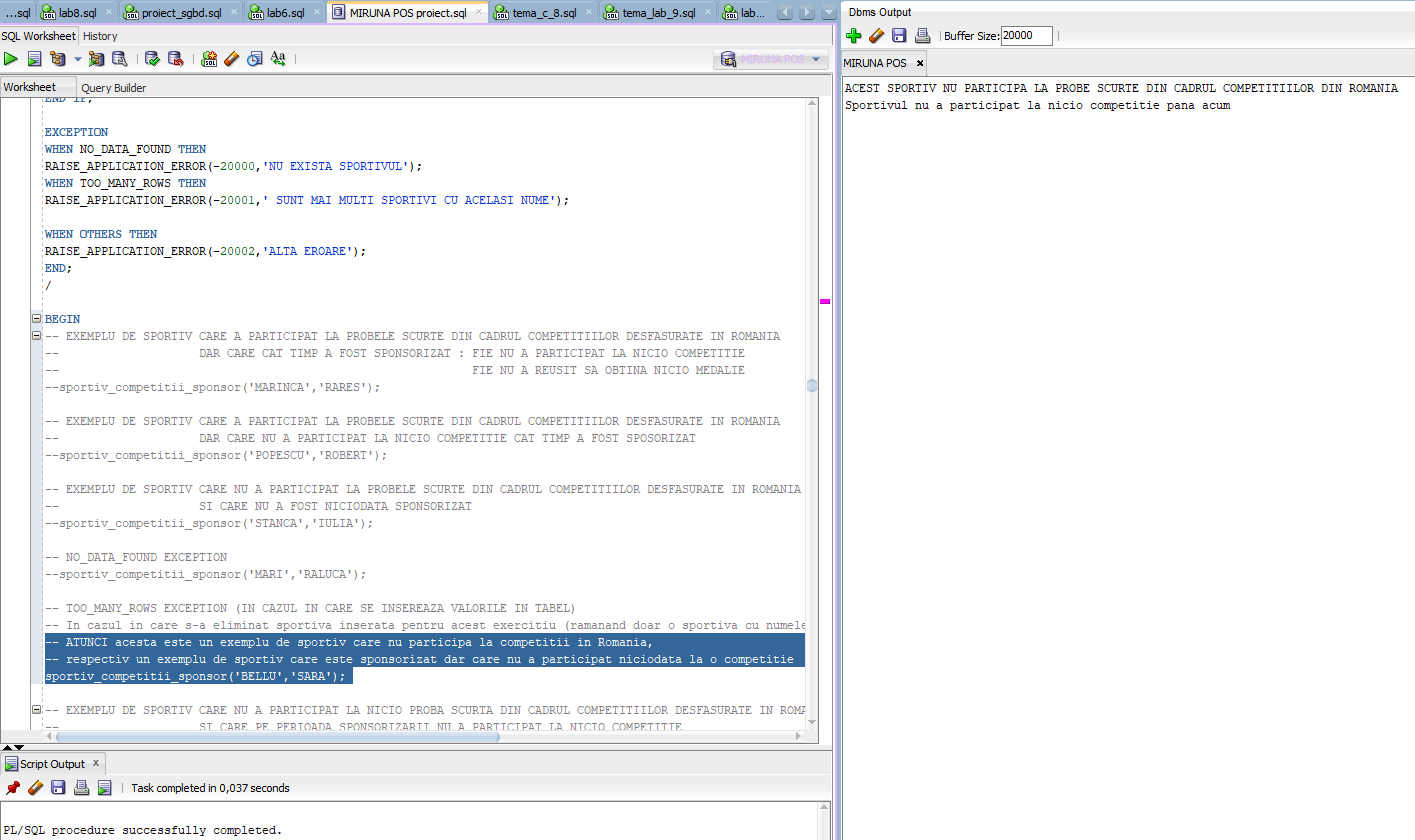


-- In cazul in care s-a eliminat sportiva inserata pentru acest exercitiu (ramanand doar o sportiva cu numele BELLU SARA)

-- ATUNCI acesta este un exemplu de sportiv care nu participa la competitii in Romania,

-- respectiv un exemplu de sportiv care este sponsorizat dar care nu a participat niciodata la o competitie

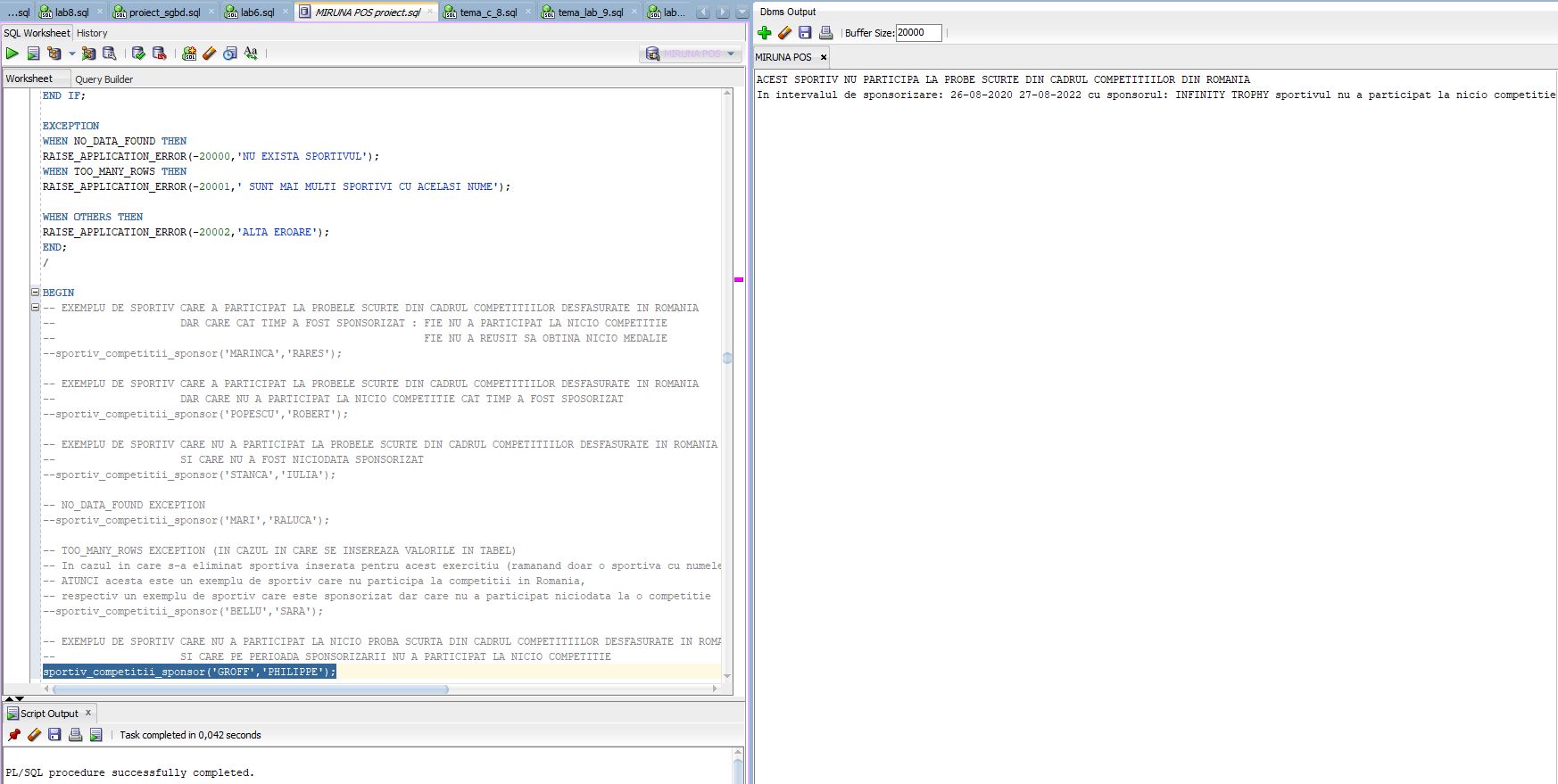
--sportiv\_competitii\_sponsor('BELLU','SARA');



-- EXEMPLU DE SPORTIV CARE NU A PARTICIPAT LA NICIO PROBA SCURTA DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- SI CARE PE PERIOADA SPONSORIZARII NU A PARTICIPAT LA NICIO COMPETITIE

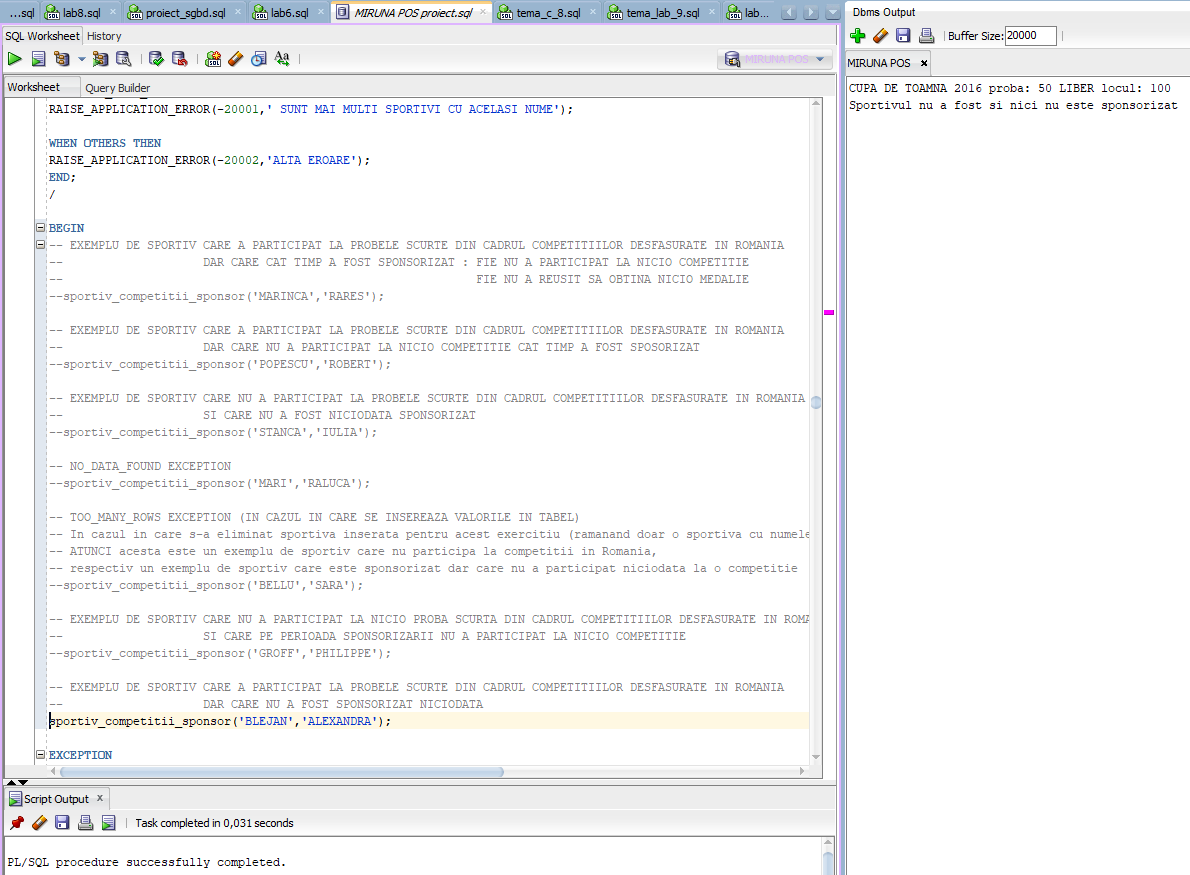
--sportiv\_competitii\_sponsor('GROFF','PHILIPPE');



-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE NU A FOST SPONSORIZAT NICIODATA

sportiv\_competitii\_sponsor('BLEJAN','ALEXANDRA');



EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

END;

/

# 10.

/\***CERINTA**:

Creati un **trigger la nivel de comanda** care sa ii permita administratorul bazei de date ('MIRUNA') sa insereze in tabela Antrenor doar in zilele de 2 ale lunilor (in aceasta zi se dau testele de certificare ca antrenor): ianuarie, martie, iunie, septembrie si decembrie intre orele 8 a.m. si 11 a.m. Mai mult, numarul de antrenori nu poate sa depaseasca numarul de 600 de persoane deoarece acesta este maximul admis de federatie.

In afara acestui program sunt interzise modificarile de catre administrator.

\*/

/

CREATE OR REPLACE TRIGGER trigger\_antrenor

BEFORE INSERT

ON ANTRENOR

DECLARE

v\_antrenori number;

BEGIN

SELECT COUNT(id\_antrenor)

INTO v\_antrenori

FROM antrenor;

IF (EXTRACT (MONTH FROM SYSDATE) IN (1,3,6,9,12)) AND (TO\_CHAR (SYSDATE,'hh24') between 8 and 11) and

user = UPPER('miruna') AND (EXTRACT (DAY FROM SYSDATE))=2 THEN

--conditia de if de mai jos este pentru a face verificari (imi dau permisiunea pentru ziua in care sunt ca sa pot verifica

-- conditiile de inserare cand se depasesc 600 de antrenori)

--IF (EXTRACT (MONTH FROM SYSDATE) IN (1,3,6,9,12)) AND (TO\_CHAR (SYSDATE,'hh24') between 0 and 24) and

--user = UPPER('miruna') AND (EXTRACT (DAY FROM SYSDATE))=23 THEN

IF v\_antrenori >=600 then

RAISE\_APPLICATION\_ERROR(-20003,'S-a atins limita de antrenori care pot fi introdusi');

else

DBMS\_OUTPUT.PUT\_LINE('Inserarea a avut loc cu succes');

end if;

ELSE

RAISE\_APPLICATION\_ERROR(-20001,'Administratorul nu poate insera in tabelul ANTRENOR');

end if;

END;

/

select\*from antrenor;

/

begin

FOR I IN 121..811 LOOP

insert into antrenor (id\_antrenor,nume,prenume,anul\_nasterii,gen,ani\_experienta,salariu,id\_club)

values(I,'nume','prenume',sysdate,'M',1,501,70);

END LOOP;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

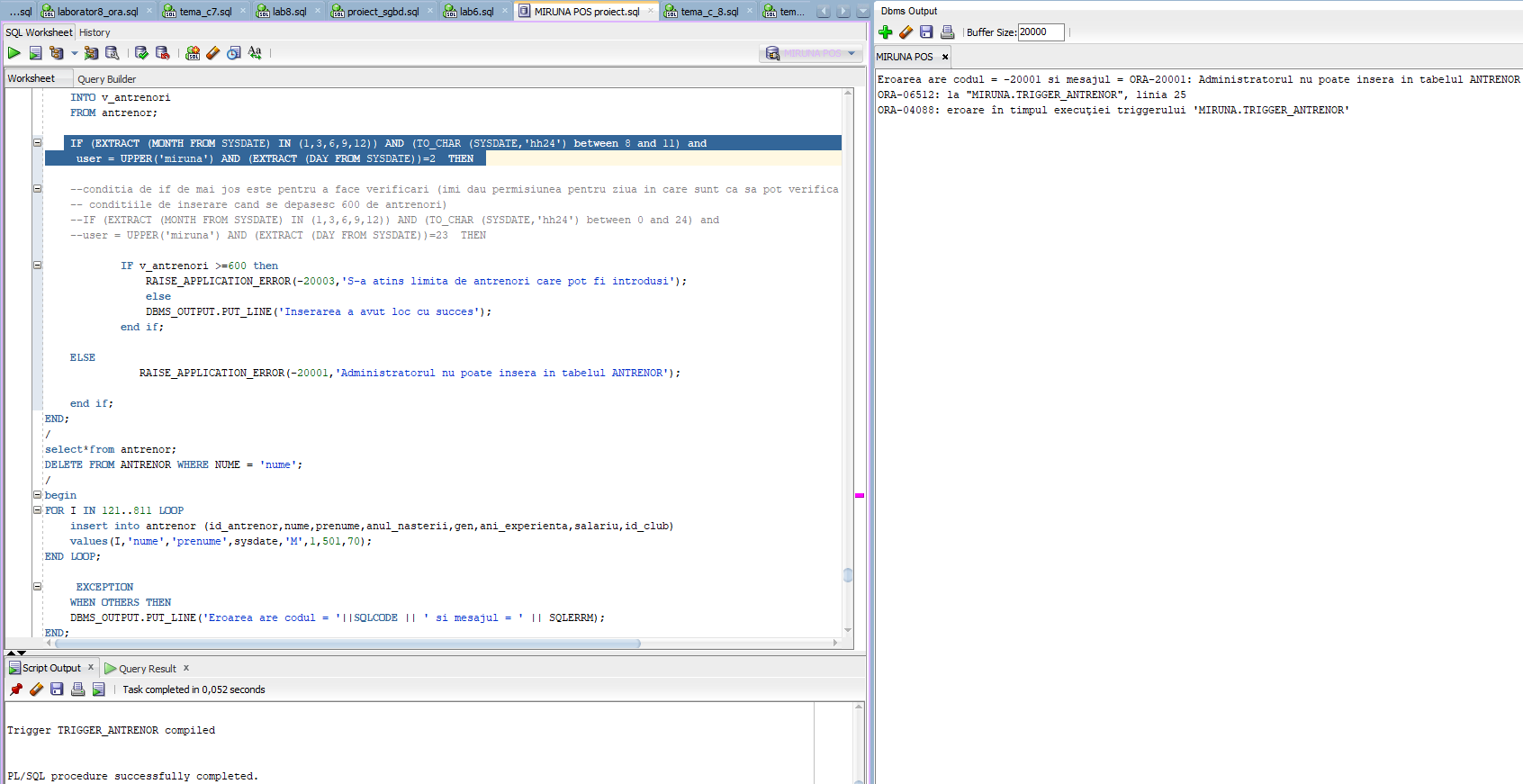
END;

