

Proiect Sisteme de Gestiune a Bazelor de Date

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

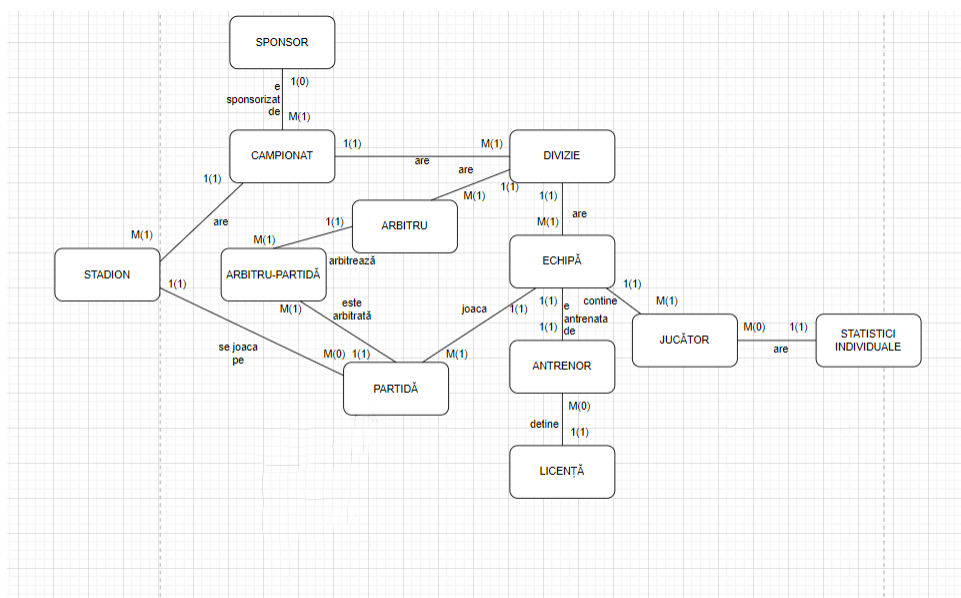
Baza de date stochează și gestionează informații despre campionatele de fotbal ale lumii, ale echipelor care evoluează în aceste campionate, precum și jucătorii din componența echipelor.

Scopul creării acestei baze de date este posibilitatea reținerii istoricului fiecărui sezon de fotbal, disputat în toate țările lumii, pentru a nu se pierde tradiția acestui sport iubit de numeroșii fani internaționali.

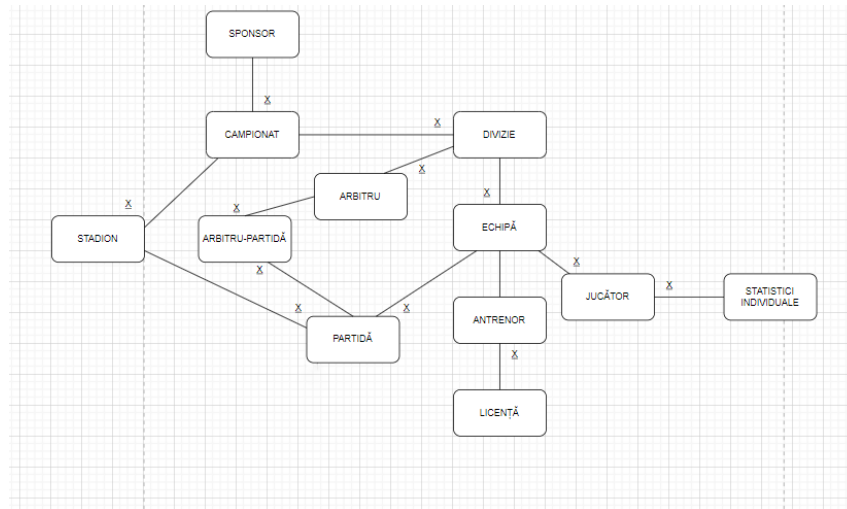
Fiecare campionat poate avea un sponsor și este împărțit pe mai multe divizii, în funcție de calitatea echipelor care evoluează în acestea. Un campionat are un număr de stadioane pe care echipele care evoluează pot susține partide. Orice divizie are asociată arbitrii care iau parte la meciurile din cadrul diviziei specifice.

