Gestionarea unei baze de date pentru turnee de esports

Sefcic Adrian Ionut, 342

# Prezentarea si utilitatea bazei de date

Aceasta este o baza de date care contine diferite informatii relevante unui mediu profesionist de esports precum datele jucatorilor si ce meciuri au jucat, echipele acestora si sponsorii echipelor etc.. Utilitatea principala a acestei baze de date este accesul usor la aceste informatii pentru a fi puse in online. (spre ex. <https://liquipedia.net/valorant/Portal:Teams> )

# Diagrama entitate-relatie

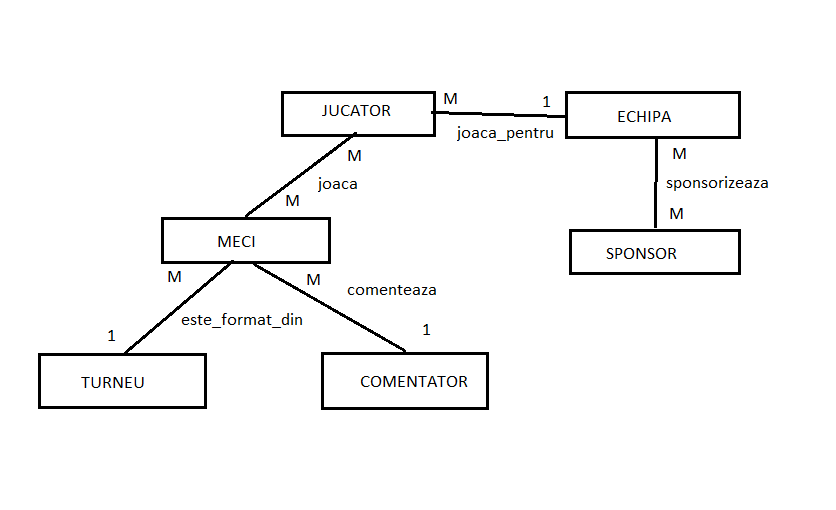
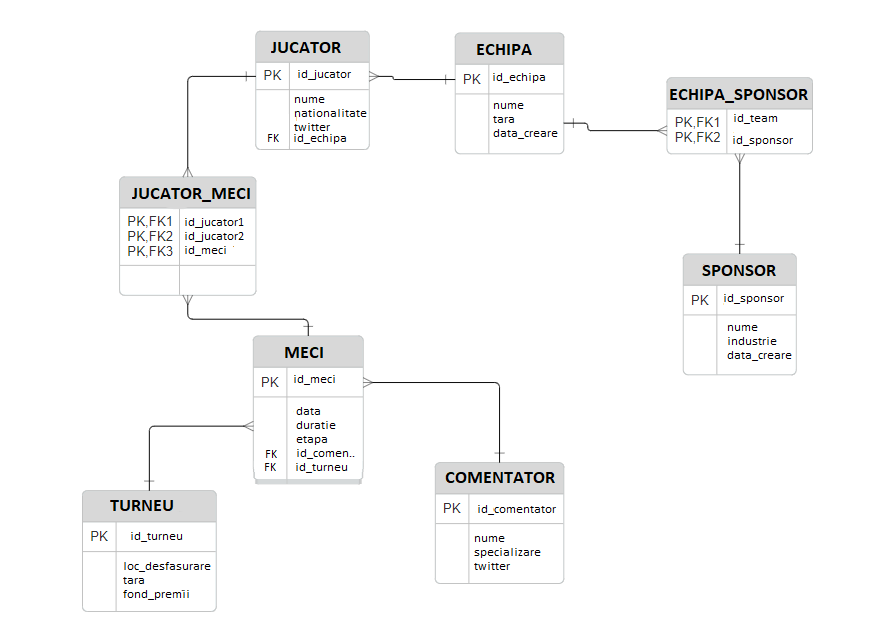


Diagrama Conceptuala



# Implementarea diagramei

CREATE TABLE ECHIPA (

id\_echipa NUMBER PRIMARY KEY,

nume VARCHAR2(50),

tara VARCHAR2(50),

data\_creare DATE

);

CREATE TABLE JUCATOR (

id\_jucator NUMBER PRIMARY KEY,

nume VARCHAR2(50),

nationalitate VARCHAR2(50),

twitter VARCHAR2(30),

id\_echipa NUMBER,

FOREIGN KEY (id\_echipa) REFERENCES ECHIPA(id\_echipa)

);

CREATE TABLE SPONSOR (

id\_sponsor NUMBER PRIMARY KEY,

nume VARCHAR2(50),

industrie VARCHAR2(50),

data\_creare DATE

);

CREATE TABLE ECHIPA\_SPONSOR (

id\_echipa NUMBER,

id\_sponsor NUMBER,

PRIMARY KEY (id\_echipa, id\_sponsor),

FOREIGN KEY (id\_echipa) REFERENCES ECHIPA(id\_echipa),

FOREIGN KEY (id\_sponsor) REFERENCES SPONSOR(id\_sponsor)

);

CREATE TABLE TURNEU (

id\_turneu NUMBER PRIMARY KEY,

loc\_desfasurare VARCHAR2(100),

tara VARCHAR2(50),

fond\_premii NUMBER

);

CREATE TABLE COMENTATOR (

id\_comentator NUMBER PRIMARY KEY,

nume VARCHAR2(100),

specializare VARCHAR2(50),

twitter VARCHAR2(255)

);

CREATE TABLE MECI (

id\_meci NUMBER PRIMARY KEY,

data DATE,

duratie NUMBER,

etapa VARCHAR2(50),

id\_comentator NUMBER,

id\_turneu NUMBER,

FOREIGN KEY (id\_comentator) REFERENCES COMENTATOR(id\_comentator)

);