/

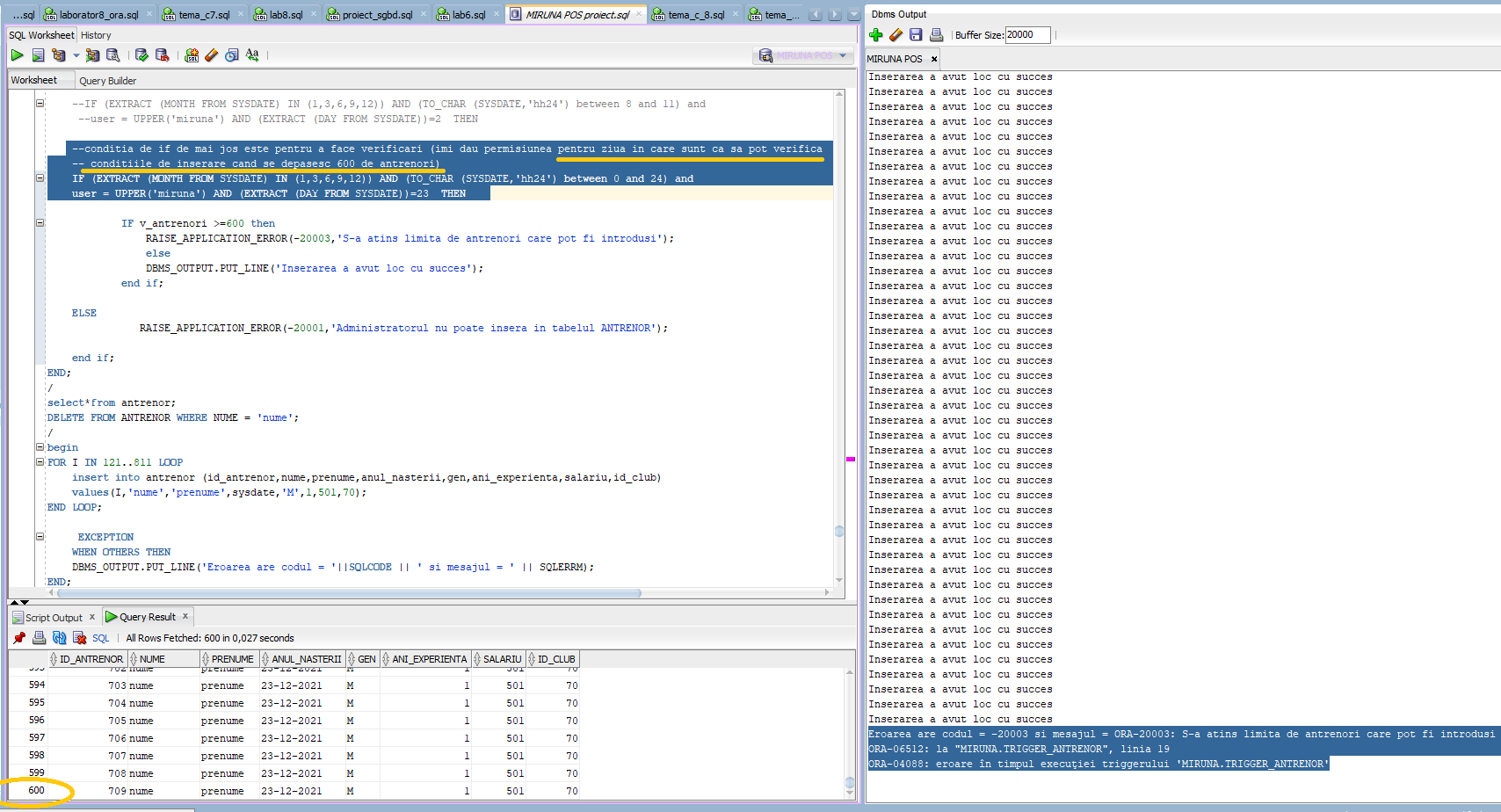
rollback;

/

Când nu se îndeplinesc condițiile din instrucțiunea IF privind data



Când se îndeplinesc condițiile privind ziua și ora, dar se introduc prea mulți antrenori.



# 11.

--**CERINTA**: Creati un **trigger** **la nivel de linie** care sa nu ii permita unui sponsor sa inceapa colaborarea cu mai mult de 4 sportivi anual.

/

CREATE OR REPLACE PACKAGE pachet\_sponsorizari

AS

TYPE tip\_record IS RECORD

(id\_sp SPONSORIZEAZA.ID\_SPONSOR%TYPE,

nr NUMBER(5));

TYPE tip\_indexat IS TABLE OF tip\_record INDEX BY PLS\_INTEGER;

t tip\_indexat;

contor NUMBER(2):=0;

END;

/

CREATE OR REPLACE TRIGGER triger\_comanda\_spnosorizare

BEFORE INSERT on sponsorizeaza

BEGIN

pachet\_sponsorizari.contor:=0;

SELECT id\_sponsor, count(\*)--numarul de sportivi pe care ii sponsorizeaza

BULK COLLECT INTO pachet\_sponsorizari.t

FROM SPONSORIZEAZA

WHERE EXTRACT(YEAR FROM INCEPUT\_COLABORARE) = EXTRACT(YEAR FROM SYSDATE)

GROUP BY ID\_SPONSOR;

END;

/

CREATE OR REPLACE TRIGGER triger\_linii\_spnosorizare

BEFORE INSERT ON SPONSORIZEAZA

FOR EACH ROW

BEGIN

FOR i IN 1..pachet\_sponsorizari.t.last LOOP

IF pachet\_sponsorizari.t(i).id\_sp = :NEW.id\_sponsor

AND pachet\_sponsorizari.t(i).nr + pachet\_sponsorizari.contor =4

--daca sponsorizeaza 4

THEN

RAISE\_APPLICATION\_ERROR(-20030,'Sponsorul cu id-ul '||:NEW.id\_sponsor||' depaseste numarul maxim de sportivi pe care

ii poate sponsoriza incepand din acest an');

END IF;

END LOOP;

pachet\_sponsorizari.contor:=pachet\_sponsorizari.contor+1;

END;

/

--TESTARE

INSERT INTO sponsorizeaza(ID\_SPONSOR,ID\_SPORTIV,INCEPUT\_COLABORARE,SFARSIT\_COLABORARE)

VALUES (1,11,to\_date('23-12-2021','dd-mm-yyyy'),to\_date('30-04-2024','dd-mm-yyyy'));

INSERT INTO sponsorizeaza(ID\_SPONSOR,ID\_SPORTIV,INCEPUT\_COLABORARE,SFARSIT\_COLABORARE)

VALUES (1,5,to\_date('30-10-2021','dd-mm-yyyy'),to\_date('25-04-2025','dd-mm-yyyy'));

INSERT INTO sponsorizeaza(ID\_SPONSOR,ID\_SPORTIV,INCEPUT\_COLABORARE,SFARSIT\_COLABORARE)

VALUES (1,4,to\_date('23-11-2021','dd-mm-yyyy'),to\_date('30-03-2024','dd-mm-yyyy'));

INSERT INTO sponsorizeaza(ID\_SPONSOR,ID\_SPORTIV,INCEPUT\_COLABORARE,SFARSIT\_COLABORARE)

VALUES (1,6,to\_date('28-11-2021','dd-mm-yyyy'),to\_date('30-06-2023','dd-mm-yyyy'));

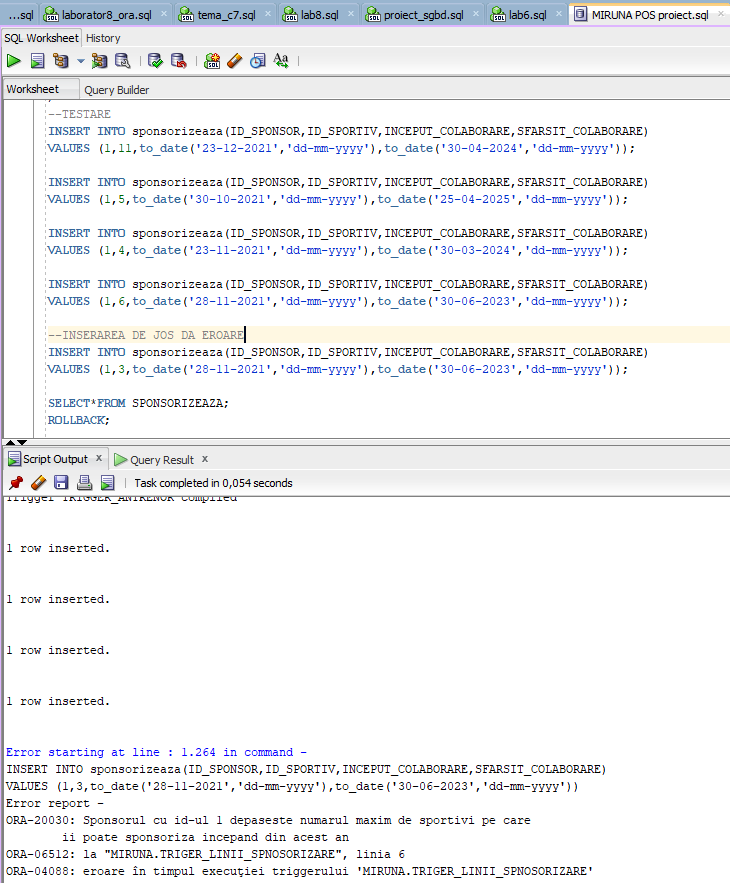
--INSERAREA DE JOS DA EROARE

INSERT INTO sponsorizeaza(ID\_SPONSOR,ID\_SPORTIV,INCEPUT\_COLABORARE,SFARSIT\_COLABORARE)

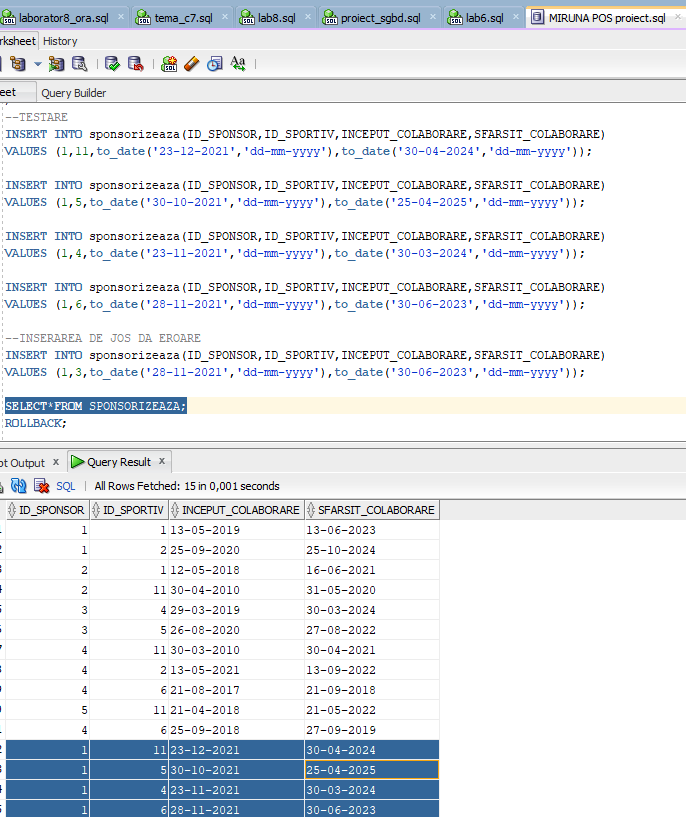
VALUES (1,3,to\_date('28-11-2021','dd-mm-yyyy'),to\_date('30-06-2023','dd-mm-yyyy'));

SELECT\*FROM SPONSORIZEAZA;

ROLLBACK;



**Modificarea apărută în tabelul SPONSORIZEAZĂ**



# 12.

/\***CERINTA**:

Sa se creeze un **TRIGGER LDD** care sa retina utilizatorul,data si actiunile facute de utilizator pe schema.

Mai mult, doar administratorul bazei de date ('MIRUNA') poate face modificarile de tip LDD;

in zilele de duminica nu se permit modificari la nivelul schemei, pentru ca este zi libera.

\*/

/

CREATE TABLE info\_user

(numele\_user VARCHAR2(40),

evenimentul VARCHAR2(40),

numele\_obiect VARCHAR2(40),

data\_modificare DATE,

numele\_baza\_date VARCHAR2(40)

);

/

CREATE OR REPLACE TRIGGER date\_utilizator\_modif

BEFORE CREATE OR DROP OR ALTER

ON SCHEMA

BEGIN

IF TO\_CHAR(SYSDATE,'D') = 7 THEN

RAISE\_APPLICATION\_ERROR(-20010,'Duminica nu se lucreaza');

ELSIF user != UPPER('miruna') THEN

RAISE\_APPLICATION\_ERROR(-20011,'Doar administratorul are dreptul sa creeze,altereze sau sa scoata tabele');

ELSIF TO\_CHAR(SYSDATE,'D') != 7 AND user = UPPER('miruna') THEN

INSERT INTO info\_user (numele\_user,evenimentul,numele\_obiect,data\_modificare,numele\_baza\_date)

VALUES (SYS.LOGIN\_USER,SYS.SYSEVENT,SYS.DICTIONARY\_OBJ\_NAME,SYSDATE,SYS.DATABASE\_NAME);

DBMS\_OUTPUT.PUT\_LINE('Comanda realizata cu succes');

END IF;

END;

/

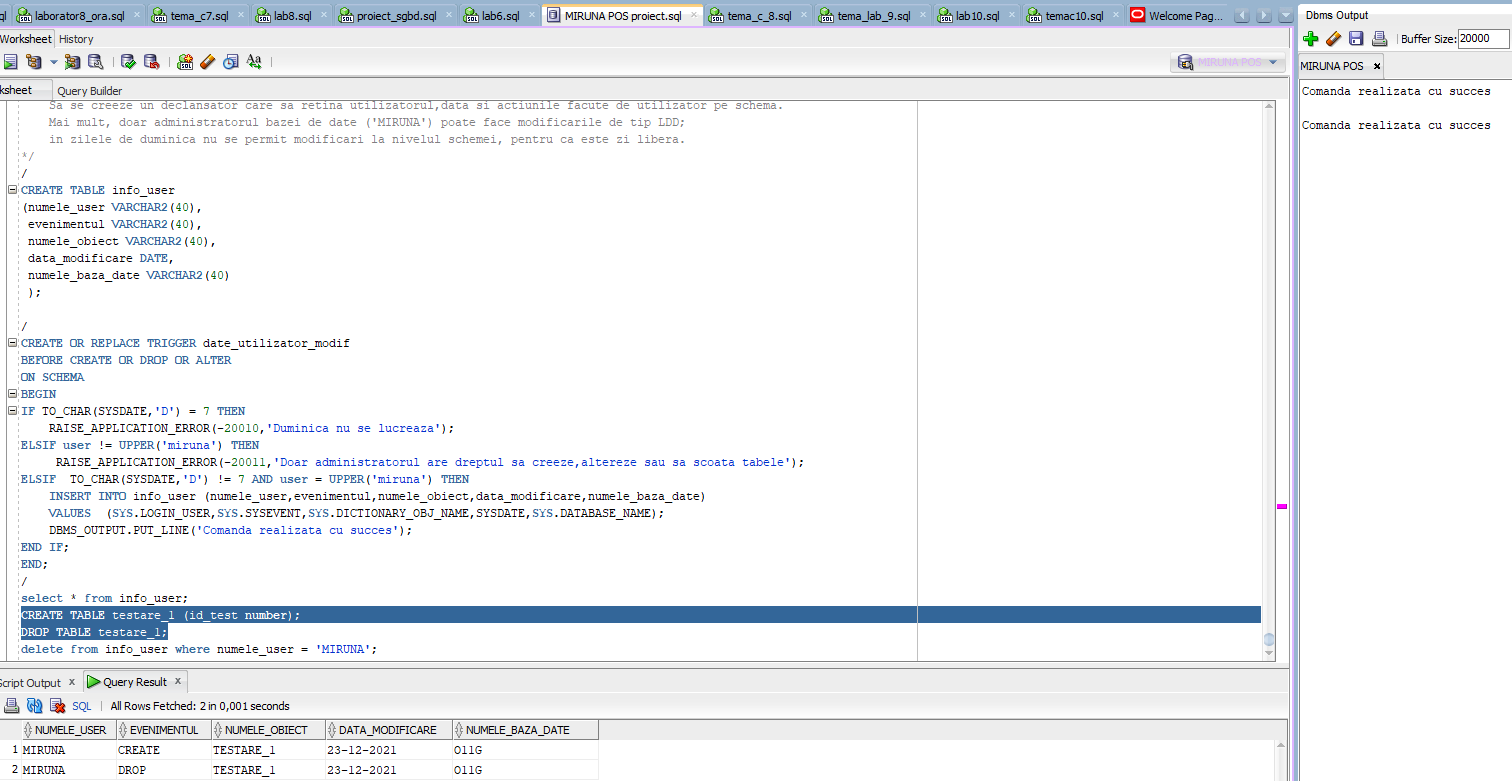
select \* from info\_user;

CREATE TABLE testare\_1 (id\_test number);