Echipele de fotbal au în componența sa mai mulți jucători, iar fiecare jucător are anumite statistici individuale obținute pe timpul partidelor (număr de goluri date, număr de assist-uri etc). Orice echipă este antrenată de un singur antrenor, care la rândul său NU poate antrena mai multe echipe simultan, iar fiecare antrenor deține o licență care îi permite să antreneze la un nivel cât mai înalt. Fiecare echipă dispută mai multe partide.

2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați **diagrama conceptuală** a modelului propus, integrând toate atributele necesare.



4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementați toate constrângerile de integritate necesare (chei primare, chei externe etc).
5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

Pentru aceste 2 cerințe, din cauza faptului că am inserat un număr destul de mare de înregistrări în tabele, am considerat că este ok păstrarea acestor 2 cerințe doar în fișierul .sql, pentru a nu încărca și acest document.

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.

-- 6. Pentru o lista de id uri de jucatori date, afisati in consola

-- cate goluri a in scris fiecare.

create or replace type tablou IS TABLE OF NUMBER;

CREATE OR REPLACE PROCEDURE number_of_goals (players IN tablou) IS

gol NUMBER;

type vector is TABLE OF NUMBER INDEX BY PLS_INTEGER;

goals vector;

BEGIN

```
FOR i in players.FIRST..players.LAST LOOP

SELECT STATISTICI_INDIVIDUALE.numar_goluri

INTO gol

FROM STATISTICI_INDIVIDUALE, JUCATOR

WHERE JUCATOR.id_jucator = players(i) AND

STATISTICI_INDIVIDUALE.id_statistici = JUCATOR.id_statistici;

goals(i) := gol;

END LOOP;

FOR i in goals.FIRST..goals.LAST LOOP

DBMS_OUTPUT.PUT_LINE('Jucatorul cu ID ' || players(i) || ' a inscris '

|| goals(i) || ' goluri.');
```



```
END LOOP;

END;

DECLARE

input tablou := tablou(1,3,7,9,23,81,27,123);

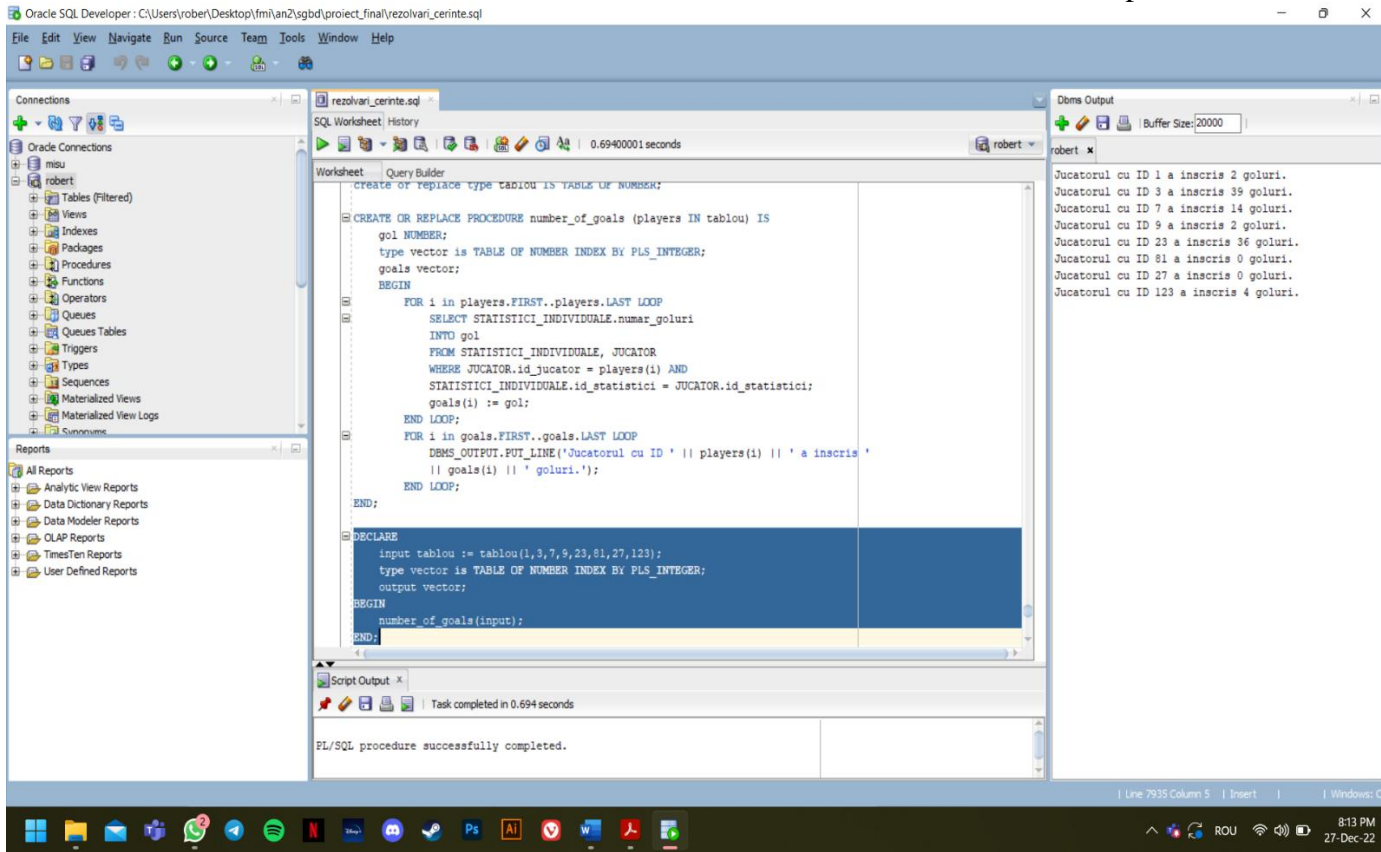
type vector is TABLE OF NUMBER INDEX BY PLS_INTEGER;

output vector;

BEGIN

number_of_goals(input);

END;
```