CREATE TABLE JUCATOR\_MECI (

id\_jucator1 NUMBER,

id\_jucator2 NUMBER,

id\_meci NUMBER,

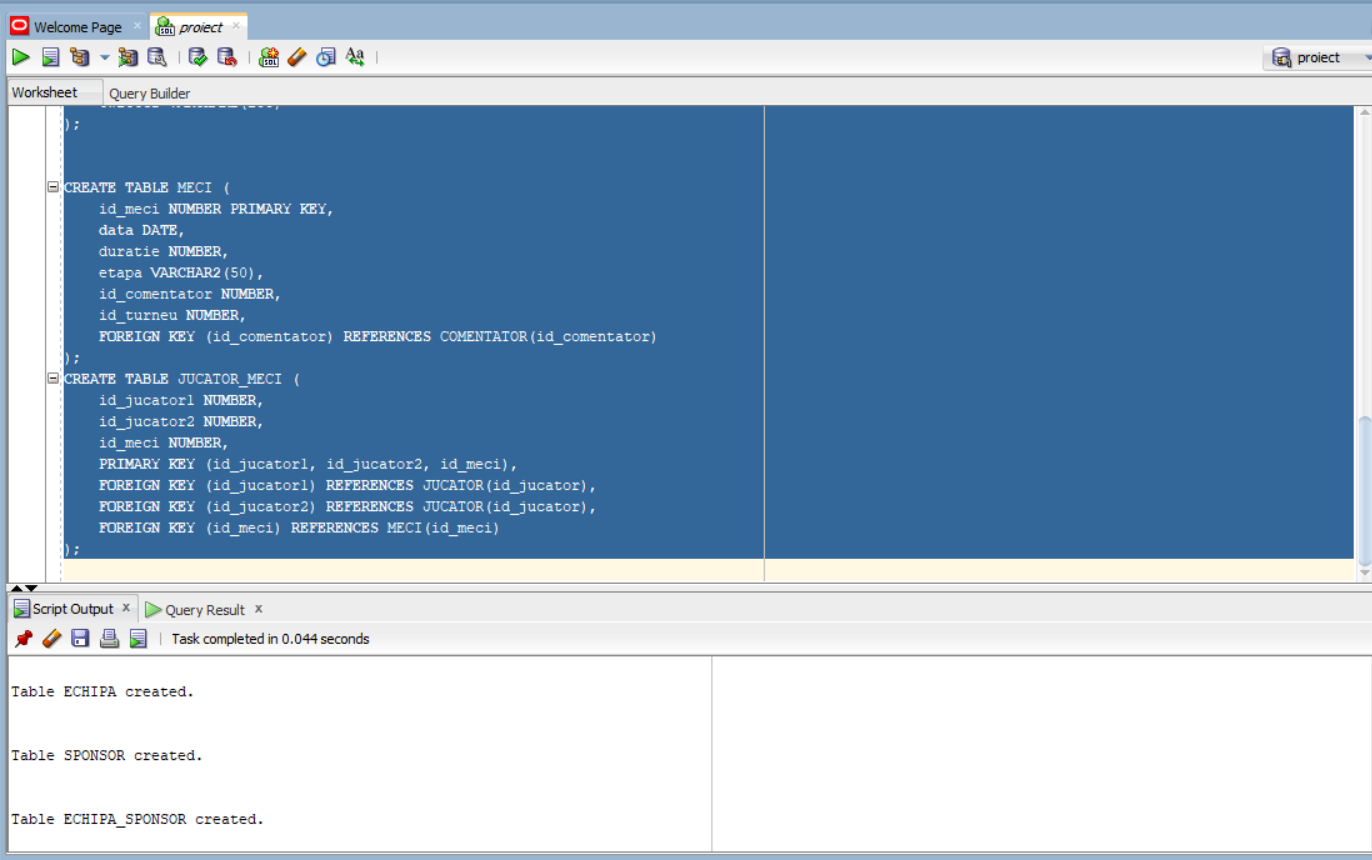
PRIMARY KEY (id\_jucator1, id\_jucator2, id\_meci),

FOREIGN KEY (id\_jucator1) REFERENCES JUCATOR(id\_jucator),

FOREIGN KEY (id\_jucator2) REFERENCES JUCATOR(id\_jucator),

FOREIGN KEY (id\_meci) REFERENCES MECI(id\_meci)

);



# Adaugarea informatiilor in tabele

INSERT INTO COMENTATOR VALUES (1, 'Sophie Müller', 'Color', 'sophiemuller');

INSERT INTO COMENTATOR VALUES (2, 'Luca Rossi', 'Play By Play', 'therealrossi');

INSERT INTO COMENTATOR VALUES (3, 'Elena Popescu', 'Play By Play', 'popescuelena');

INSERT INTO COMENTATOR VALUES (4, 'Andreas Schmidt', 'Play By Play', 'aschmidt');

INSERT INTO COMENTATOR VALUES (5, 'Isabella Conte', 'Color', 'isabella\_conte');

INSERT INTO COMENTATOR VALUES(6, 'Testeanu Verif', 'Play By Play', '@verift');

INSERT INTO TURNEU VALUES (100, 'Berlin Esports Arena', 'Germany', 100000);

INSERT INTO TURNEU VALUES (101, 'Paris Gaming Stadium', 'France', 120000);

INSERT INTO TURNEU VALUES (102, 'London Esports Center', 'United Kingdom', 90000);

INSERT INTO TURNEU VALUES (103, 'Stockholm Esports Hall', 'Sweden', 110000);

INSERT INTO TURNEU VALUES (104, 'Madrid Gaming Arena', 'Spain', 95000);

INSERT INTO SPONSOR VALUES (200, 'TechGear', 'Technology', TO\_DATE('2023-03-01', 'YYYY-MM-DD'));

INSERT INTO SPONSOR VALUES (201, 'FinanceCorp', 'Finance', TO\_DATE('2023-04-01', 'YYYY-MM-DD'));

INSERT INTO SPONSOR VALUES (202, 'GamingTech', 'Gaming', TO\_DATE('2023-05-01', 'YYYY-MM-DD'));

INSERT INTO SPONSOR VALUES (203, 'FashionArena', 'Fashion', TO\_DATE('2023-06-01', 'YYYY-MM-DD'));

INSERT INTO SPONSOR VALUES (204, 'FoodFiesta', 'Food', TO\_DATE('2023-07-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA VALUES (300, 'G2 Esports', 'Germany', TO\_DATE('2023-01-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA VALUES (301, 'Fnatic', 'France', TO\_DATE('2023-02-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA VALUES (302, 'Ninjas in Pyjamas', 'Sweden', TO\_DATE('2023-03-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA VALUES (303, 'Team Vitality', 'Italy', TO\_DATE('2023-04-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA VALUES (304, 'OG Esports', 'Spain', TO\_DATE('2023-05-01', 'YYYY-MM-DD'));

INSERT INTO ECHIPA\_SPONSOR VALUES (300, 200);

INSERT INTO ECHIPA\_SPONSOR VALUES (301, 201);

INSERT INTO ECHIPA\_SPONSOR VALUES (302, 202);

INSERT INTO ECHIPA\_SPONSOR VALUES (303, 203);

INSERT INTO ECHIPA\_SPONSOR VALUES (304, 204);

INSERT INTO ECHIPA\_SPONSOR VALUES (300, 201);

INSERT INTO ECHIPA\_SPONSOR VALUES (301, 202);

INSERT INTO ECHIPA\_SPONSOR VALUES (302, 203);

INSERT INTO ECHIPA\_SPONSOR VALUES (303, 204);

INSERT INTO ECHIPA\_SPONSOR VALUES (304, 200);

INSERT INTO JUCATOR VALUES (400, 'Maximilian Schmidt', 'Germany', 'max\_smith\_gamer', 300);

INSERT INTO JUCATOR VALUES (401, 'Rafael Silva', 'Portugal', 'rafael\_silva\_esports', 300);

INSERT INTO JUCATOR VALUES (402, 'Lukas Novak', 'Czech Republic', 'lukas\_novak\_88', 301);

INSERT INTO JUCATOR VALUES (403, 'Igor Petrov', 'Russia', 'iggy\_petrov', 301);

INSERT INTO JUCATOR VALUES (404, 'Matej Horvat', 'Slovenia', 'matty\_horizon', 302);

INSERT INTO JUCATOR VALUES (405, 'Gustav Berg', 'Sweden', 'gustav\_b\_thunder', 302);

INSERT INTO JUCATOR VALUES (406, 'Luca Rossi', 'Italy', 'luca\_the\_sniper', 303);

INSERT INTO JUCATOR VALUES (407, 'Carlos López', 'Spain', 'carlos\_lightning', 303);

INSERT INTO JUCATOR VALUES (408, 'Sebastian Weber', 'Germany', 'seb\_gaming\_master', 304);

INSERT INTO JUCATOR VALUES (409, 'Antonio Costa', 'Portugal', 'toni\_costa\_legend', 304);

INSERT INTO JUCATOR VALUES (410, 'Adriano Sai', 'Romania', 'saimeistr', 304);

INSERT INTO MECI VALUES (500, TO\_DATE('2023-08-01', 'YYYY-MM-DD'), 30, 'Semifinals', 1, 100);