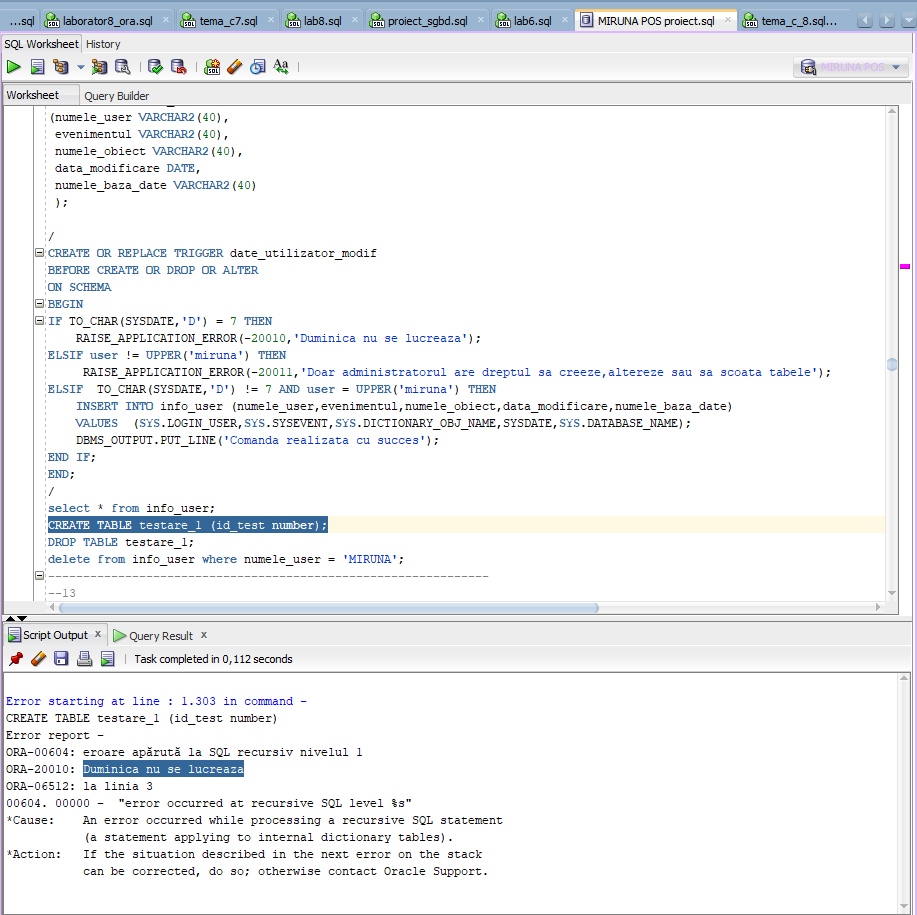
DROP TABLE testare\_1;

delete from info\_user where numele\_user = 'MIRUNA';

**În zile diferite de duminică modificările pot fi făcute:**



**În zilele de duminică:**



# 13. Pachetul proiectului

CREATE OR REPLACE PACKAGE pachet\_proiect

IS

procedure competitie\_cheltuiala\_2; --EX 6

procedure club\_antr\_sp;--EX 7

FUNCTION NR\_SPORTIVI\_NU\_LOT\_COMP(V\_NUME COMPETITIE.NUME%TYPE)--EX 8

RETURN NUMBER;

procedure sportiv\_competitii\_sponsor(v\_nume\_sp SPORTIV.NUME%TYPE,v\_pren\_sp SPORTIV.PRENUME%TYPE);--EX 9

END pachet\_proiect;

/

CREATE OR REPLACE PACKAGE BODY pachet\_proiect

IS

procedure competitie\_cheltuiala\_2

is

--ex 6

TYPE tip\_record IS RECORD

(nume\_comp COMPETITIE.NUME%TYPE,

nume\_sp SPORTIV.NUME%TYPE,

prenume\_sp SPORTIV.PRENUME%TYPE,

taxa COMPETITIE.TAXA\_PARTICIPARE%TYPE,

id\_sp SPORTIV.ID\_SPORTIV%TYPE);

type tablou\_indexat is table of tip\_record index by pls\_integer;

type tablou\_imbricat is table of tip\_record;

t\_indexat\_aq tablou\_indexat;

t\_imbricat\_bsc tablou\_imbricat:= tablou\_imbricat();

suma\_bsc number:=0;

suma\_aq number:=0;

begin

SELECT distinct(COMPETITIE.NUME),SPORTIV.NUME,SPORTIV.PRENUME,COMPETITIE.TAXA\_PARTICIPARE,SPORTIV.ID\_SPORTIV

bulk collect INTO t\_indexat\_aq

FROM SPORTIV

JOIN ANTRENOR ON SPORTIV.ID\_ANTRENOR = ANTRENOR.ID\_ANTRENOR

JOIN CLUB ON ANTRENOR.ID\_CLUB = CLUB.ID\_CLUB

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

WHERE NUME\_CLUB = 'AQUA TEAM'

AND TIP\_SPORTIV <> 'SPORTIV LOT'

AND MONTHS\_BETWEEN(sysdate,COMPETITIE.DATA\_FINAL)<=72

order by sportiv.id\_sportiv;--6 ANI

SELECT distinct(COMPETITIE.NUME),SPORTIV.NUME,SPORTIV.PRENUME,COMPETITIE.TAXA\_PARTICIPARE,SPORTIV.ID\_SPORTIV

bulk collect INTO t\_imbricat\_bsc

FROM SPORTIV

JOIN ANTRENOR ON SPORTIV.ID\_ANTRENOR = ANTRENOR.ID\_ANTRENOR

JOIN CLUB ON ANTRENOR.ID\_CLUB = CLUB.ID\_CLUB

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

WHERE NUME\_CLUB ='BUCHAREST SPORT CLUB ELITE'

AND TIP\_SPORTIV <> 'SPORTIV LOT'

AND MONTHS\_BETWEEN(sysdate,COMPETITIE.DATA\_FINAL)<=72 --6 ANI

order by sportiv.id\_sportiv;

DBMS\_OUTPUT.PUT\_LINE('CLUBUL AQUA TEAM');

DBMS\_OUTPUT.NEW\_LINE();

if t\_indexat\_aq.count >0 then

for i in t\_indexat\_aq.FIRST..t\_indexat\_aq.LAST LOOP

if t\_indexat\_aq.exists(i) then

suma\_aq:=suma\_aq+t\_indexat\_aq(i).taxa;

if i = t\_indexat\_aq.FIRST then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_sp||' '||t\_indexat\_aq(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

end if;

if i>=t\_indexat\_aq.FIRST+1 and t\_indexat\_aq(i).id\_sp <> t\_indexat\_aq(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_sp||' '||t\_indexat\_aq(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

elsif i>=t\_indexat\_aq.FIRST+1 and t\_indexat\_aq(i).id\_sp = t\_indexat\_aq(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_indexat\_aq(i).nume\_comp);

end if;

end if;

END LOOP;

end if;

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('Clubul AQUA TEAM a cheltuit: '||suma\_aq||' lei');

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('CLUBUL BUCHAREST SPORT CLUB ELITE');

DBMS\_OUTPUT.NEW\_LINE();

if t\_imbricat\_bsc.count>0 then

for i in t\_imbricat\_bsc.FIRST..t\_imbricat\_bsc.LAST LOOP

if t\_imbricat\_bsc.exists(i) then

suma\_bsc:=suma\_bsc+t\_imbricat\_bsc(i).taxa;

if i = t\_imbricat\_bsc.FIRST then --primul rand din colectie

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_sp||' '||t\_imbricat\_bsc(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

end if;

if i>=t\_imbricat\_bsc.FIRST+1 and t\_imbricat\_bsc(i).id\_sp <> t\_imbricat\_bsc(i-1).id\_sp --restul randurilor (daca am trecut la un alt sportiv)

then

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_sp||' '||t\_imbricat\_bsc(i).prenume\_sp||' '||'a concurat la competitia: ');

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

elsif i>=t\_imbricat\_bsc.FIRST+1 and t\_imbricat\_bsc(i).id\_sp = t\_imbricat\_bsc(i-1).id\_sp

then

DBMS\_OUTPUT.PUT\_LINE(t\_imbricat\_bsc(i).nume\_comp);

end if;

end if;

END LOOP;

end if;

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE('Clubul BUCHAREST SPORT CLUB ELITE a cheltuit: '||suma\_bsc||' lei');

end;

--ex 7

procedure club\_antr\_sp

is

i number;

venit\_lunar number;

BEGIN

for v\_club in (select club.nume\_club, club.id\_club

from club

join antrenor on club.id\_club = antrenor.id\_club

where antrenor.id\_antrenor in ( select antrenor.id\_antrenor

from antrenor

join sportiv on antrenor.id\_antrenor= sportiv.id\_antrenor

join sponsorizeaza on sportiv.id\_sportiv = sponsorizeaza.id\_sportiv

join participa on sportiv.id\_sportiv = participa.id\_sportiv

where sponsorizeaza.inceput\_colaborare <=sysdate and sponsorizeaza.sfarsit\_colaborare>=sysdate and medaliat=1

group by antrenor.id\_antrenor)

)

loop

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE('Clubul '||v\_club.nume\_club);

DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------------------------------------------------');

i:=0;

venit\_lunar:=0;

for v\_antr in ( select nume,prenume,salariu

from antrenor

where id\_club = v\_club.id\_club

)

loop

i:=i+1;

venit\_lunar:=venit\_lunar+ v\_antr.salariu;

DBMS\_OUTPUT.PUT\_LINE(i||'. '||v\_antr.nume||' '||v\_antr.prenume);

end loop;

DBMS\_OUTPUT.NEW\_LINE;

DBMS\_OUTPUT.PUT\_LINE('Numarul antrenorilor este de: '||i);

DBMS\_OUTPUT.PUT\_LINE('Valoarea lunara a veniturilor antrenorilor este de: '||venit\_lunar);

if i<>0 then

DBMS\_OUTPUT.PUT\_LINE('Valoarea medie a veniturilor antrenorilor este de: '||venit\_lunar/i);

end if;

end loop;

end;

--ex 8

FUNCTION NR\_SPORTIVI\_NU\_LOT\_COMP(V\_NUME COMPETITIE.NUME%TYPE)

RETURN NUMBER

IS

V\_ID\_COMP COMPETITIE.ID\_COMPETITIE%TYPE;

V\_ID\_C COMPETITIE.ID\_COMPETITIE%TYPE;

V\_NR\_TOTAL\_SP NUMBER;

V\_NR\_SP\_NU\_LOT NUMBER;

NU\_PARTICIPANTI EXCEPTION;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(V\_NUME);

SELECT ID\_COMPETITIE

INTO V\_ID\_COMP

FROM COMPETITIE

WHERE COMPETITIE.NUME = V\_NUME; --NO DATA FOUND CAND INTRODUC UN NUME CARE NU EXISTA

SELECT COUNT(competitie.id\_competitie)--INTOARCE 0 PENTRU COMPETITIILE CARE EXISTA DAR CARE NU AU PARTICIPANTI

INTO V\_NR\_TOTAL\_SP

FROM COMPETITIE

FULL OUTER JOIN PARTICIPA ON COMPETITIE.ID\_COMPETITIE= PARTICIPA.ID\_COMPETITIE

WHERE ID\_SPORTIV IS NOT NULL AND competitie.id\_competitie= V\_ID\_COMP;

IF V\_NR\_TOTAL\_SP <>0 THEN

SELECT COUNT(COMPETITIE.ID\_COMPETITIE)

INTO V\_NR\_SP\_NU\_LOT

FROM COMPETITIE

FULL OUTER JOIN PARTICIPA ON COMPETITIE.ID\_COMPETITIE= PARTICIPA.ID\_COMPETITIE

JOIN SPORTIV ON PARTICIPA.ID\_SPORTIV= SPORTIV.ID\_SPORTIV

WHERE SPORTIV.TIP\_SPORTIV <> 'SPORTIV LOT' AND PARTICIPA.ID\_SPORTIV IS NOT NULL

AND competitie.id\_competitie= V\_ID\_COMP;

RETURN (V\_NR\_SP\_NU\_LOT\*100)/v\_nr\_total\_sp;

ELSE

RAISE NU\_PARTICIPANTI;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000,'Nu exista nicio competite cu numele dat.');

WHEN NU\_PARTICIPANTI THEN

RAISE\_APPLICATION\_ERROR(-20001,'Aceasta competite nu are participanti');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002,'Alta eroare');

END;

--ex 9

procedure sportiv\_competitii\_sponsor(v\_nume\_sp SPORTIV.NUME%TYPE,v\_pren\_sp SPORTIV.PRENUME%TYPE)

is

v\_id SPORTIV.ID\_SPORTIV%TYPE;

TYPE tip\_record\_cpr IS RECORD

(nume\_comp COMPETITIE.NUME%TYPE,

nume\_proba PROBA.NUME%TYPE,

loc\_clasam REZULTAT.LOC\_CLASAMENT%TYPE);

type tablou\_indexat\_cpr is table of tip\_record\_cpr index by pls\_integer;

t\_indexat\_cpr tablou\_indexat\_cpr;

--pentru partea a doua a exercitiului

TYPE tip\_record\_spons IS RECORD

(spons\_inc SPONSORIZEAZA.INCEPUT\_COLABORARE%TYPE,

spons\_sf SPONSORIZEAZA.SFARSIT\_COLABORARE%TYPE,

spons\_nume SPONSOR.NUME%TYPE);

TYPE tip\_record\_comp IS RECORD

(medalie PARTICIPA.MEDALIAT%TYPE,

comp\_inc COMPETITIE.DATA\_START%TYPE,

comp\_sf COMPETITIE.DATA\_FINAL%TYPE);

type tablou\_indexat is table of tip\_record\_spons index by pls\_integer;

t\_indexat\_sponsor tablou\_indexat;

type tablou\_indexat\_comp is TABLE OF tip\_record\_comp index by pls\_integer;

t\_indexat\_comp tablou\_indexat\_comp;

nr\_medaliati number;

numar\_comp\_int number; --numarul de competitii la care a participat sportivul in intervalul in care a fost sponsorizat

BEGIN

SELECT ID\_SPORTIV

INTO v\_id

FROM SPORTIV

WHERE NUME = v\_nume\_sp AND PRENUME = v\_pren\_sp;

SELECT competitie.nume,proba.nume,rezultat.loc\_clasament

BULK COLLECT INTO t\_indexat\_cpr

FROM SPORTIV

--SUNT LUATI IN CONSIDERARE DOAR SPORTIVII CARE PARTICIPA LA COMPETITIILE DIN ROMANIA

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

JOIN ORAS ON COMPETITIE.ID\_ORAS = ORAS.ID\_ORAS

JOIN TARA ON ORAS.ID\_TARA = TARA.ID\_TARA

join istoric\_sportiv on sportiv.id\_sportiv = istoric\_sportiv.id\_sportiv --mai intai datele sunt trecute in istoric si dupa in rezultat

JOIN PROBA ON PARTICIPA.ID\_PROBA = PROBA.ID\_PROBA

JOIN REZULTAT ON PROBA.ID\_PROBA = REZULTAT.ID\_PROBA

AND SPORTIV.ID\_SPORTIV = REZULTAT.ID\_SPORTIV

and istoric\_sportiv.id\_rezultat= rezultat.id\_rezultat

WHERE EXTRACT(MONTH FROM COMPETITIE.DATA\_START) IN (3,4,5,9,10,11)

AND TARA.NUME\_TARA = 'ROMANIA'

AND (PROBA.NUME LIKE '100%' OR PROBA.NUME LIKE '50%')

and SPORTIV.ID\_SPORTIV=v\_id;

IF t\_indexat\_cpr.COUNT <>0 THEN

for i in t\_indexat\_cpr.first .. t\_indexat\_cpr.last

loop

dbms\_output.put\_line(t\_indexat\_cpr(i).nume\_comp||' proba: '||t\_indexat\_cpr(i).nume\_proba||' locul: '||t\_indexat\_cpr(i).loc\_clasam);

end loop;

ELSE

DBMS\_OUTPUT.PUT\_LINE('ACEST SPORTIV NU PARTICIPA LA PROBE SCURTE DIN CADRUL COMPETITIILOR DIN ROMANIA');

END IF;

--partea a doua a exercitiului

select sponsorizeaza.inceput\_colaborare,sponsorizeaza.sfarsit\_colaborare,SPONSOR.NUME

BULK COLLECT into t\_indexat\_sponsor

from sportiv

join sponsorizeaza on sportiv.id\_sportiv = sponsorizeaza.id\_sportiv

join sponsor on sponsorizeaza.id\_sponsor = sponsor.id\_sponsor

where sportiv.id\_sportiv =v\_id;

SELECT PARTICIPA.MEDALIAT, COMPETITIE.DATA\_START,COMPETITIE.DATA\_FINAL

BULK COLLECT INTO t\_indexat\_comp

FROM SPORTIV

JOIN PARTICIPA ON SPORTIV.ID\_SPORTIV = PARTICIPA.ID\_SPORTIV

JOIN COMPETITIE ON PARTICIPA.ID\_COMPETITIE = COMPETITIE.ID\_COMPETITIE

where sportiv.id\_sportiv =v\_id;

IF t\_indexat\_sponsor.COUNT <>0 THEN

for i in t\_indexat\_sponsor.FIRST..t\_indexat\_sponsor.LAST

loop --aici intru pe un interval de sponsorizare

IF t\_indexat\_comp.COUNT <>0 THEN

nr\_medaliati:=0;

numar\_comp\_int:=0;

FOR j IN t\_indexat\_comp.FIRST..t\_indexat\_comp.LAST

loop

if t\_indexat\_comp(j).comp\_inc >= t\_indexat\_sponsor(i).spons\_inc

and t\_indexat\_comp(j).comp\_sf <= t\_indexat\_sponsor(i).spons\_sf

then

numar\_comp\_int:=numar\_comp\_int+1; --numarul de competitii la care a participat sportivul in intervalul in care a fost sponsorizat

end if;

if t\_indexat\_comp(j).comp\_inc >= t\_indexat\_sponsor(i).spons\_inc

and t\_indexat\_comp(j).comp\_sf <= t\_indexat\_sponsor(i).spons\_sf

and t\_indexat\_comp(j).medalie =1

then

nr\_medaliati:=nr\_medaliati+1;

end if;

end loop;

if numar\_comp\_int = 0 then

DBMS\_OUTPUT.PUT\_LINE('In intervalul de sponsorizare: '||t\_indexat\_sponsor(i).spons\_inc||' '||t\_indexat\_sponsor(i).spons\_sf||

' cu sponsorul: '||t\_indexat\_sponsor(i).spons\_nume||' sportivul nu a participat la nicio competitie');

elsif nr\_medaliati = 0 then

DBMS\_OUTPUT.PUT\_LINE('In intervalul de sponsorizare: '||t\_indexat\_sponsor(i).spons\_inc||' '||t\_indexat\_sponsor(i).spons\_sf||

' cu sponsorul: '||t\_indexat\_sponsor(i).spons\_nume||' sportivul nu a reusit sa obtina nicio medalie din '||numar\_comp\_int||' competitii la care a participat');

end if;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Sportivul nu a participat la nicio competitie pana acum');