7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.

-- 7. Pentru fiecare jucator din divizia Premier League (id_divizie=1) sa se

-- calculeze contributia sa la goluri (goluri + pase de gol) si sa se afiseze

CREATE OR REPLACE PROCEDURE contributii IS

CURSOR prem_players IS

SELECT JUCATOR.id_jucator id

FROM JUCATOR, ECHIPA

WHERE JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id_divizie = 1;

CURSOR contributie(j_id NUMBER) IS

SELECT STATISTICI_INDIVIDUALE.numar_goluri + STATISTICI_INDIVIDUALE.numar_assisturi

FROM STATISTICI_INDIVIDUALE, JUCATOR

WHERE STATISTICI_INDIVIDUALE.id_statistici = JUCATOR.id_statistici AND

JUCATOR.id_jucator = j_id;

```
g_a NUMBER;
```

```
BEGIN
```

```
FOR p_id IN prem_players LOOP
```

```
OPEN contributie(p_id.id);
```

```
LOOP
```

```
FETCH contributie INTO g_a;
```

```
EXIT WHEN contributie%NOTFOUND;
```

```
DBMS_OUTPUT.PUT_LINE(p_id.id || ' cu contributia ' || g_a);
```

```
END LOOP;
```

```
CLOSE contributie;
```

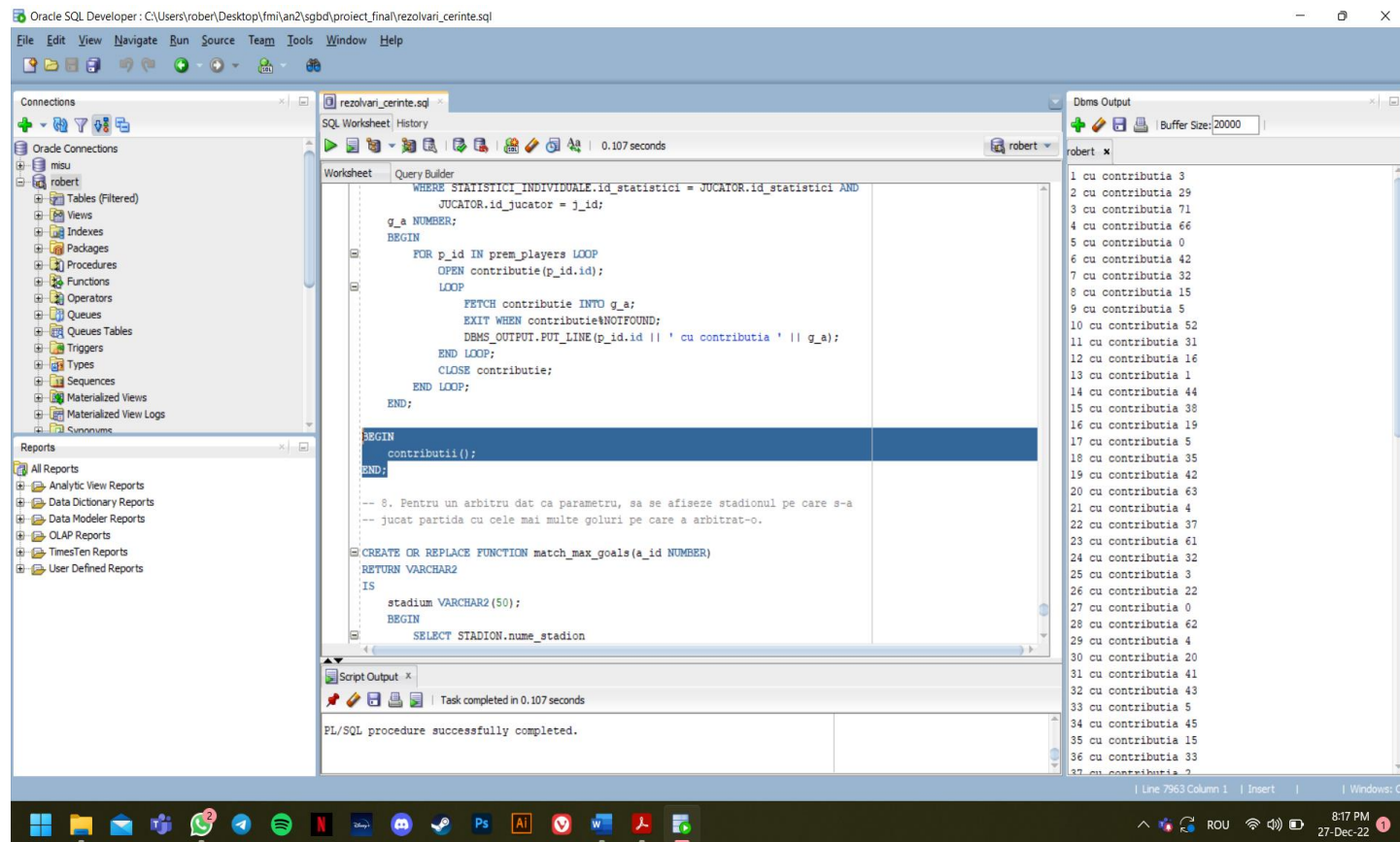
```
END LOOP;
```

```
END;
```

```
BEGIN
```

```
contributii();
```

```
END;
```



8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

-- 8. Pentru un arbitru dat ca parametru, sa se afiseze stadionul pe care s-a

-- jucat partida cu cele mai multe goluri pe care a arbitrat-o.