INSERT INTO MECI VALUES (501, TO\_DATE('2023-08-03', 'YYYY-MM-DD'), 35, 'Semifinals', 2, 100);

INSERT INTO MECI VALUES (502, TO\_DATE('2023-08-06', 'YYYY-MM-DD'), 40, 'Grand Finals', 3, 100);

INSERT INTO MECI VALUES (503, TO\_DATE('2023-08-10', 'YYYY-MM-DD'), 25, 'Semifinals', 4, 101);

INSERT INTO MECI VALUES (504, TO\_DATE('2023-08-15', 'YYYY-MM-DD'), 30, 'Semifinals', 5, 101);

INSERT INTO MECI VALUES (505, TO\_DATE('2023-08-18', 'YYYY-MM-DD'), 35, 'Grand Finals', 1, 101);

INSERT INTO MECI VALUES (506, TO\_DATE('2023-08-22', 'YYYY-MM-DD'), 20, 'Semifinals', 2, 102);

INSERT INTO MECI VALUES (507, TO\_DATE('2023-08-26', 'YYYY-MM-DD'), 25, 'Semifinals', 3, 102);

INSERT INTO MECI VALUES (508, TO\_DATE('2023-08-30', 'YYYY-MM-DD'), 30, 'Grand Finals', 4, 102);

INSERT INTO MECI VALUES (509, TO\_DATE('2023-09-4', 'YYYY-MM-DD'), 33, 'Semifinals', 5, 103);

INSERT INTO MECI VALUES (510, TO\_DATE('2023-09-5', 'YYYY-MM-DD'), 27, 'Semifinals', 1, 103);

INSERT INTO MECI VALUES (511, TO\_DATE('2023-09-7', 'YYYY-MM-DD'), 43, 'Grand Finals', 2, 103);

INSERT INTO MECI VALUES (512, TO\_DATE('2023-09-7', 'YYYY-MM-DD'), 21, 'Showmatch', 1, 103);

INSERT INTO JUCATOR\_MECI VALUES (400, 401, 500);

INSERT INTO JUCATOR\_MECI VALUES (402, 403, 501);

INSERT INTO JUCATOR\_MECI VALUES (400, 402, 502);

INSERT INTO JUCATOR\_MECI VALUES (404, 405, 503);

INSERT INTO JUCATOR\_MECI VALUES (406, 407, 504);

INSERT INTO JUCATOR\_MECI VALUES (404, 406, 505);

INSERT INTO JUCATOR\_MECI VALUES (407, 408, 506);

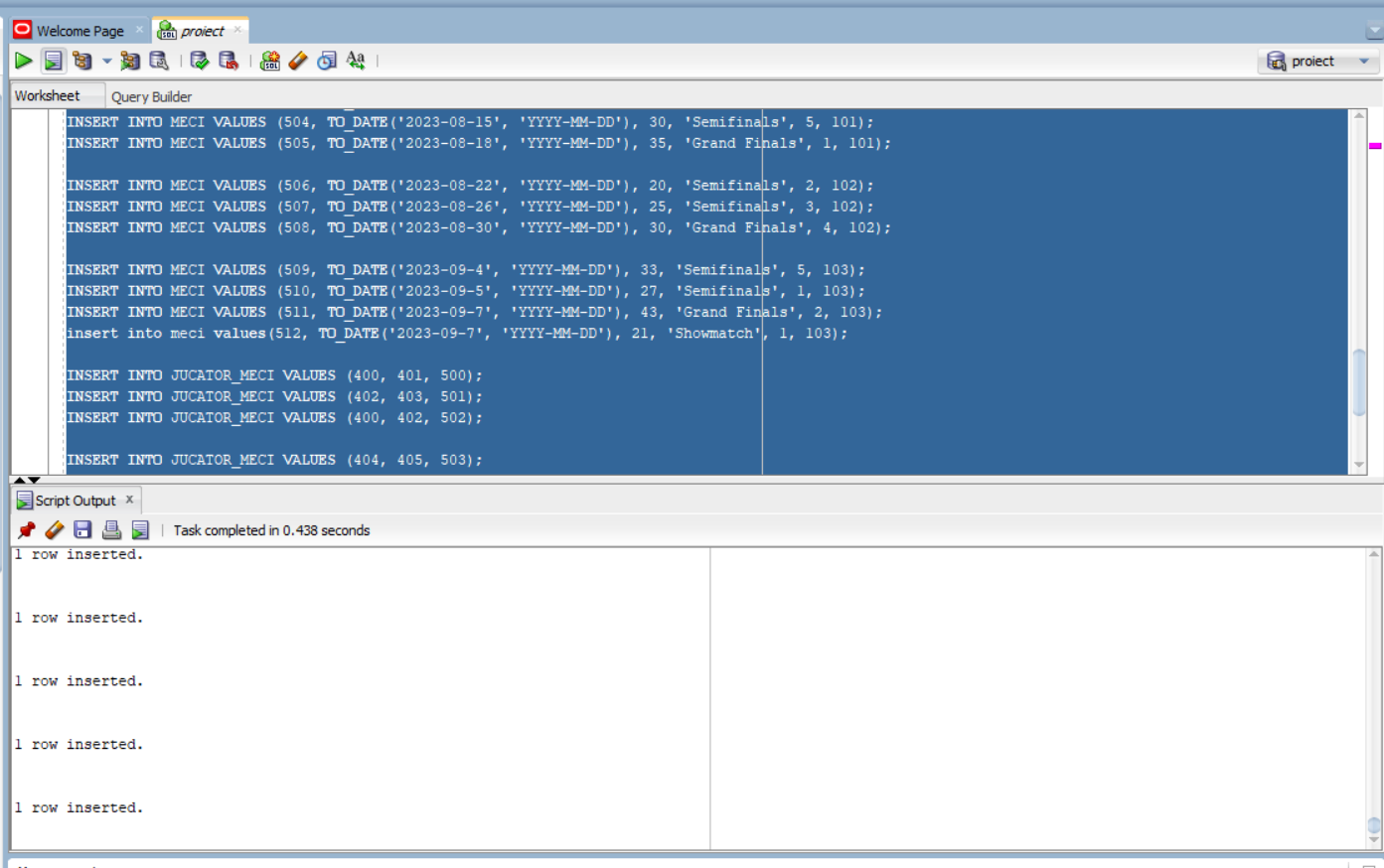
INSERT INTO JUCATOR\_MECI VALUES (409, 401, 507);

INSERT INTO JUCATOR\_MECI VALUES (409, 408, 508);

INSERT INTO JUCATOR\_MECI VALUES (402, 404, 509);

INSERT INTO JUCATOR\_MECI VALUES (401, 407, 510);

INSERT INTO JUCATOR\_MECI VALUES (401, 402, 511);



# CERINTA 6

Date ca parametri nume a doi sponsori sa se afiseze:

-pentru primul, numele si id-ul echipelor pe care le sponsorizeaza

-pentru al doilea, doar numele echipelor

-numarul de echipe sponsorizate pentru fiecare

CREATE OR REPLACE PROCEDURE cerinta6 (nume Sponsor.nume%TYPE, nume\_doi Sponsor.nume%TYPE)

AS

TYPE tabel\_indexat IS TABLE OF Echipa%ROWTYPE INDEX BY PLS\_INTEGER;

t tabel\_indexat;

TYPE tabel\_imbricat IS TABLE OF VARCHAR2(50);

y tabel\_imbricat;

TYPE vector IS VARRAY(20) OF NUMBER;

u vector:= vector();