END IF;

end loop;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Sportivul nu a fost si nici nu este sponsorizat');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20000,'NU EXISTA SPORTIVUL');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001,' SUNT MAI MULTI SPORTIVI CU ACELASI NUME');

WHEN OTHERS THEN

RAISE\_APPLICATION\_ERROR(-20002,'ALTA EROARE');

END;

END pachet\_proiect;

/

--TESTAM PACHETUL

--EX 6

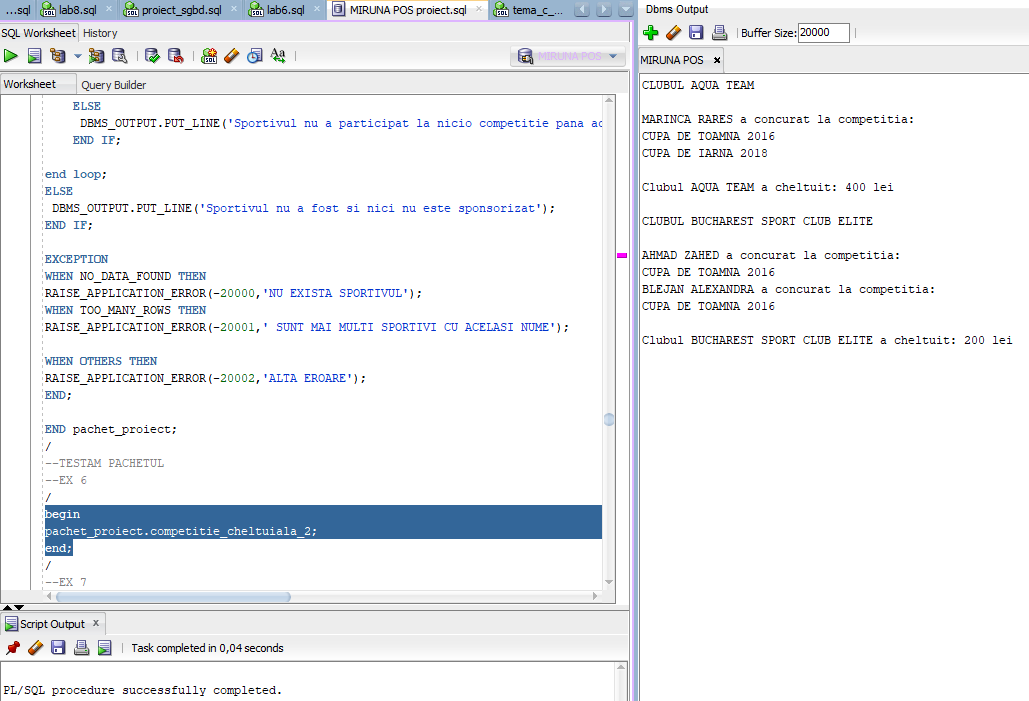
/

begin

pachet\_proiect.competitie\_cheltuiala\_2;

end;

/



--EX 7

/

begin

pachet\_proiect.club\_antr\_sp;

end;

/

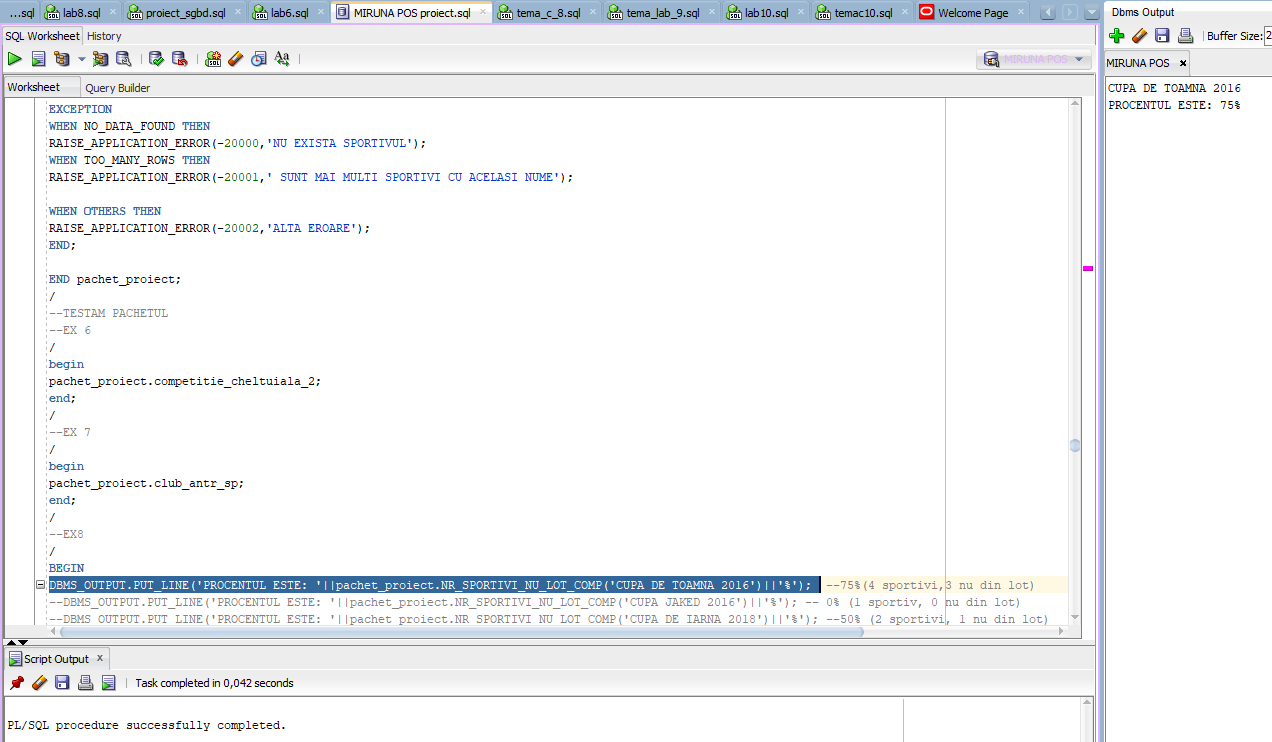


--EX8

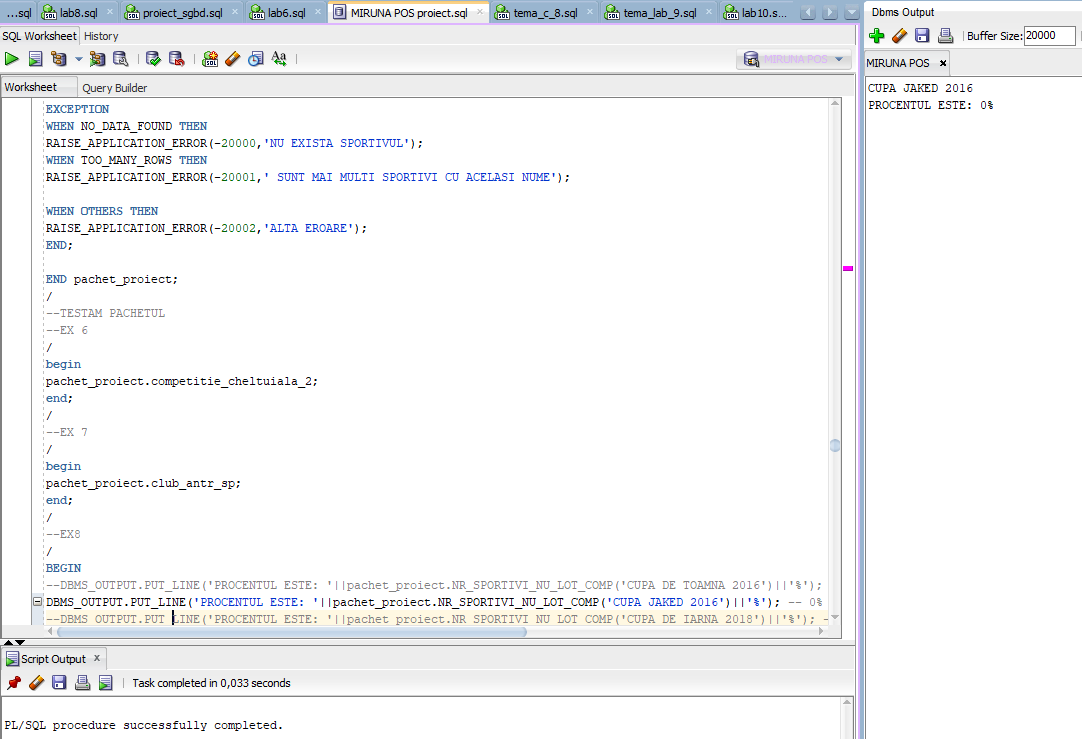
/

BEGIN

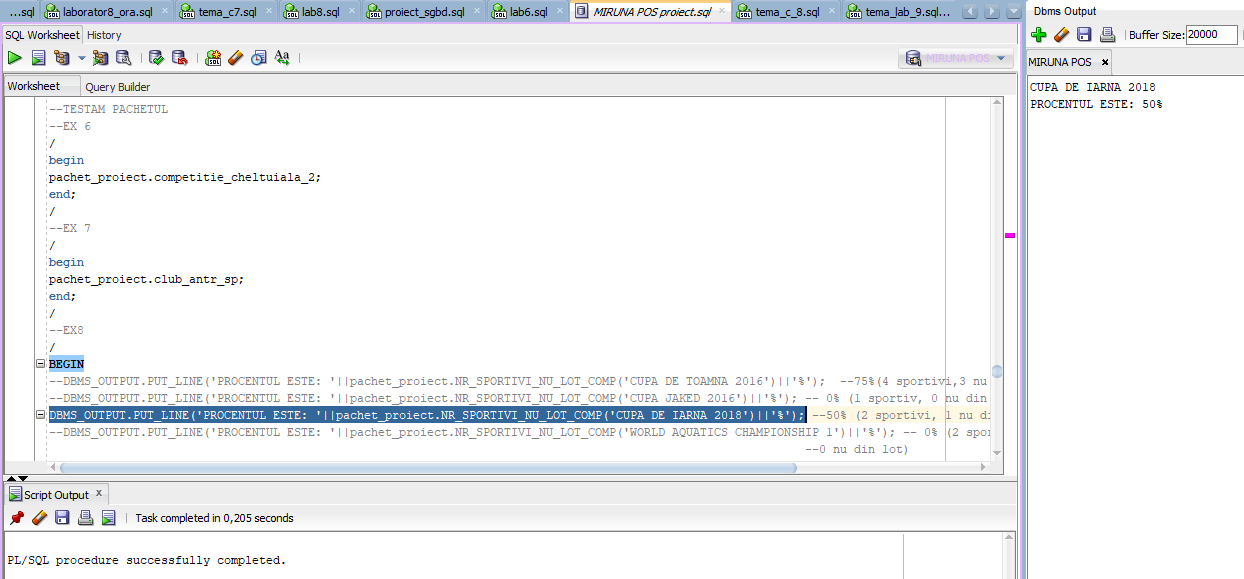
--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA DE TOAMNA 2016')||'%'); --75%(4 sportivi,3 nu din lot)



--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA JAKED 2016')||'%'); -- 0% (1 sportiv, 0 nu din lot)

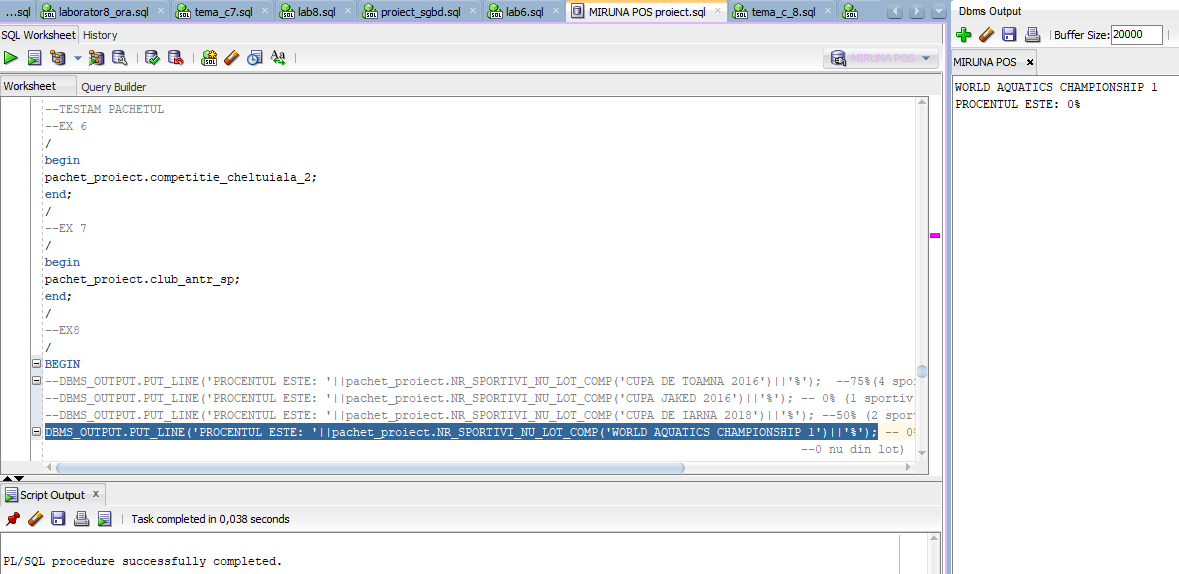


--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('CUPA DE IARNA 2018')||'%'); --50% (2 sportivi, 1 nu din lot)

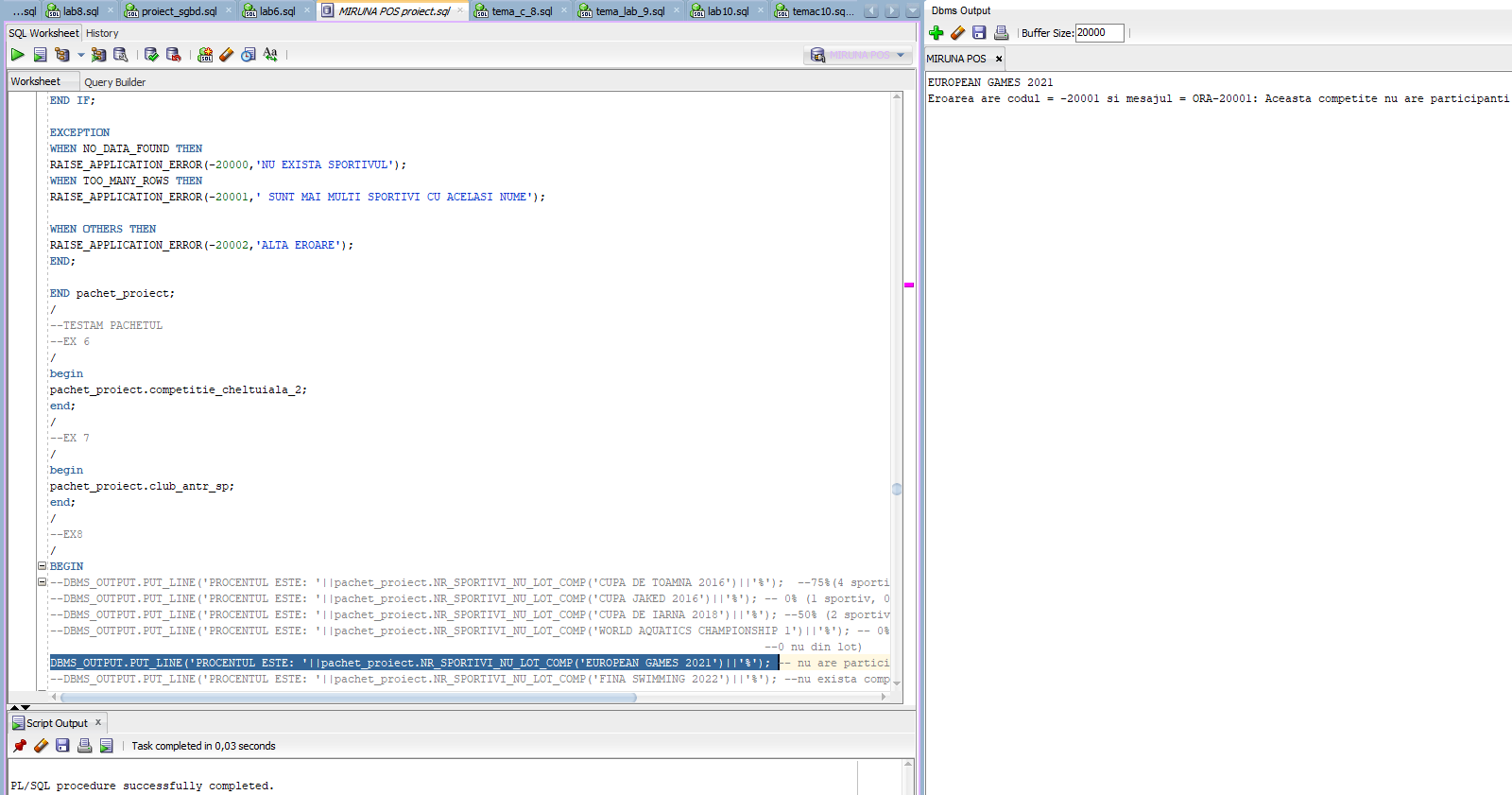


--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('WORLD AQUATICS CHAMPIONSHIP 1')||'%'); -- 0% (2 sportivi,

--0 nu din lot)