CREATE OR REPLACE FUNCTION match_max_goals(a_id NUMBER)

RETURN VARCHAR2

IS

 stadium VARCHAR2(50);

BEGIN

 SELECT STADION.ume_stadion

 INTO stadium

 FROM STADION, ARBITRU, PARTIDA, ARBITRU_PARTIDA

 WHERE ARBITRU.id_arbitru = a_id AND ARBITRU_PARTIDA.id_partida =

 PARTIDA.id_partida AND ARBITRU.id_arbitru = ARBITRU_PARTIDA.id_arbitru

 AND STADION.id_stadion = PARTIDA.id_stadion AND

 PARTIDA.nr_goluri_gazda + PARTIDA.nr_goluri_oaspeti = (

 SELECT max(PARTIDA.nr_goluri_gazda + PARTIDA.nr_goluri_oaspeti)

 FROM PARTIDA, ARBITRU_PARTIDA, ARBITRU

 WHERE ARBITRU.id_arbitru = a_id AND PARTIDA.id_partida =

 ARBITRU_PARTIDA.id_partida AND ARBITRU.id_arbitru =

 ARBITRU_PARTIDA.id_arbitru);

RETURN stadium;

EXCEPTION

 WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nici o partida arbitrata'); return null;

 WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE('Mai multe meciuri cu aceleasi goluri'); return null;

END;

DECLARE

test1 VARCHAR2(50);

test2 VARCHAR2(50);

```
test3 VARCHAR2(50);
```

```
BEGIN
```

```
test1 := match_max_goals(1);
```

```
test2 := match_max_goals(25);
```

```
test3 := match_max_goals(39);
```

```
IF test1 IS NOT NULL THEN
```

```
    DBMS_OUTPUT.PUT_LINE(test1);
```

```
END IF;
```

```
IF test2 IS NOT NULL THEN
```

```
    DBMS_OUTPUT.PUT_LINE(test2);
```

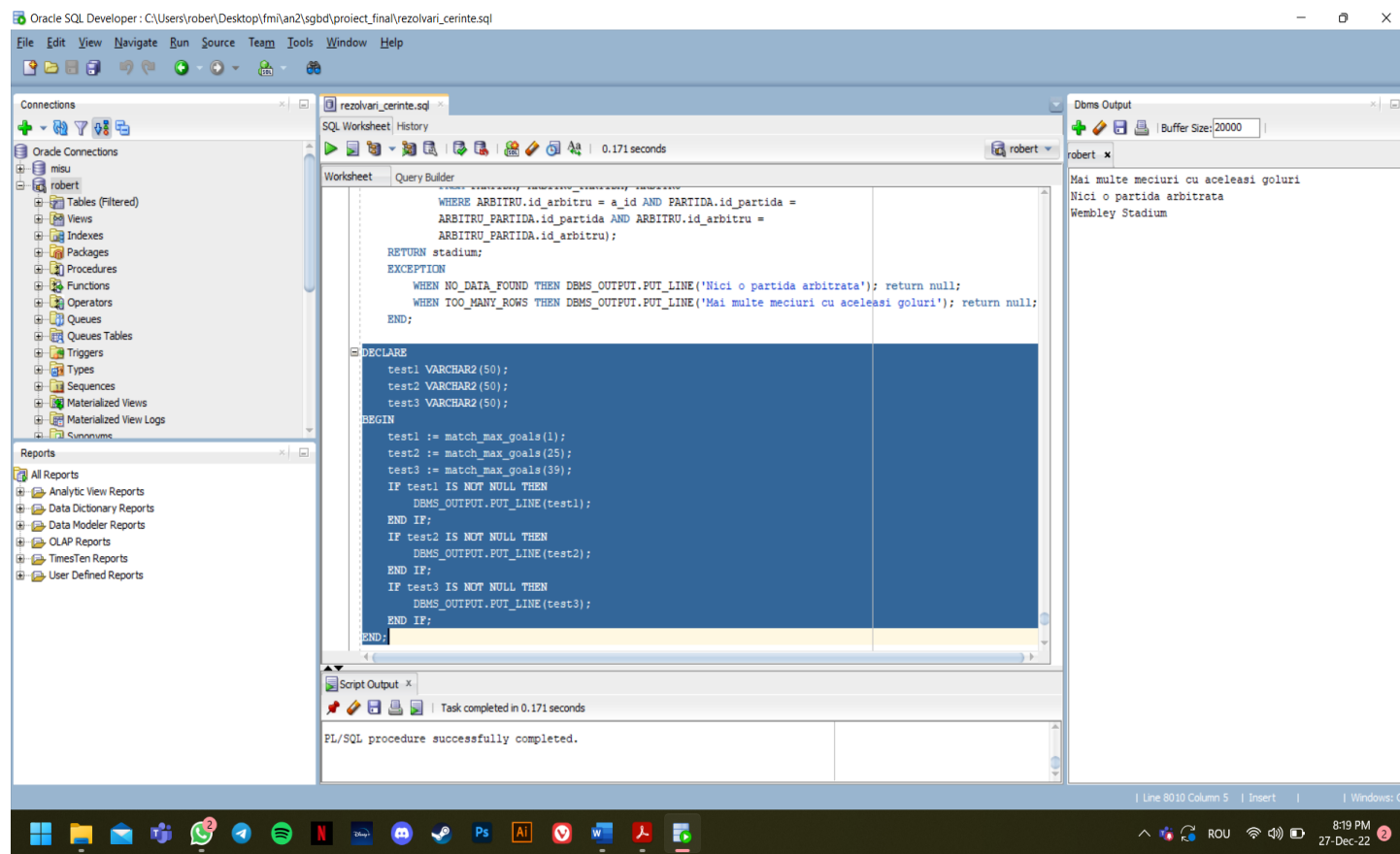
```
END IF;
```

```
IF test3 IS NOT NULL THEN
```

```
    DBMS_OUTPUT.PUT_LINE(test3);
```

```
END IF;
```

```
END;
```