BEGIN

SELECT E.\*

BULK COLLECT INTO t

FROM ECHIPA E

JOIN ECHIPA\_SPONSOR ES ON E.ID\_ECHIPA = ES.ID\_ECHIPA

JOIN SPONSOR S ON ES.ID\_SPONSOR = S.ID\_SPONSOR

WHERE S.NUME = cerinta6.nume;

u.extend;

u(1) := t.COUNT;

FOR i IN 1..t.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(t(i).nume || ', ' || t(i).id\_echipa);

END LOOP;

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

SELECT E.nume

BULK COLLECT INTO y

FROM ECHIPA E

JOIN ECHIPA\_SPONSOR ES ON E.ID\_ECHIPA = ES.ID\_ECHIPA

JOIN SPONSOR S ON ES.ID\_SPONSOR = S.ID\_SPONSOR

WHERE S.NUME = cerinta6.nume\_doi;

u.extend;

u(2) := y.COUNT;

FOR i IN 1..y.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(y(i));

END LOOP;

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

DBMS\_OUTPUT.PUT\_LINE(u(1) || ', ' || u(2));

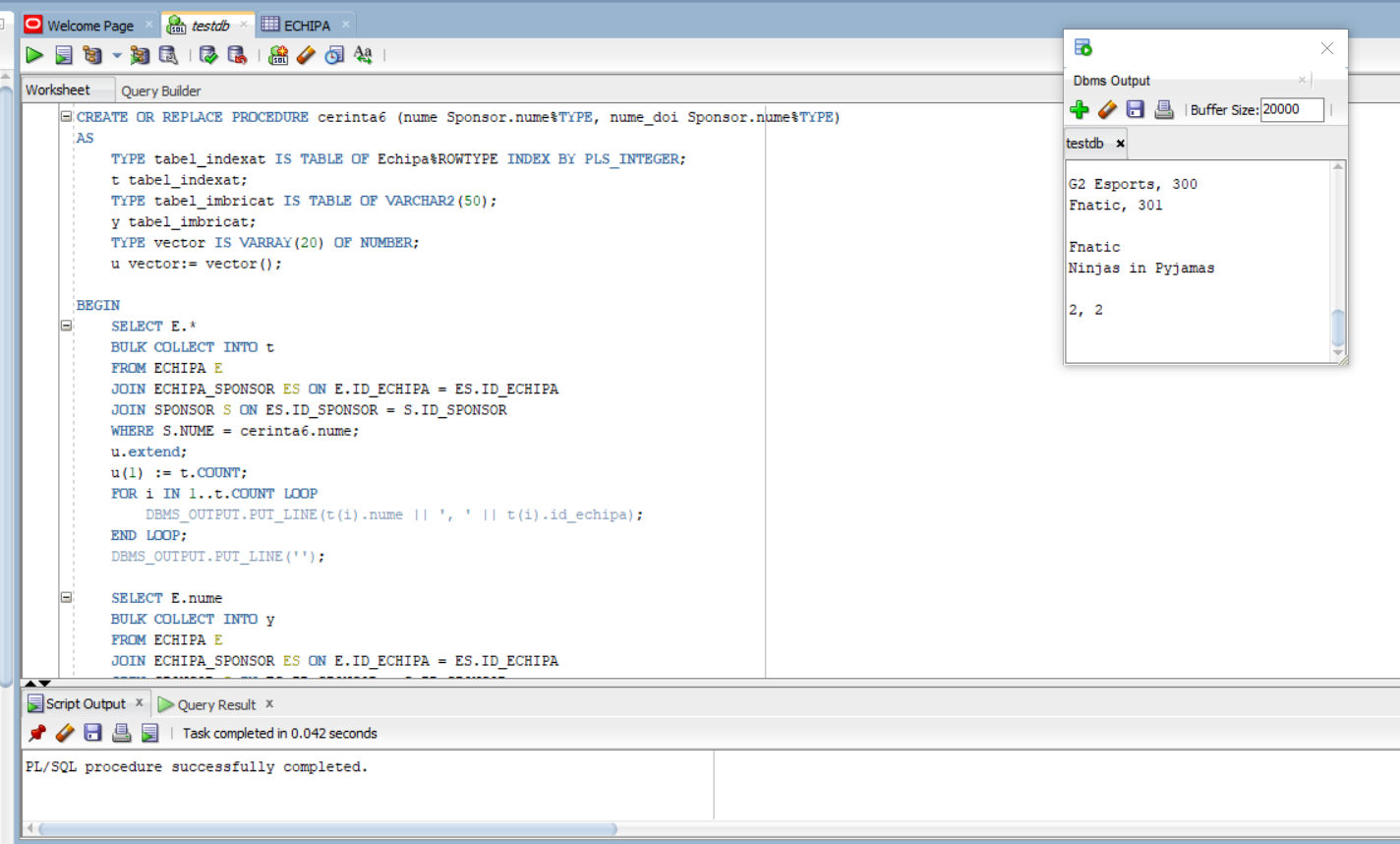
END cerinta6;

/

BEGIN

cerinta6('FinanceCorp', 'GamingTech');

END;

/

# CERINTA 7

Sa se afiseze id-ul si numele fiecarui jucator si numarul de jucatori care fac parte din aceasi echipa ca el (coechipieri).

CREATE OR REPLACE PROCEDURE cerinta7 AS

CURSOR c1 IS

SELECT j.id\_jucator, j.nume, j.id\_echipa

FROM JUCATOR j;

CURSOR c2 (v\_id\_echipa ECHIPA.id\_echipa%TYPE) IS

SELECT COUNT(\*)

FROM ECHIPA e

JOIN JUCATOR j ON e.id\_echipa = j.id\_echipa

WHERE e.id\_echipa = v\_id\_echipa;

v\_id\_jucator JUCATOR.id\_jucator%TYPE;

v\_nume\_jucator JUCATOR.nume%TYPE;

v\_id\_echipa ECHIPA.id\_echipa%TYPE;

v\_total\_jucatori NUMBER;

BEGIN

OPEN c1;

LOOP

FETCH c1 INTO v\_id\_jucator, v\_nume\_jucator, v\_id\_echipa;

EXIT WHEN c1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_id\_jucator || ', ' || v\_nume\_jucator);

OPEN c2(v\_id\_echipa);

FETCH c2 INTO v\_total\_jucatori;

DBMS\_OUTPUT.PUT\_LINE('are ' || (v\_total\_jucatori - 1) || ' coechipier(i).');

CLOSE c2;

END LOOP;

CLOSE c1;

END;