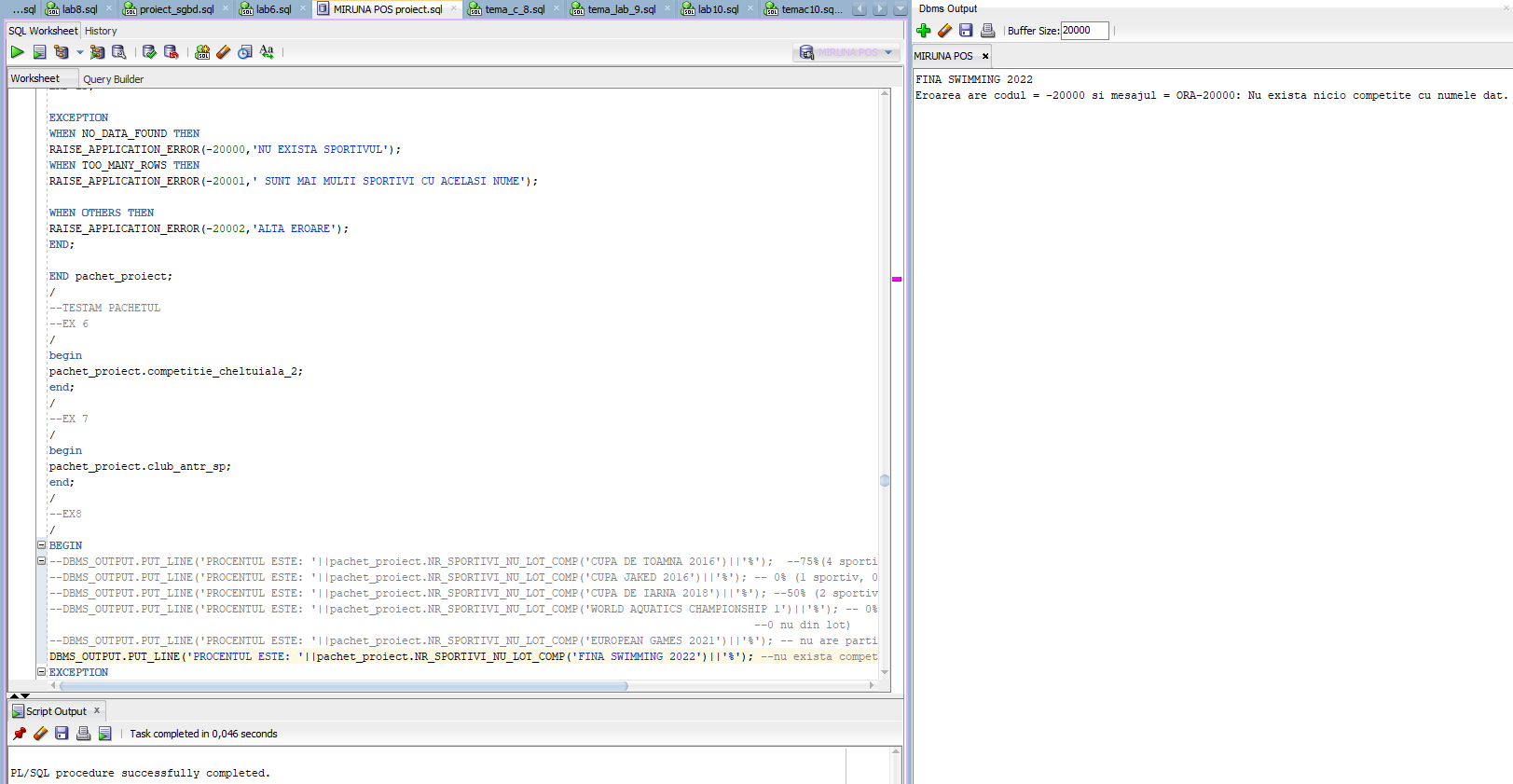


--DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('EUROPEAN GAMES 2021')||'%'); -- nu are participanti



DBMS\_OUTPUT.PUT\_LINE('PROCENTUL ESTE: '||pachet\_proiect.NR\_SPORTIVI\_NU\_LOT\_COMP('FINA SWIMMING 2022')||'%');

--nu exista competitia



EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

END;

/

--EX 9 (PENTRU A DA EROAREA DE TOO MANY ROWS SE VA INSERA ACEST SPORTIV, IAR DUPA EXECUTAREA PROCEDURII SE DA ROLLBACK)

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (13,'F','BELLU','SARA',TO\_DATE('16-10-2001','dd-mm-yyyy'),1,'SPORTIV LOT',301,50);

ROLLBACK;

/

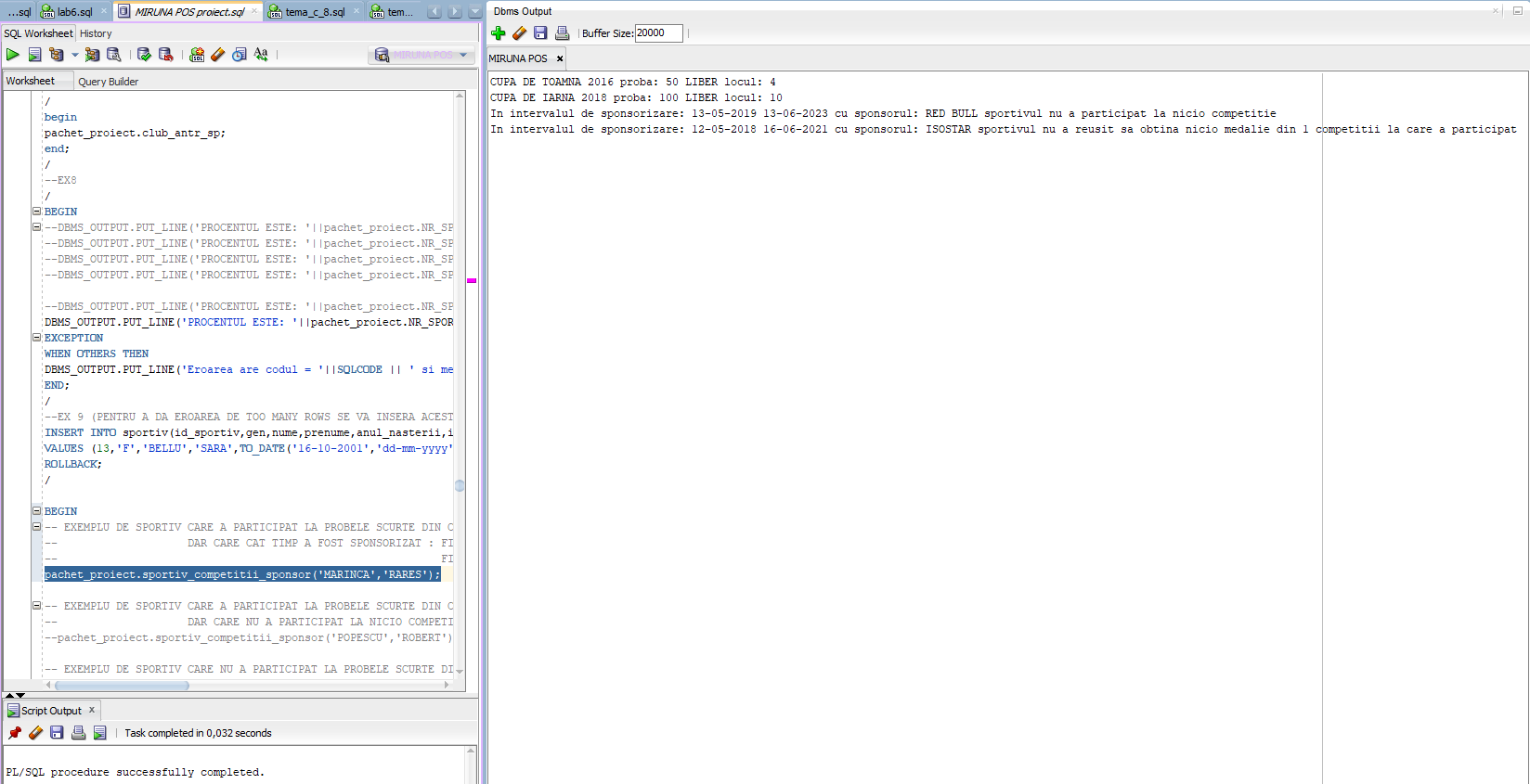
BEGIN

-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE CAT TIMP A FOST SPONSORIZAT : FIE NU A PARTICIPAT LA NICIO COMPETITIE

-- FIE NU A REUSIT SA OBTINA NICIO MEDALIE

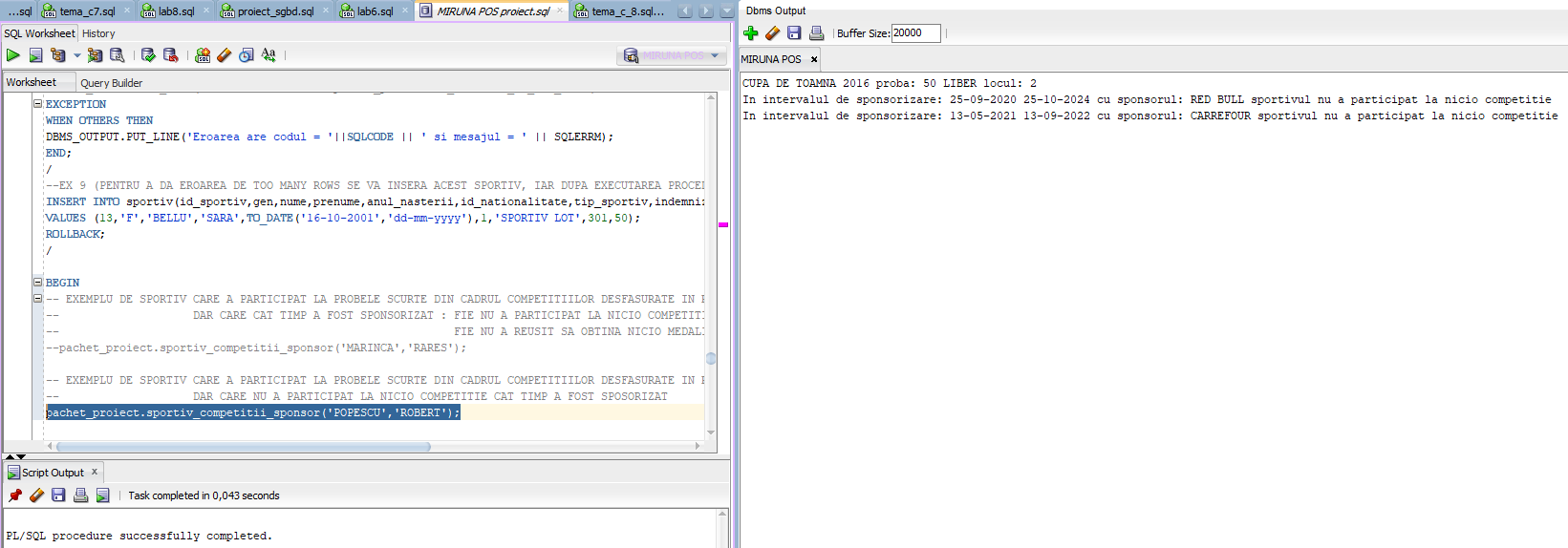
pachet\_proiect.sportiv\_competitii\_sponsor('MARINCA','RARES');



-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE NU A PARTICIPAT LA NICIO COMPETITIE CAT TIMP A FOST SPOSORIZAT

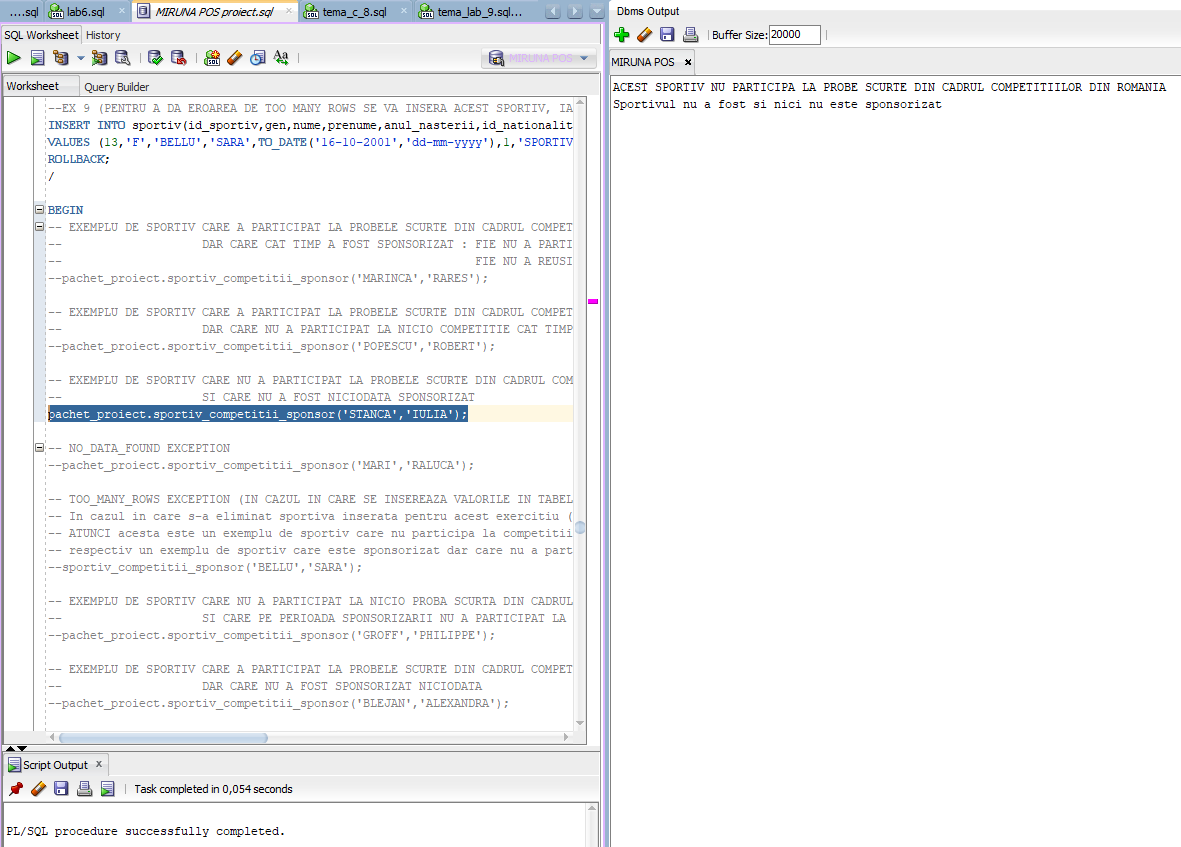
--pachet\_proiect.sportiv\_competitii\_sponsor('POPESCU','ROBERT');



-- EXEMPLU DE SPORTIV CARE NU A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

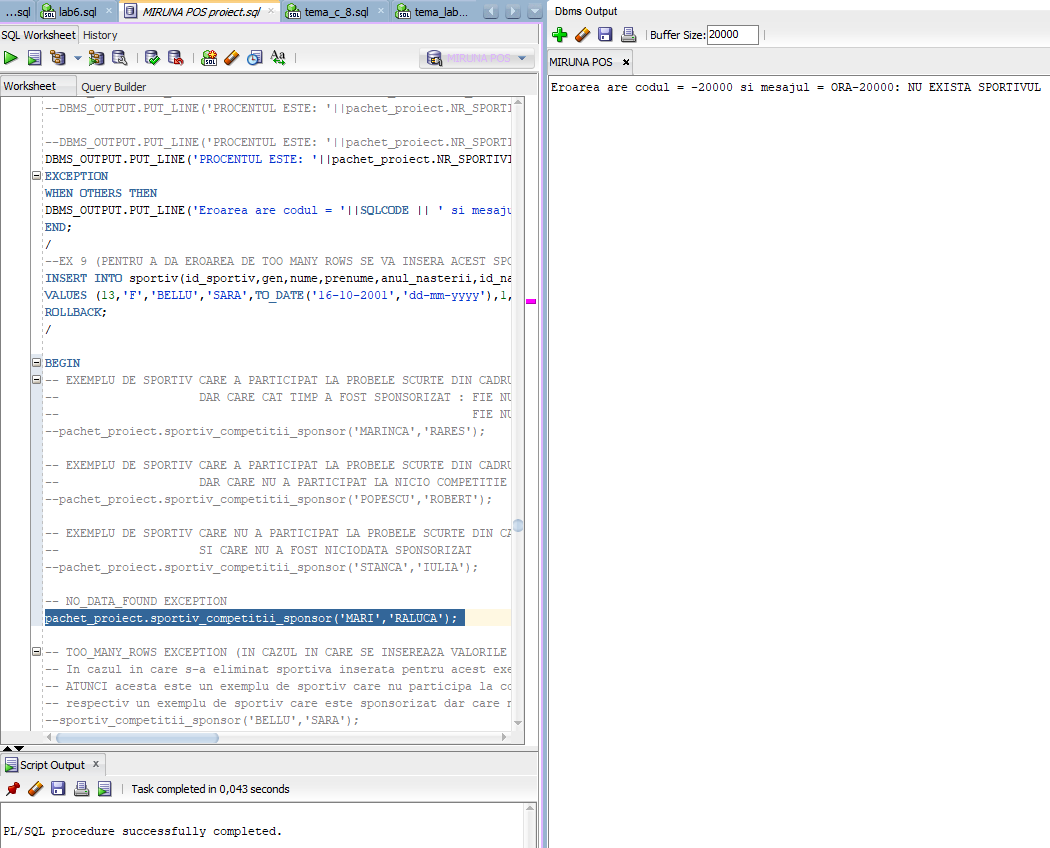
-- SI CARE NU A FOST NICIODATA SPONSORIZAT

--pachet\_proiect.sportiv\_competitii\_sponsor('STANCA','IULIA');