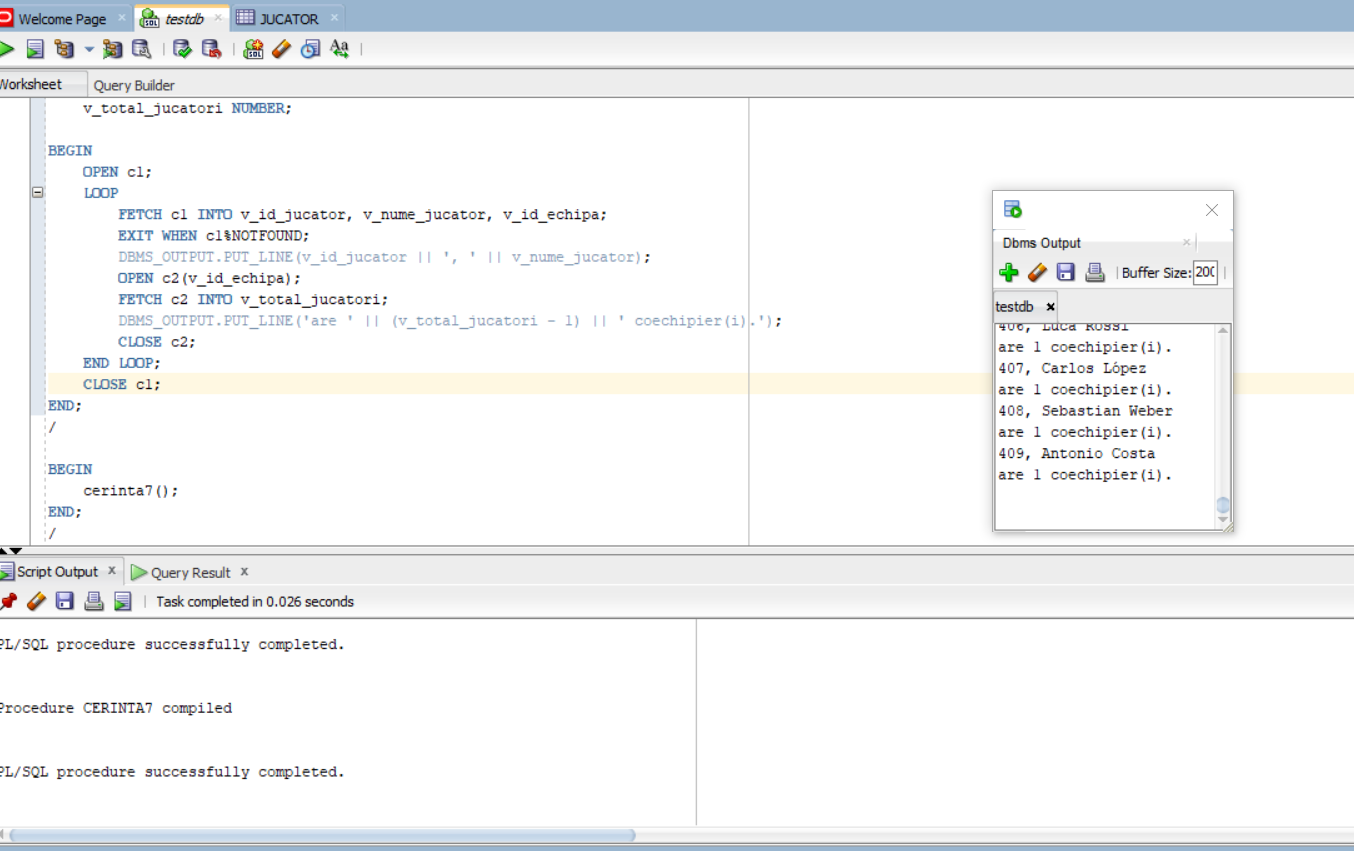
/

BEGIN

cerinta7();

END;

/



# CERINTA 8

Sa se afiseze numarul de turnee la care un commentator a fost prezent la meciuri.

CREATE OR REPLACE FUNCTION cerinta8(id NUMBER) RETURN NUMBER IS

nr\_turnee NUMBER;

aux NUMBER;

nume\_com COMENTATOR.nume%TYPE;

v\_comentator COMENTATOR%ROWTYPE;

WRONG\_ID EXCEPTION;

NO\_TOURNAMENTS EXCEPTION;

BEGIN

SELECT COUNT(\*) into aux

FROM comentator

WHERE id\_comentator = id;

IF aux = 0 THEN

RAISE WRONG\_ID;

END IF;

SELECT COUNT(DISTINCT t.id\_turneu) into nr\_turnee

FROM MECI m

JOIN COMENTATOR c on m.id\_comentator = c.id\_comentator

JOIN TURNEU t on m.id\_turneu = t.id\_turneu

WHERE m.id\_comentator = id;

IF nr\_turnee = 0 THEN

RAISE NO\_TOURNAMENTS;

END IF;

RETURN nr\_turnee;

EXCEPTION

WHEN WRONG\_ID THEN

DBMS\_OUTPUT.PUT\_LINE('Niciun comentator nu are acest id');

RETURN -1;

WHEN NO\_TOURNAMENTS THEN

DBMS\_OUTPUT.PUT\_LINE('Comentatorul nu a fost la niciun turneu');

RETURN -1;

END;

/

DECLARE

nr NUMBER;

BEGIN

nr := cerinta8(1);

DBMS\_OUTPUT.PUT\_LINE(nr);

nr := cerinta8(6);

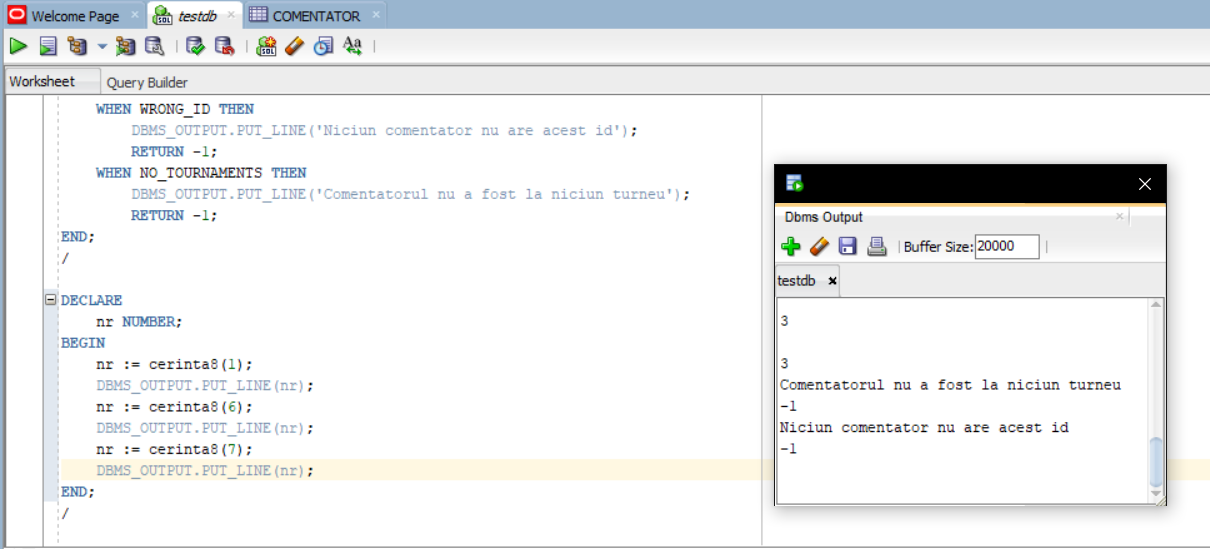
DBMS\_OUTPUT.PUT\_LINE(nr);

nr := cerinta8(7);

DBMS\_OUTPUT.PUT\_LINE(nr);

END;

/



# CERINTA 9

Dandu-se ca parametru id-ul unui meci sa se afiseze sponsorii prezenti in meciul respectiv.