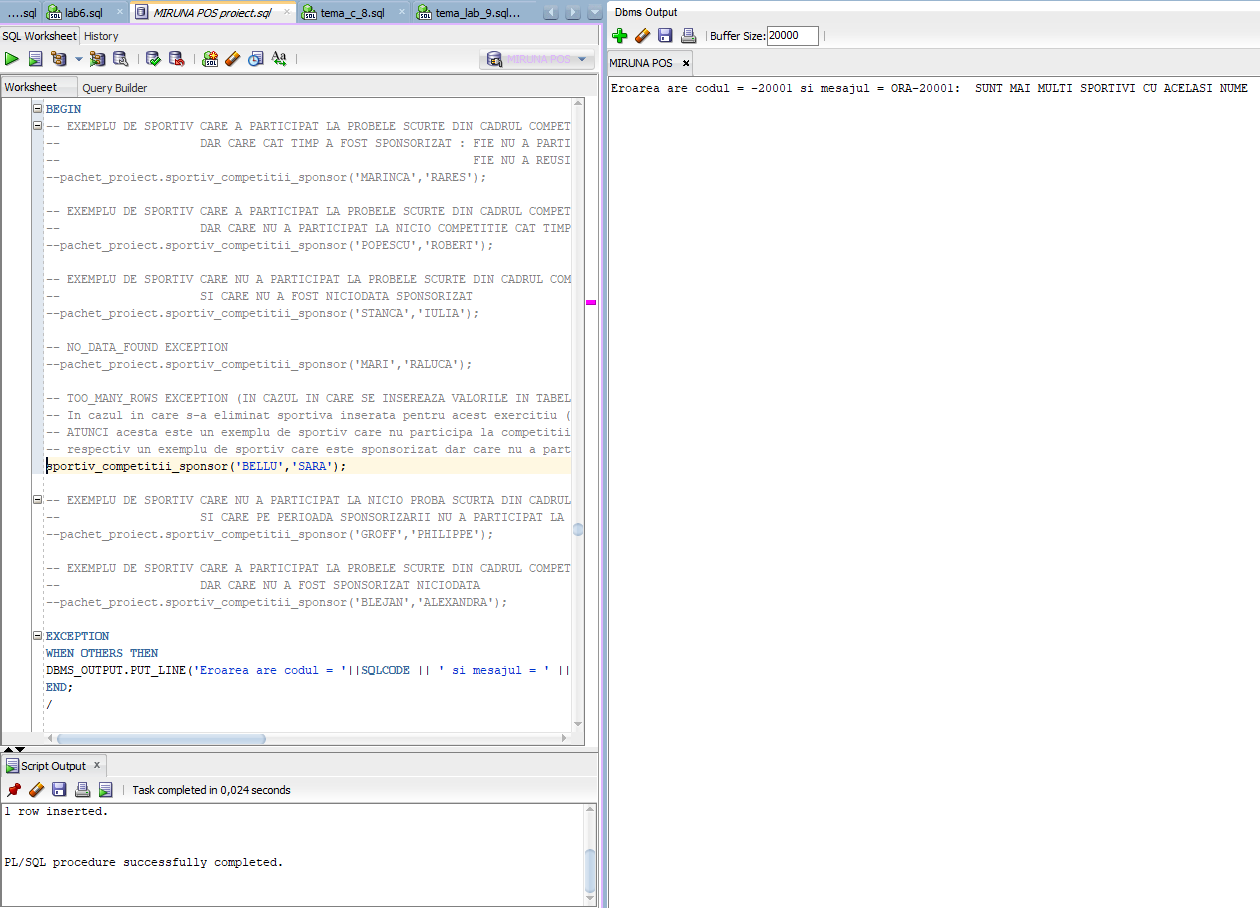


-- NO\_DATA\_FOUND EXCEPTION

--pachet\_proiect.sportiv\_competitii\_sponsor('MARI','RALUCA');



-- TOO\_MANY\_ROWS EXCEPTION (IN CAZUL IN CARE SE INSEREAZA VALORILE IN TABEL)

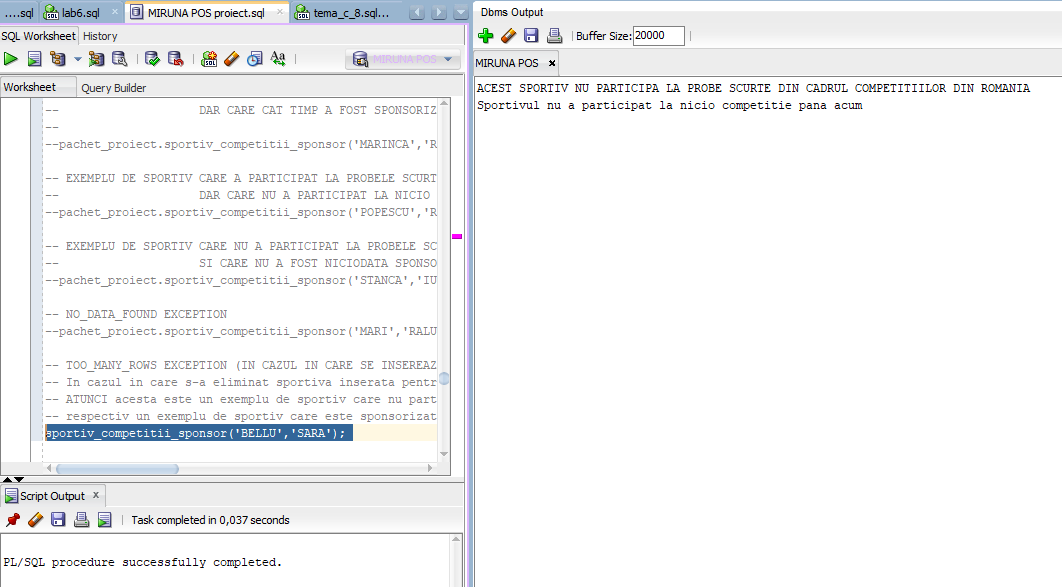


-- In cazul in care s-a eliminat sportiva inserata pentru acest exercitiu (ramanand doar o sportiva cu numele BELLU SARA)

-- ATUNCI acesta este un exemplu de sportiv care nu participa la competitii in Romania,

-- respectiv un exemplu de sportiv care este sponsorizat dar care nu a participat niciodata la o competitie

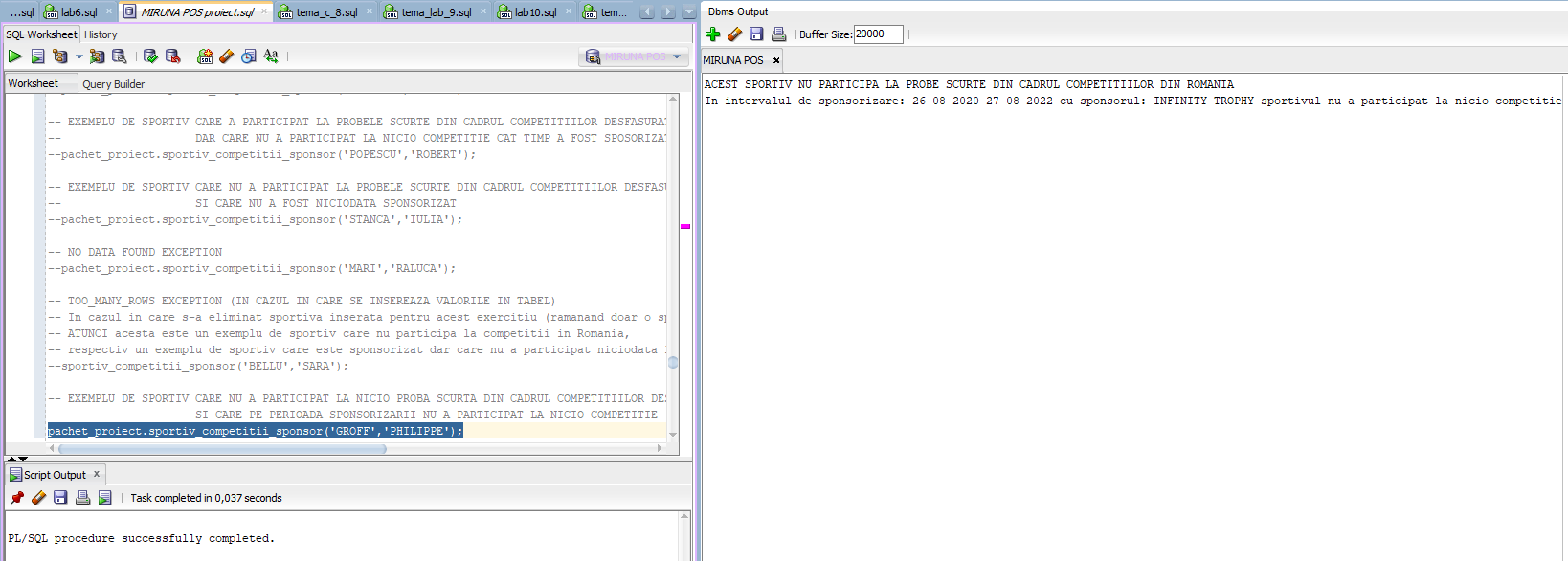
--sportiv\_competitii\_sponsor('BELLU','SARA');



-- EXEMPLU DE SPORTIV CARE NU A PARTICIPAT LA NICIO PROBA SCURTA DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- SI CARE PE PERIOADA SPONSORIZARII NU A PARTICIPAT LA NICIO COMPETITIE

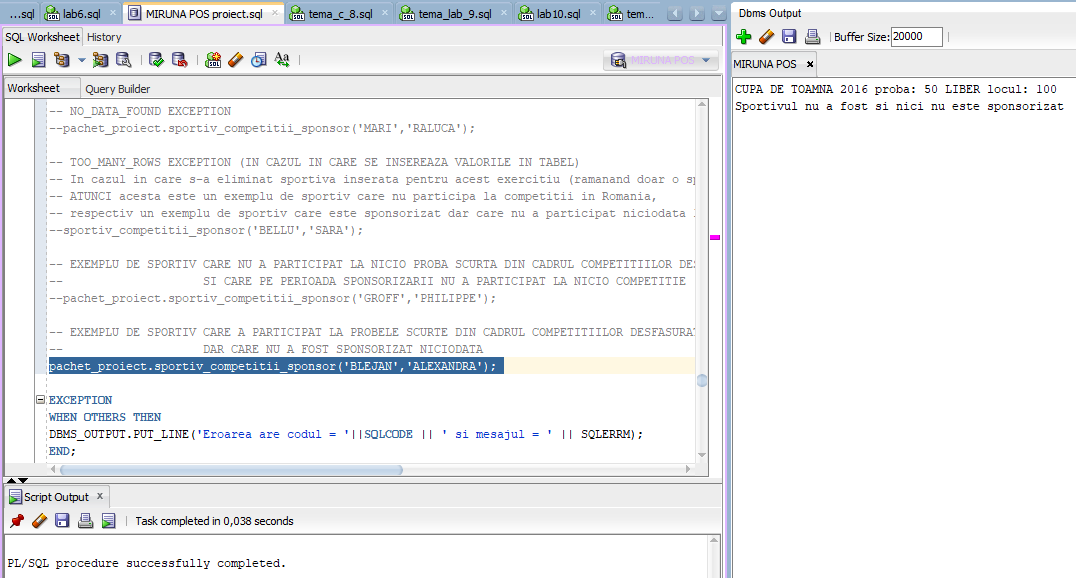
--pachet\_proiect.sportiv\_competitii\_sponsor('GROFF','PHILIPPE');



-- EXEMPLU DE SPORTIV CARE A PARTICIPAT LA PROBELE SCURTE DIN CADRUL COMPETITIILOR DESFASURATE IN ROMANIA

-- DAR CARE NU A FOST SPONSORIZAT NICIODATA

--pachet\_proiect.sportiv\_competitii\_sponsor('BLEJAN','ALEXANDRA');



EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

END;

/

# 14.

/\*

Pentru un sportiv al carui nume este dat ca parametru se doreste sa fie inscris la o proba al carei nume este dat ca

parametru din cadrul unei competitii al carei nume este dat ca parametru.

Daca in trecut (ADICA LA COMPETITIILE CARE AU FOST INAINTEA ACESTEI COMPETITII CURENTE) s-a clasat

cel putin o data in primii 15 la aceasta proba sau daca este prima sa participare la o astfel de proba

si daca nu a fost inscris deja la aceasta proba (in competitia data) se vor insera datele corespunzatoare.

(initial sportivul va fi trecut ca nemedaliat, ulterior dupa terminarea probei se va cunoaste daca va fi medaliat)

In cazul in care a fost deja inscris sa se afiseze locurile,timpii obtinute in trecut la aceasta proba, dar si data cand a

participat la proba.

In cazul in care sportivul a fost deja inscris si este pentru prima data cand participa la acea proba, atunci se va afisa

un mesaj din care sa reiasa ca nu are un istoric privind evolutia sa in cadrul acelei probe.

A se stii ca in cazul in care sportivul nu a reusit sa se claseze in primii 15 in urma participarii la aceasta proba,

antrenorul poate decide sa fie inscris manual la concurs la acea proba. (aceste inserari se vor face la cerere si nu vor fi

tratate in pachet).

\*/

/

CREATE OR REPLACE PACKAGE pachet\_participari\_comp\_pr

IS

function verif\_participare(nume\_sportiv SPORTIV.NUME%TYPE, --VERIFICAM DACA A MAI PARTICIPAT LA O ASTFEL DE

prenume\_sportiv SPORTIV.PRENUME%TYPE, --PROBA IN TRECUT

nume\_proba PROBA.NUME%TYPE,

nume\_competitie COMPETITIE.NUME%TYPE)

RETURN BOOLEAN;

FUNCTION in\_primii\_15(nume\_sportiv\_2 SPORTIV.NUME%TYPE,

prenume\_sportiv\_2 SPORTIV.PRENUME%TYPE,

nume\_proba\_2 PROBA.NUME%TYPE,

nume\_competitie\_2 COMPETITIE.NUME%TYPE)

RETURN NUMBER;

FUNCTION verif\_deja\_inserat(sp\_nume SPORTIV.NUME%TYPE,

sp\_prenune SPORTIV.PRENUME%TYPE,

pr\_nume PROBA.NUME%TYPE,

comp\_nume COMPETITIE.NUME%TYPE)

RETURN BOOLEAN;

PROCEDURE loc\_timp\_comp (nume\_sp SPORTIV.NUME%TYPE,

prenume\_sp SPORTIV.PRENUME%TYPE,

nume\_pr PROBA.NUME%TYPE

);

PROCEDURE inserare\_afisare\_sp\_pr\_comp(spo\_nume SPORTIV.NUME%TYPE,

spo\_prenume SPORTIV.PRENUME%TYPE,

pro\_nume PROBA.NUME%TYPE,

compe\_nume COMPETITIE.NUME%TYPE);

END pachet\_participari\_comp\_pr;

/

CREATE OR REPLACE PACKAGE BODY pachet\_participari\_comp\_pr

IS

function verif\_participare(nume\_sportiv SPORTIV.NUME%TYPE, --VERIFICAM DACA A MAI PARTICIPAT LA O ASTFEL DE

prenume\_sportiv SPORTIV.PRENUME%TYPE, --PROBA IN TRECUT

nume\_proba PROBA.NUME%TYPE,

nume\_competitie COMPETITIE.NUME%TYPE)

RETURN BOOLEAN IS

TYPE tablou\_indexat\_1 IS TABLE OF SPORTIV.ID\_SPORTIV%TYPE INDEX BY PLS\_INTEGER;

tablou\_indexat\_sportivi tablou\_indexat\_1;

id\_sp SPORTIV.ID\_SPORTIV%TYPE;

id\_pr PROBA.ID\_PROBA%TYPE;

id\_comp COMPETITIE.ID\_COMPETITIE%TYPE;

ok NUMBER;

nr\_partic\_la\_proba\_in\_trecut NUMBER;

BEGIN

SELECT ID\_SPORTIV

INTO id\_sp

FROM SPORTIV

WHERE NUME = nume\_sportiv AND PRENUME = prenume\_sportiv; --DACA NU EXISTA SPORTIVUL DAT => INTOARCE NO\_DATA\_FOUND

--DACA SUNT MAI MULTI SPORTIVI CU ACELASI NUME => TOO\_MANY\_ROWS

SELECT ID\_PROBA

INTO id\_pr

FROM PROBA

WHERE NUME = nume\_proba;--DACA NU EXISTA PROBA => NO\_DATA\_FOUND

SELECT ID\_COMPETITIE

INTO id\_comp

FROM COMPETITIE

WHERE NUME = nume\_competitie;

--CAZUL IN CARE SPORTIVUL NU A MAI PARTICIPAT LA NICIO COMPETITE

SELECT DISTINCT(ID\_SPORTIV)

BULK COLLECT INTO tablou\_indexat\_sportivi

FROM PARTICIPA;

ok:=0;

IF tablou\_indexat\_sportivi.COUNT <> 0 THEN

FOR I IN tablou\_indexat\_sportivi.FIRST..tablou\_indexat\_sportivi.LAST

LOOP

IF tablou\_indexat\_sportivi(I) = id\_sp THEN

ok:=1;--sportivul a mai participat la competitii

END IF;

END LOOP;

END IF;

IF ok=0 THEN

RETURN FALSE; --DACA SPORTIVUL NU A PARTICIPAT LA NICIO COMPETITIE PANA ACUM ESTE CLAR CA NU A MAI PARTICIPAT LA ACEA PROBA

END IF;

--Acum se verifica daca sportivul a mai participat in trecut la aceasta proba

select count(\*)

INTO nr\_partic\_la\_proba\_in\_trecut

from sportiv

right join participa on sportiv.id\_sportiv = participa.id\_sportiv

left join proba on proba.id\_proba = participa.id\_proba

left join competitie on competitie.id\_competitie = participa.id\_competitie

where sportiv.nume= nume\_sportiv AND sportiv.prenume=prenume\_sportiv AND competitie.nume!=nume\_competitie

AND proba.nume = nume\_proba;

IF nr\_partic\_la\_proba\_in\_trecut = 0 THEN

RETURN FALSE;

ELSE

RETURN TRUE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20044,'NU EXISTA CEEA CE CAUTATI');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20045,'EXISTA MAI MULTI SPORTIVI CU ACEST NUME');

END;

FUNCTION in\_primii\_15(nume\_sportiv\_2 SPORTIV.NUME%TYPE,

prenume\_sportiv\_2 SPORTIV.PRENUME%TYPE,

nume\_proba\_2 PROBA.NUME%TYPE,

nume\_competitie\_2 COMPETITIE.NUME%TYPE)

RETURN NUMBER IS

id\_sp\_2 SPORTIV.ID\_SPORTIV%TYPE;

id\_pr\_2 PROBA.ID\_PROBA%TYPE;

id\_comp\_2 COMPETITIE.ID\_COMPETITIE%TYPE;

loc\_pr\_15 NUMBER:=-1;

BEGIN

IF verif\_participare(nume\_sportiv\_2,prenume\_sportiv\_2,nume\_proba\_2,nume\_competitie\_2) THEN --DACA A MAI PARTICIPAT LA PROBA

SELECT ID\_SPORTIV

INTO id\_sp\_2

FROM SPORTIV

WHERE NUME = nume\_sportiv\_2 AND PRENUME = prenume\_sportiv\_2; --DACA NU EXISTA SPORTIVUL DAT => INTOARCE NO\_DATA\_FOUND

--DACA SUNT MAI MULTI SPORTIVI CU ACELASI NUME => TOO\_MANY\_ROWS

SELECT ID\_PROBA

INTO id\_pr\_2

FROM PROBA

WHERE NUME = nume\_proba\_2;--DACA NU EXISTA PROBA => NO\_DATA\_FOUND

SELECT ID\_COMPETITIE

INTO id\_comp\_2

FROM COMPETITIE

WHERE NUME = nume\_competitie\_2;

SELECT COUNT(LOC\_CLASAMENT)

INTO loc\_pr\_15

FROM REZULTAT

WHERE ID\_PROBA = id\_pr\_2 AND ID\_SPORTIV = id\_sp\_2 AND LOC\_CLASAMENT < = 15;

END IF;

RETURN loc\_pr\_15;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20044,'NU EXISTA CEEA CE CAUTATI');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20045,'EXISTA MAI MULTI SPORTIVI CU ACEST NUME');

END;

FUNCTION verif\_deja\_inserat(sp\_nume SPORTIV.NUME%TYPE,

sp\_prenune SPORTIV.PRENUME%TYPE,

pr\_nume PROBA.NUME%TYPE,

comp\_nume COMPETITIE.NUME%TYPE)

RETURN BOOLEAN

IS

id\_sporti SPORTIV.ID\_SPORTIV%TYPE;

id\_prob PROBA.ID\_PROBA%TYPE;

id\_compet COMPETITIE.ID\_COMPETITIE%TYPE;

este\_inscris NUMBER;

BEGIN

SELECT ID\_SPORTIV

INTO id\_sporti

FROM SPORTIV

WHERE NUME = sp\_nume AND PRENUME = sp\_prenune; --DACA NU EXISTA SPORTIVUL DAT => INTOARCE NO\_DATA\_FOUND

--DACA SUNT MAI MULTI SPORTIVI CU ACELASI NUME => TOO\_MANY\_ROWS

SELECT ID\_PROBA

INTO id\_prob

FROM PROBA

WHERE NUME = pr\_nume;--DACA NU EXISTA PROBA => NO\_DATA\_FOUND

SELECT ID\_COMPETITIE

INTO id\_compet

FROM COMPETITIE

WHERE NUME = comp\_nume;

SELECT COUNT(\*)

INTO este\_inscris

FROM PARTICIPA

WHERE ID\_SPORTIV = id\_sporti AND ID\_COMPETITIE = id\_compet AND ID\_PROBA = id\_prob;

IF este\_inscris = 0 THEN

RETURN FALSE;

ELSE

RETURN TRUE;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20044,'NU EXISTA CEEA CE CAUTATI');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20045,'EXISTA MAI MULTI SPORTIVI CU ACEST NUME');

END;

PROCEDURE loc\_timp\_comp (nume\_sp SPORTIV.NUME%TYPE,

prenume\_sp SPORTIV.PRENUME%TYPE,

nume\_pr PROBA.NUME%TYPE

)

IS

idul\_sp SPORTIV.ID\_SPORTIV%TYPE;

idul\_pr PROBA.ID\_PROBA%TYPE;

TYPE tipul\_record IS RECORD

(loc\_obtinut REZULTAT.LOC\_CLASAMENT%TYPE,

timp REZULTAT.TIMP\_OBTINUT%TYPE,

data\_rez ISTORIC\_SPORTIV.DATA\_REZULTATE\_SPORTIV%TYPE

);

TYPE tablou\_imbricat IS TABLE OF tipul\_record;

tablou tablou\_imbricat := tablou\_imbricat();

BEGIN

SELECT ID\_SPORTIV

INTO idul\_sp

FROM SPORTIV

WHERE NUME = nume\_sp AND PRENUME = prenume\_sp; --DACA NU EXISTA SPORTIVUL DAT => INTOARCE NO\_DATA\_FOUND

--DACA SUNT MAI MULTI SPORTIVI CU ACELASI NUME => TOO\_MANY\_ROWS

SELECT ID\_PROBA

INTO idul\_pr

FROM PROBA

WHERE NUME = nume\_pr;--DACA NU EXISTA PROBA => NO\_DATA\_FOUND

SELECT LOC\_CLASAMENT,TIMP\_OBTINUT,DATA\_REZULTATE\_SPORTIV

BULK COLLECT INTO tablou

FROM REZULTAT

LEFT JOIN ISTORIC\_SPORTIV ON REZULTAT.ID\_REZULTAT = ISTORIC\_SPORTIV.ID\_REZULTAT

WHERE REZULTAT.ID\_PROBA = idul\_pr AND REZULTAT.ID\_SPORTIV = idul\_sp;

IF tablou.COUNT <> 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Sportivul '||nume\_sp||' '||prenume\_sp||' a mai participat la proba '||nume\_pr);

DBMS\_OUTPUT.PUT\_LINE('Avand rezultatele:');

DBMS\_OUTPUT.NEW\_LINE;

FOR i IN tablou.FIRST..tablou.LAST

LOOP

DBMS\_OUTPUT.PUT\_LINE('Locul:'||tablou(i).loc\_obtinut||', cu timpul:'||tablou(i).timp||'s, in data de:'

||tablou(i).data\_rez);

END LOOP;

ELSE DBMS\_OUTPUT.PUT\_LINE('Sportivul '||nume\_sp||' '||prenume\_sp||' nu a mai participat la proba '||nume\_pr||', fiind

introdus manual la aceasta proba');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20044,'NU EXISTA CEEA CE CAUTATI');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20045,'EXISTA MAI MULTI SPORTIVI CU ACEST NUME');

END;

PROCEDURE inserare\_afisare\_sp\_pr\_comp(spo\_nume SPORTIV.NUME%TYPE,

spo\_prenume SPORTIV.PRENUME%TYPE,

pro\_nume PROBA.NUME%TYPE,

compe\_nume COMPETITIE.NUME%TYPE)

IS

id\_spo SPORTIV.ID\_SPORTIV%TYPE;

id\_pro PROBA.ID\_PROBA%TYPE;

id\_compe COMPETITIE.ID\_COMPETITIE%TYPE;

BEGIN

SELECT ID\_SPORTIV

INTO id\_spo

FROM SPORTIV

WHERE NUME = spo\_nume AND PRENUME = spo\_prenume;

SELECT ID\_PROBA

INTO id\_pro

FROM PROBA

WHERE NUME = pro\_nume;

SELECT ID\_COMPETITIE

INTO id\_compe

FROM COMPETITIE

WHERE NUME = compe\_nume;

IF (in\_primii\_15(spo\_nume,spo\_prenume,pro\_nume,compe\_nume)>=1 OR verif\_participare(spo\_nume,spo\_prenume,pro\_nume,compe\_nume)= FALSE)

AND verif\_deja\_inserat(spo\_nume,spo\_prenume,pro\_nume,compe\_nume)= FALSE

THEN

INSERT INTO PARTICIPA VALUES(id\_spo,id\_compe,id\_pro,0);

--INITIAL SE VA TRECE CA NEMEDALIAT DEOARECE INCA NU A PARTICIPAT LA PROBA SI ULTERIOR DUPA TERMINAREA PROBEI SE VA

--ACTUALIZA DACA ESTE CAZUL

ELSIF verif\_deja\_inserat(spo\_nume,spo\_prenume,pro\_nume,compe\_nume)= TRUE THEN

loc\_timp\_comp(spo\_nume,spo\_prenume,pro\_nume);

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE\_APPLICATION\_ERROR(-20044,'NU EXISTA CEEA CE CAUTATI');

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20045,'EXISTA MAI MULTI SPORTIVI CU ACEST NUME');

END;

END pachet\_participari\_comp\_pr;

/

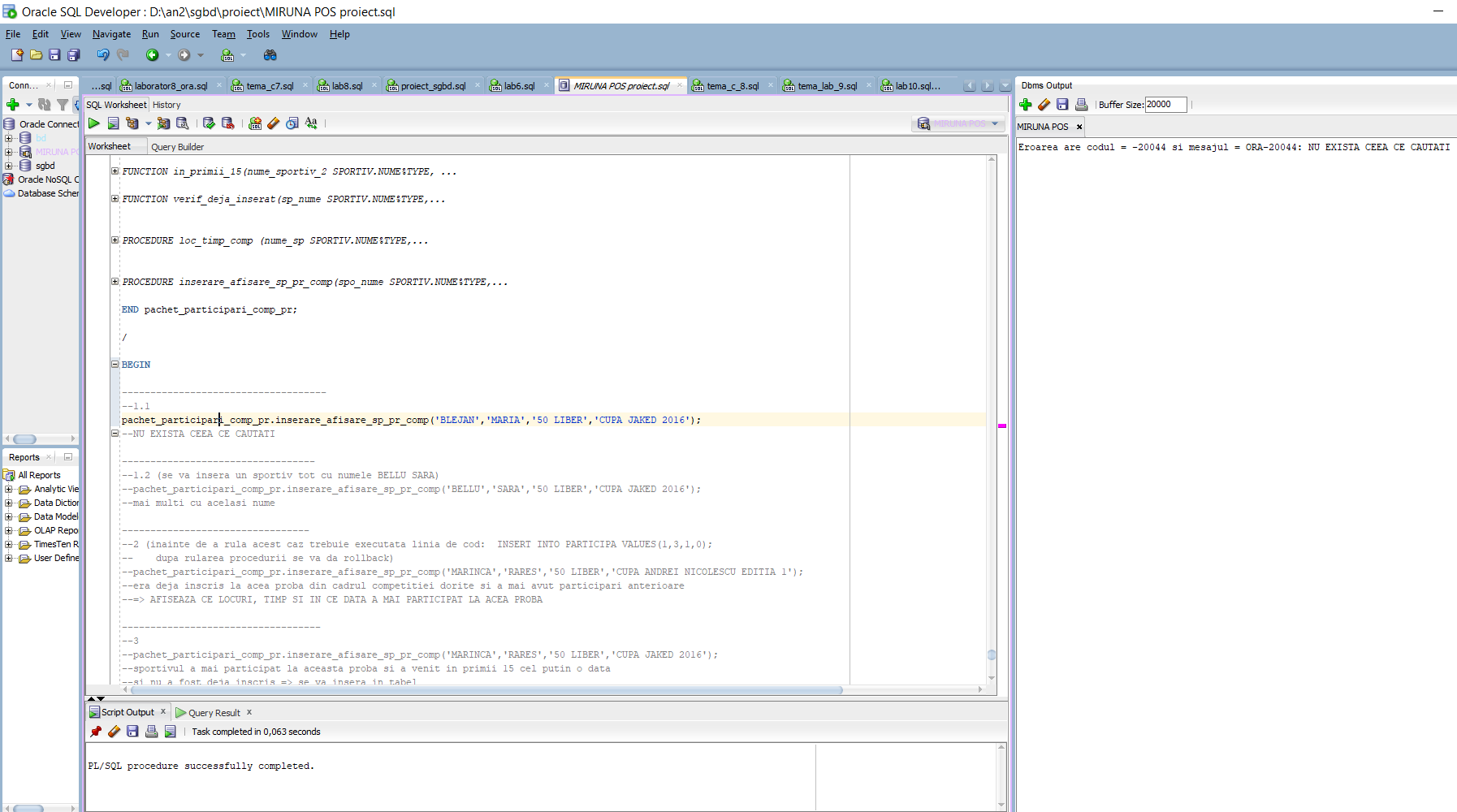
BEGIN

------------------------------------

--1.1

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('BLEJAN','MARIA','50 LIBER','CUPA JAKED 2016');

--NU EXISTA CEEA CE CAUTATI

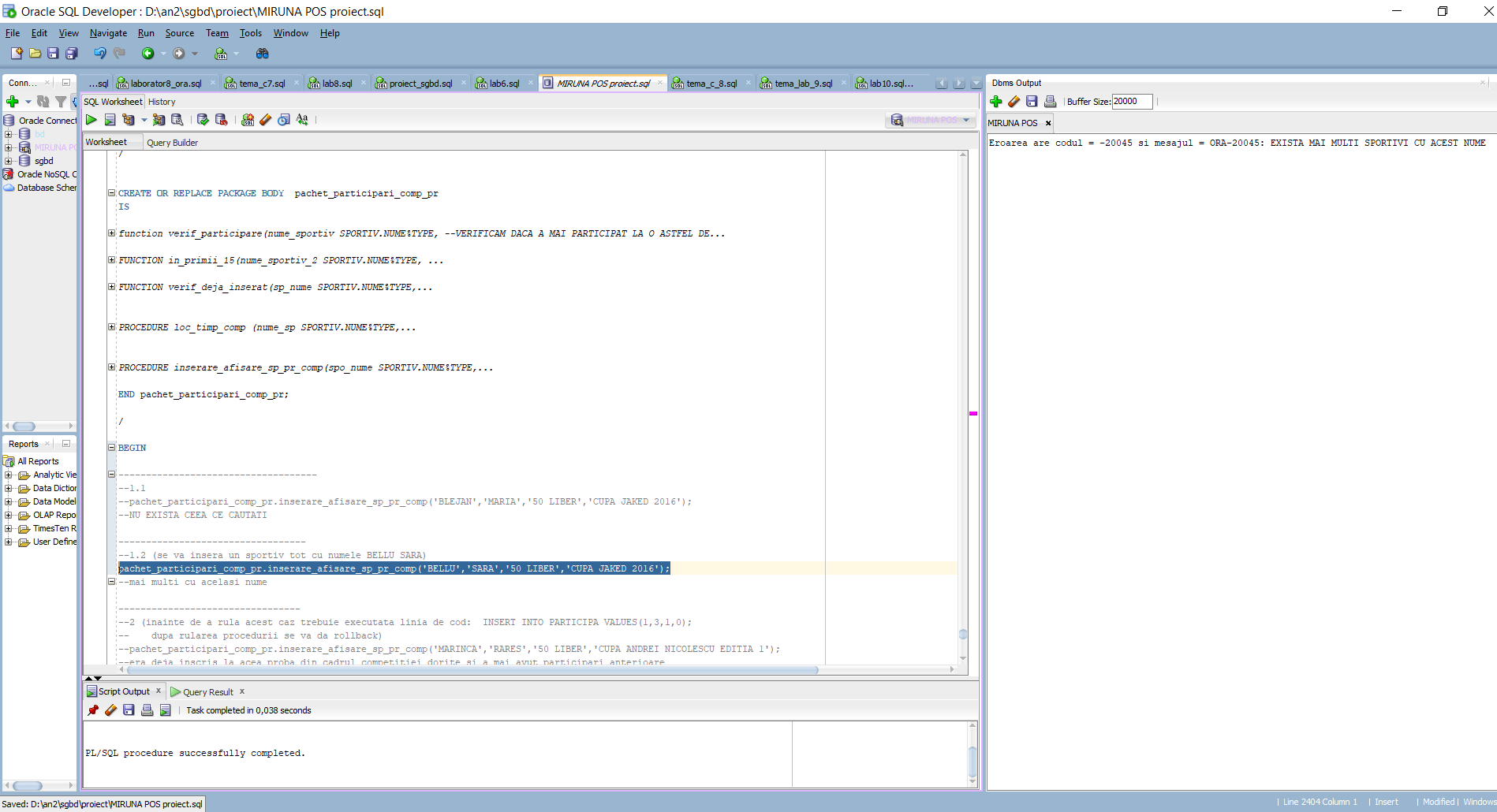


----------------------------------

--1.2 (se va insera un sportiv tot cu numele BELLU SARA)

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('BELLU','SARA','50 LIBER','CUPA JAKED 2016');

--mai multi cu acelasi nume



---------------------------------

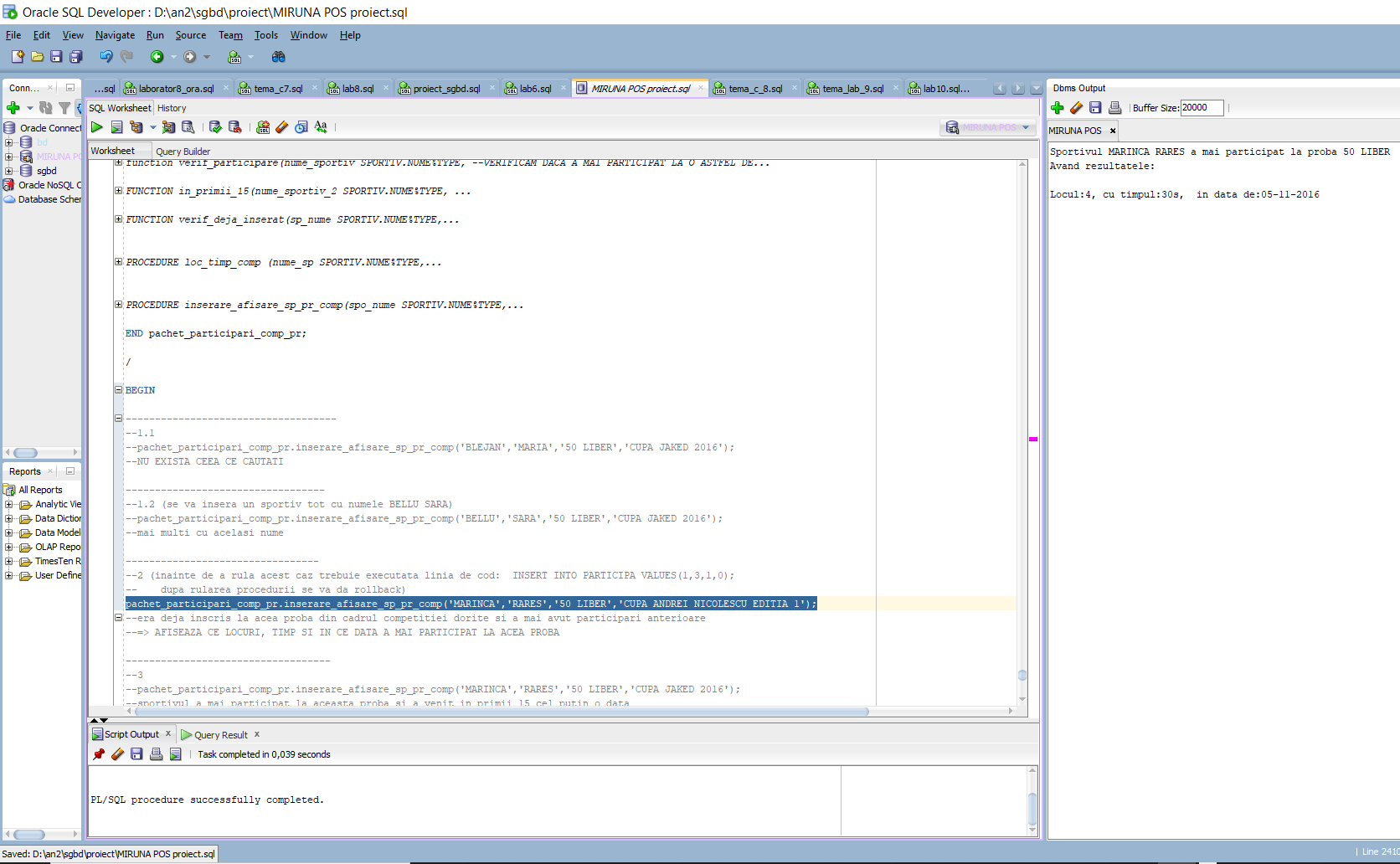
--2 (inainte de a rula acest caz trebuie executata linia de cod: INSERT INTO PARTICIPA VALUES(1,3,1,0);