CREATE OR REPLACE PROCEDURE cerinta9(id NUMBER)

AS

TYPE vector\_id\_echipe IS VARRAY(2) OF NUMBER;

u vector\_id\_echipe:= vector\_id\_echipe();

v\_sponsor VARCHAR2(50);

NO\_DATA\_FOUND EXCEPTION;

TOO\_MANY\_ROWS EXCEPTION;

aux NUMBER;

WRONG\_ID EXCEPTION;

BEGIN

SELECT COUNT(\*) into aux

FROM meci

WHERE id\_meci = id;

IF aux = 0 THEN

RAISE WRONG\_ID;

END IF;

FOR rec IN (

SELECT DISTINCT j.id\_echipa

FROM meci m

JOIN jucator\_meci jm ON m.id\_meci = jm.id\_meci

JOIN jucator j ON jm.id\_jucator1 = j.id\_jucator OR jm.id\_jucator2 = j.id\_jucator

WHERE m.id\_meci = id

) LOOP

u.EXTEND;

u(u.LAST) := rec.id\_echipa;

END LOOP;

IF u(u.LAST) IS null THEN

RAISE NO\_DATA\_FOUND;

RETURN;

END IF;

IF u.COUNT > 2 THEN

RAISE TOO\_MANY\_ROWS;

RETURN;

END IF;

DBMS\_OUTPUT.PUT\_LINE(u.LAST);

DBMS\_OUTPUT.PUT\_LINE('Pentru meciul ' || id);

FOR val in 1..u.COUNT LOOP

IF u.COUNT = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Ambii jucatori fac parte din aceeasi organizatie (' || u(1) || ')');

DBMS\_OUTPUT.PUT\_LINE('Sponsorii prezenti sunt:');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Echipa jucatorului ' || val || ' (' || u(1) || ') are sponsorii:');

END IF;

FOR rez IN (SELECT s.nume

FROM sponsor s

JOIN echipa\_sponsor es ON es.id\_sponsor = s.id\_sponsor

JOIN echipa e ON e.id\_Echipa = es.id\_echipa AND e.id\_echipa = u(val))

LOOP

v\_sponsor := rez.nume;

DBMS\_OUTPUT.PUT\_LINE(v\_sponsor);

END LOOP;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('no data found');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('too many rows');

WHEN WRONG\_ID THEN

DBMS\_OUTPUT.PUT\_LINE('wrong id');

END;

/

BEGIN

cerinta9(511);DBMS\_OUTPUT.PUT\_LINE(' ');

cerinta9(513);DBMS\_OUTPUT.PUT\_LINE(' ');

cerinta9(550);DBMS\_OUTPUT.PUT\_LINE(' ');

END;

/

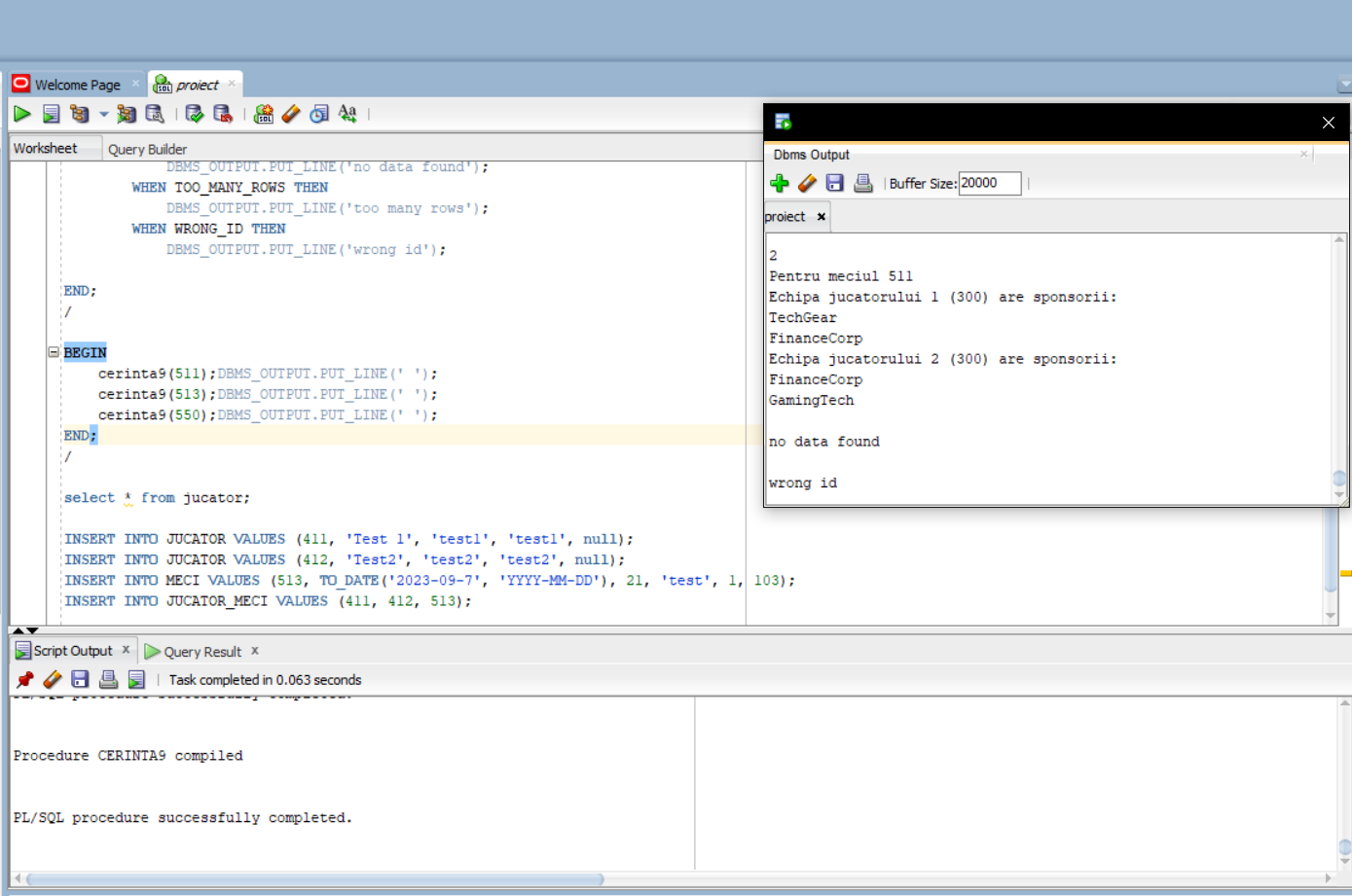
A fost nevoie de niste inserturi extra pentru a declansa

INSERT INTO JUCATOR VALUES (411, 'Test 1', 'test1', 'test1', null);

INSERT INTO JUCATOR VALUES (412, 'Test2', 'test2', 'test2', null);

INSERT INTO MECI VALUES (513, TO\_DATE('2023-09-7', 'YYYY-MM-DD'), 21, 'test', 1, 103);

INSERT INTO JUCATOR\_MECI VALUES (411, 412, 513);



# CERINTA 10

CREATE OR REPLACE TRIGGER cerinta10

BEFORE INSERT OR UPDATE OR DELETE ON MECI

DECLARE

ora CONSTANT NUMBER := 18;

BEGIN

IF TO\_NUMBER(TO\_CHAR(SYSDATE, 'HH24')) >= ora THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Operatiuni interzise dupa ora ' || ora || ':00');

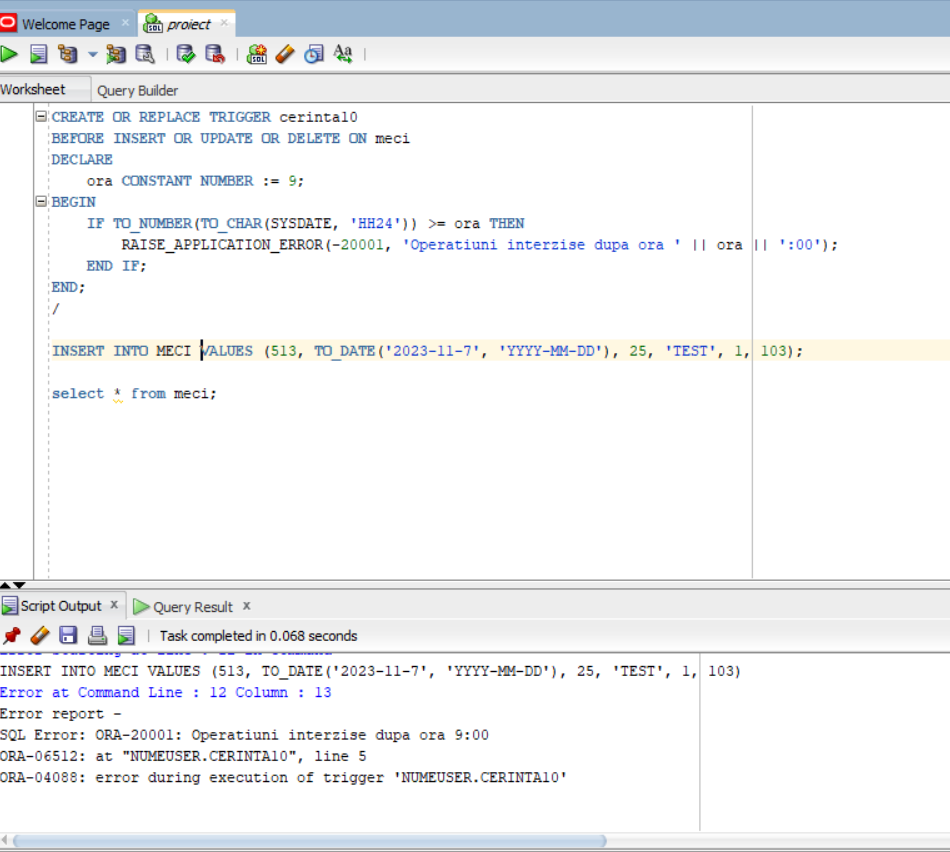
END IF;

END;

/

INSERT INTO MECI VALUES (513, TO\_DATE('2023-11-7', 'YYYY-MM-DD'), 25, 'TEST', 1, 103);

Pentru testare am folosit ora ca fiind 9



# CERINTA 11

CREATE OR REPLACE TRIGGER cerinta11

BEFORE UPDATE OF id\_jucator ON JUCATOR

FOR EACH ROW

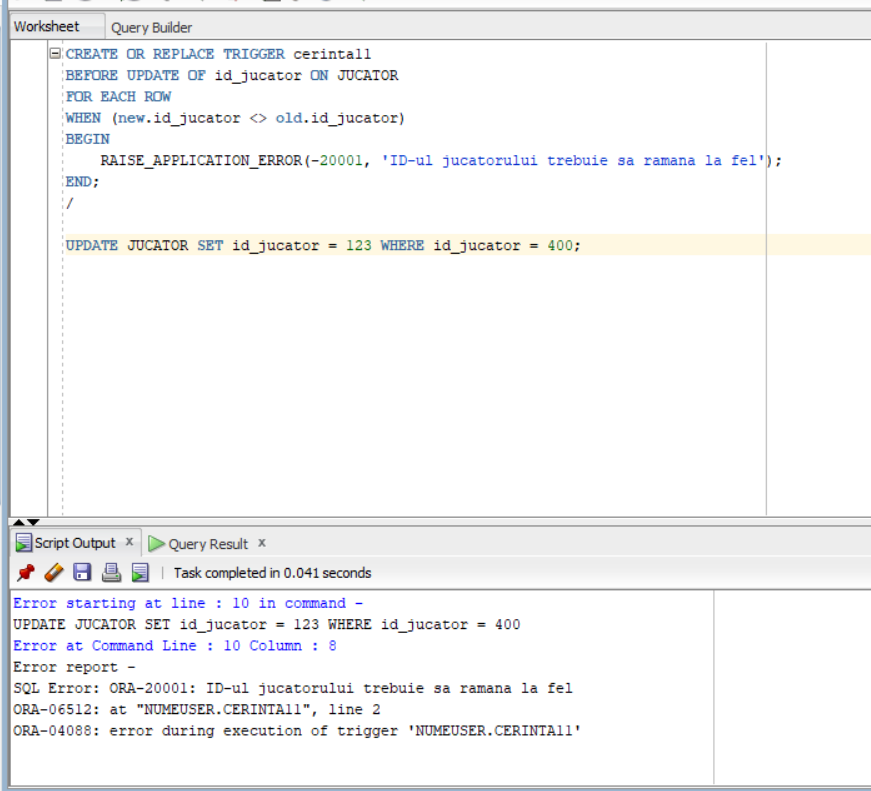
WHEN (new.id\_jucator <> old.id\_jucator)

BEGIN

RAISE\_APPLICATION\_ERROR(-20001, 'ID-ul jucatorului trebuie sa ramana la fel');

END;

/



# CERINTA 12

CREATE TABLE audit\_cerinta12(

utilizator VARCHAR2(30),

nume\_bd VARCHAR2(50),

eveniment VARCHAR2(20),

nume\_obiect VARCHAR2(30),

data DATE);

CREATE OR REPLACE TRIGGER cerinta12

AFTER CREATE OR DROP OR ALTER ON SCHEMA

BEGIN

INSERT INTO audit\_cerinta12

VALUES (SYS.LOGIN\_USER, SYS.DATABASE\_NAME, SYS.SYSEVENT,

SYS.DICTIONARY\_OBJ\_NAME, SYSDATE);

END;

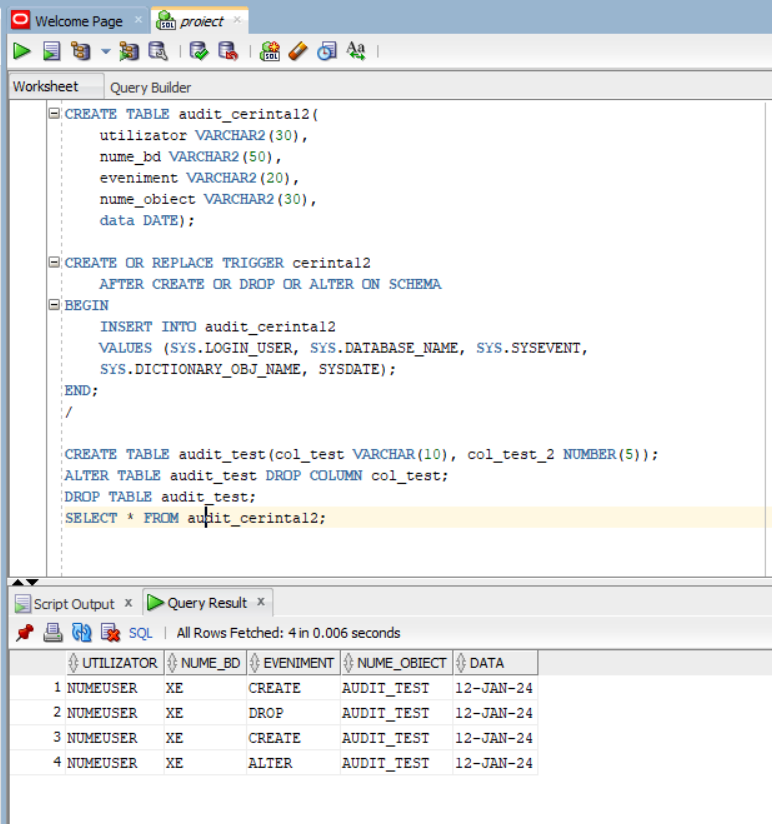
/

CREATE TABLE audit\_test(col\_test VARCHAR(10), col\_test\_2 NUMBER(5));