-- dupa rularea procedurii se va da rollback)

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('MARINCA','RARES','50 LIBER','CUPA ANDREI NICOLESCU EDITIA 1');

--era deja inscris la acea proba din cadrul competitiei dorite si a mai avut participari anterioare

--=> AFISEAZA CE LOCURI, TIMP SI IN CE DATA A MAI PARTICIPAT LA ACEA PROBA



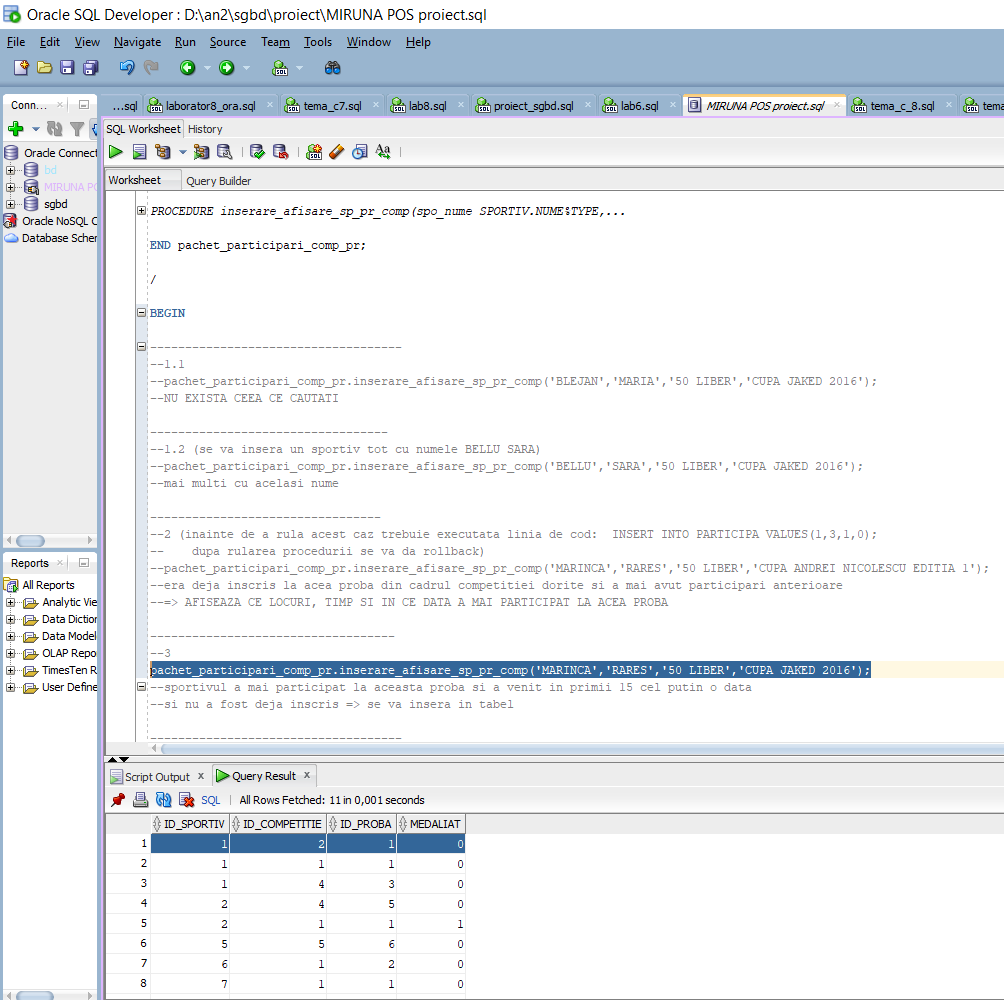
-----------------------------------

--3

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('MARINCA','RARES','50 LIBER','CUPA JAKED 2016');

--sportivul a mai participat la aceasta proba si a venit in primii 15 cel putin o data

--si nu a fost deja inscris => se va insera in tabel



------------------------------------

--4

pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('STANCA','IULIA','50 LIBER','FINA SWIMMING WORLD CUP 2021');

--acest sportiv nu a mai participat la aceasta proba pana acum (acest sportiv chiar nu a mai participat la nicio competitie)

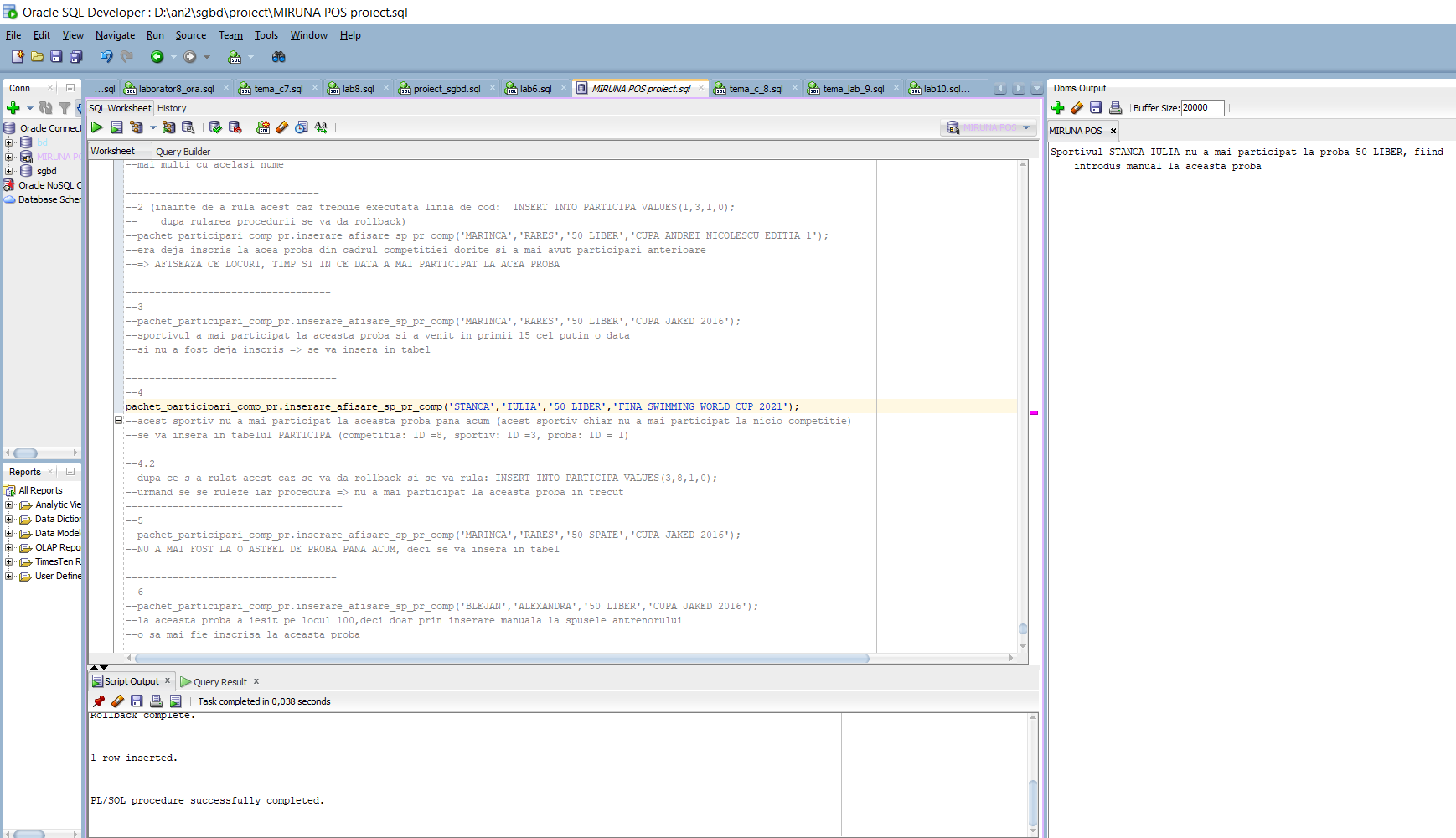
--se va insera in tabelul PARTICIPA (competitia: ID =8, sportiv: ID =3, proba: ID = 1)



--4.2

--dupa ce s-a rulat acest caz se va da rollback si se va rula: INSERT INTO PARTICIPA VALUES(3,8,1,0);

--urmand se se ruleze iar procedura => nu a mai participat la aceasta proba in trecut

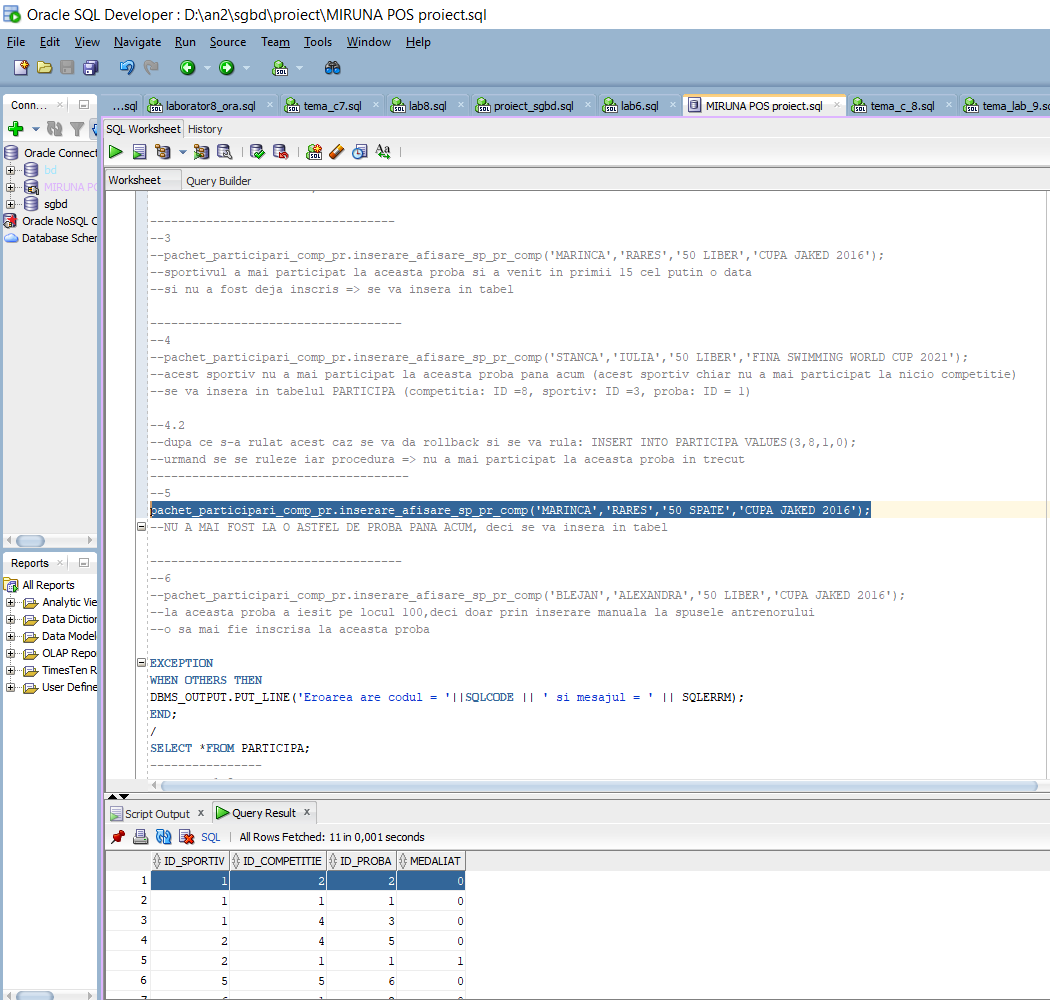


-------------------------------------

--5

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('MARINCA','RARES','50 SPATE','CUPA JAKED 2016');

--NU A MAI FOST LA O ASTFEL DE PROBA PANA ACUM, deci se va insera in tabel



------------------------------------

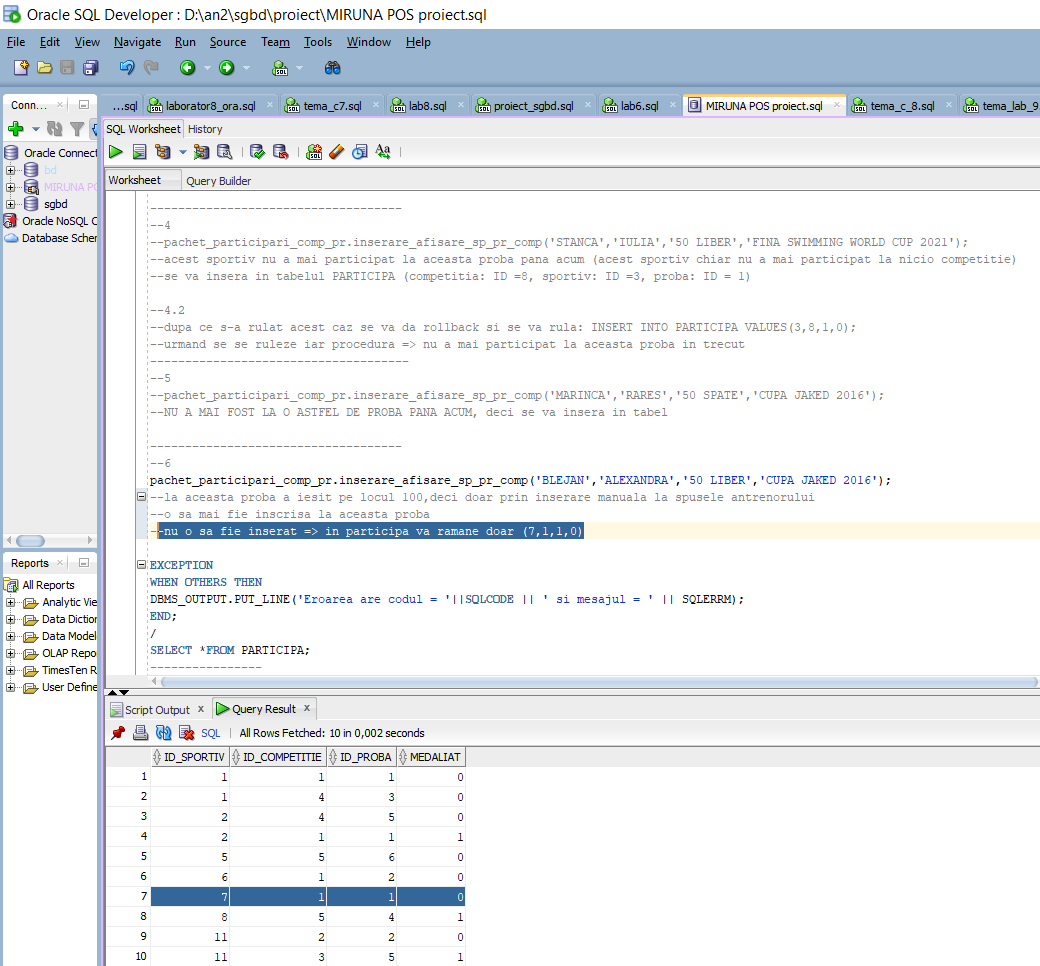
--6

--pachet\_participari\_comp\_pr.inserare\_afisare\_sp\_pr\_comp('BLEJAN','ALEXANDRA','50 LIBER','CUPA JAKED 2016');

--la aceasta proba a iesit pe locul 100,deci doar prin inserare manuala la spusele antrenorului

--o sa mai fie inscrisa la aceasta proba

--nu o sa fie inserat => in participa va ramane doar (7,1,1,0)



EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Eroarea are codul = '||SQLCODE || ' si mesajul = ' || SQLERRM);

END;

/

SELECT \*FROM PARTICIPA;

----------------

--pentru 1.2

INSERT INTO sportiv(id\_sportiv,gen,nume,prenume,anul\_nasterii,id\_nationalitate,tip\_sportiv,indemnizatie,id\_antrenor)

VALUES (13,'F','BELLU','SARA',TO\_DATE('16-08-2001','dd-mm-yyyy'),1,'SPORTIV LOT',301,50);

ROLLBACK;

----------------

--pentru 2.

INSERT INTO PARTICIPA VALUES(1,3,1,0); --COMP 3 => CUPA ANDREI NICOLESCU EDITIA 1

ROLLBACK;

-----------------

--pentru 4.2

INSERT INTO PARTICIPA VALUES(3,8,1,0);--presupunem ca se introduce manual prima participare a unui sportiv la o proba

ROLLBACK;

----------------

/