ALTER TABLE audit\_test DROP COLUMN col\_test;

DROP TABLE audit\_test;

SELECT \* FROM audit\_cerinta12;



# CERINTA 13

CREATE OR REPLACE PACKAGE cerinta13

AS

PROCEDURE cerinta6(nume Sponsor.nume%TYPE, nume\_doi Sponsor.nume%TYPE);

PROCEDURE cerinta7;

FUNCTION cerinta8(id NUMBER) RETURN NUMBER;

PROCEDURE cerinta9(id NUMBER);

END cerinta13;

/

CREATE OR REPLACE PACKAGE BODY cerinta13

AS

PROCEDURE cerinta6 (nume Sponsor.nume%TYPE, nume\_doi Sponsor.nume%TYPE) AS

TYPE tabel\_indexat IS TABLE OF Echipa%ROWTYPE INDEX BY PLS\_INTEGER;

t tabel\_indexat;

TYPE tabel\_imbricat IS TABLE OF VARCHAR2(50);

y tabel\_imbricat;

TYPE vector IS VARRAY(20) OF NUMBER;

u vector:= vector();

BEGIN

SELECT E.\*

BULK COLLECT INTO t

FROM ECHIPA E

JOIN ECHIPA\_SPONSOR ES ON E.ID\_ECHIPA = ES.ID\_ECHIPA

JOIN SPONSOR S ON ES.ID\_SPONSOR = S.ID\_SPONSOR

WHERE S.NUME = cerinta6.nume;

u.extend;

u(1) := t.COUNT;

FOR i IN 1..t.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(t(i).nume || ', ' || t(i).id\_echipa);

END LOOP;

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

SELECT E.nume

BULK COLLECT INTO y

FROM ECHIPA E

JOIN ECHIPA\_SPONSOR ES ON E.ID\_ECHIPA = ES.ID\_ECHIPA

JOIN SPONSOR S ON ES.ID\_SPONSOR = S.ID\_SPONSOR

WHERE S.NUME = cerinta6.nume\_doi;

u.extend;

u(2) := y.COUNT;

FOR i IN 1..y.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE(y(i));

END LOOP;

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

DBMS\_OUTPUT.PUT\_LINE(u(1) || ', ' || u(2));

END;

PROCEDURE cerinta7 AS

CURSOR c1 IS

SELECT j.id\_jucator, j.nume, j.id\_echipa

FROM JUCATOR j;

CURSOR c2 (v\_id\_echipa ECHIPA.id\_echipa%TYPE) IS

SELECT COUNT(\*)

FROM ECHIPA e

JOIN JUCATOR j ON e.id\_echipa = j.id\_echipa

WHERE e.id\_echipa = v\_id\_echipa;

v\_id\_jucator JUCATOR.id\_jucator%TYPE;

v\_nume\_jucator JUCATOR.nume%TYPE;

v\_id\_echipa ECHIPA.id\_echipa%TYPE;

v\_total\_jucatori NUMBER;

BEGIN

OPEN c1;

LOOP

FETCH c1 INTO v\_id\_jucator, v\_nume\_jucator, v\_id\_echipa;

EXIT WHEN c1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(v\_id\_jucator || ', ' || v\_nume\_jucator);

OPEN c2(v\_id\_echipa);

FETCH c2 INTO v\_total\_jucatori;

DBMS\_OUTPUT.PUT\_LINE('are ' || (v\_total\_jucatori - 1) || ' coechipier(i).');

CLOSE c2;

END LOOP;

CLOSE c1;

END;

FUNCTION cerinta8(id NUMBER) RETURN NUMBER IS

nr\_turnee NUMBER;

aux NUMBER;

nume\_com COMENTATOR.nume%TYPE;

v\_comentator COMENTATOR%ROWTYPE;

WRONG\_ID EXCEPTION;

NO\_TOURNAMENTS EXCEPTION;

BEGIN

SELECT COUNT(\*) into aux

FROM comentator

WHERE id\_comentator = id;

IF aux = 0 THEN

RAISE WRONG\_ID;

END IF;

SELECT COUNT(DISTINCT t.id\_turneu) into nr\_turnee

FROM MECI m

JOIN COMENTATOR c on m.id\_comentator = c.id\_comentator

JOIN TURNEU t on m.id\_turneu = t.id\_turneu

WHERE m.id\_comentator = id;

IF nr\_turnee = 0 THEN

RAISE NO\_TOURNAMENTS;

END IF;

RETURN nr\_turnee;

EXCEPTION

WHEN WRONG\_ID THEN

DBMS\_OUTPUT.PUT\_LINE('Niciun comentator nu are acest id');

RETURN -1;

WHEN NO\_TOURNAMENTS THEN

DBMS\_OUTPUT.PUT\_LINE('Comentatorul nu a fost la niciun turneu');

RETURN -1;

END;

PROCEDURE cerinta9(id NUMBER)

AS

TYPE vector\_id\_echipe IS VARRAY(2) OF NUMBER;

u vector\_id\_echipe:= vector\_id\_echipe();

v\_sponsor VARCHAR2(50);

NO\_DATA\_FOUND EXCEPTION;

TOO\_MANY\_ROWS EXCEPTION;

aux NUMBER;

WRONG\_ID EXCEPTION;

BEGIN

SELECT COUNT(\*) into aux

FROM meci

WHERE id\_meci = id;

IF aux = 0 THEN

RAISE WRONG\_ID;

END IF;

FOR rec IN (

SELECT DISTINCT j.id\_echipa

FROM meci m

JOIN jucator\_meci jm ON m.id\_meci = jm.id\_meci

JOIN jucator j ON jm.id\_jucator1 = j.id\_jucator OR jm.id\_jucator2 = j.id\_jucator

WHERE m.id\_meci = id

) LOOP

u.EXTEND;

u(u.LAST) := rec.id\_echipa;

END LOOP;

IF u(u.LAST) IS null THEN

RAISE NO\_DATA\_FOUND;

RETURN;

END IF;

IF u.COUNT > 2 THEN

RAISE TOO\_MANY\_ROWS;

RETURN;

END IF;

DBMS\_OUTPUT.PUT\_LINE(u.LAST);

DBMS\_OUTPUT.PUT\_LINE('Pentru meciul ' || id);

FOR val in 1..u.COUNT LOOP

IF u.COUNT = 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Ambii jucatori fac parte din aceeasi organizatie (' || u(1) || ')');

DBMS\_OUTPUT.PUT\_LINE('Sponsorii prezenti sunt:');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Echipa jucatorului ' || val || ' (' || u(1) || ') are sponsorii:');

END IF;

FOR rez IN (SELECT s.nume

FROM sponsor s

JOIN echipa\_sponsor es ON es.id\_sponsor = s.id\_sponsor

JOIN echipa e ON e.id\_Echipa = es.id\_echipa AND e.id\_echipa = u(val))

LOOP

v\_sponsor := rez.nume;

DBMS\_OUTPUT.PUT\_LINE(v\_sponsor);

END LOOP;

END LOOP;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('no data found');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('too many rows');

WHEN WRONG\_ID THEN

DBMS\_OUTPUT.PUT\_LINE('wrong id');

END;

END cerinta13;

/