9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

-- 9. Pentru un campionat dat ca parametru, afișați numele jucatorului cu cele

-- mai multe goluri înscrise.

```
CREATE OR REPLACE PROCEDURE top_goalscorer(c_id NUMBER) IS
  CURSOR all_goalscorers IS
    SELECT JUCATOR.ume_jucator jucator
      FROM JUCATOR, ECHIPA, DIVIZIE, STATISTICI_INDIVIDUALE
     WHERE JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =
           DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id AND
           JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici
           AND STATISTICI_INDIVIDUALE.numar goluri =
              (SELECT MAX(STATISTICI_INDIVIDUALE.numar goluri)
               FROM STATISTICI_INDIVIDUALE, JUCATOR, ECHIPA, DIVIZIE
              WHERE JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici
                AND JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =
                  DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id);
  ume_j JUCATOR.ume_jucator%TYPE;
  ume_tara CAMPIONAT.tara%TYPE;
BEGIN
  SELECT JUCATOR.ume_jucator, CAMPIONAT.tara
    INTO ume_j, ume_tara
      FROM JUCATOR, ECHIPA, DIVIZIE, CAMPIONAT, STATISTICI_INDIVIDUALE
     WHERE JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =
           DIVIZIE.id divizie AND DIVIZIE.id campionat = CAMPIONAT.id campionat AND
           CAMPIONAT.id campionat = c_id AND
           JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici
           AND STATISTICI_INDIVIDUALE.numar goluri =
              (SELECT MAX(STATISTICI_INDIVIDUALE.numar goluri)
```



```
FROM STATISTICI_INDIVIDUALE, JUCATOR, ECHIPA, DIVIZIE

WHERE JUCATOR.id_statistici = STATISTICI_INDIVIDUALE.id_statistici

AND JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =

DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id);

DBMS_OUTPUT.PUT_LINE('In campionatul tarii ' || nume_tara || ' golgheterul este '

|| nume_j);

EXCEPTION

WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Campionat inexistent');

WHEN TOO_MANY_ROWS THEN

SELECT CAMPIONAT.tara

INTO nume_tara

FROM CAMPIONAT

WHERE CAMPIONAT.id_campionat = c_id;

DBMS_OUTPUT.PUT('In campionatul tarii ' || nume_tara ||

' golgheterii sunt: ');

FOR i in all_goalscorers LOOP

DBMS_OUTPUT.PUT(i.jucator || ' ');

END LOOP;

DBMS_OUTPUT.NEW_LINE;

END;

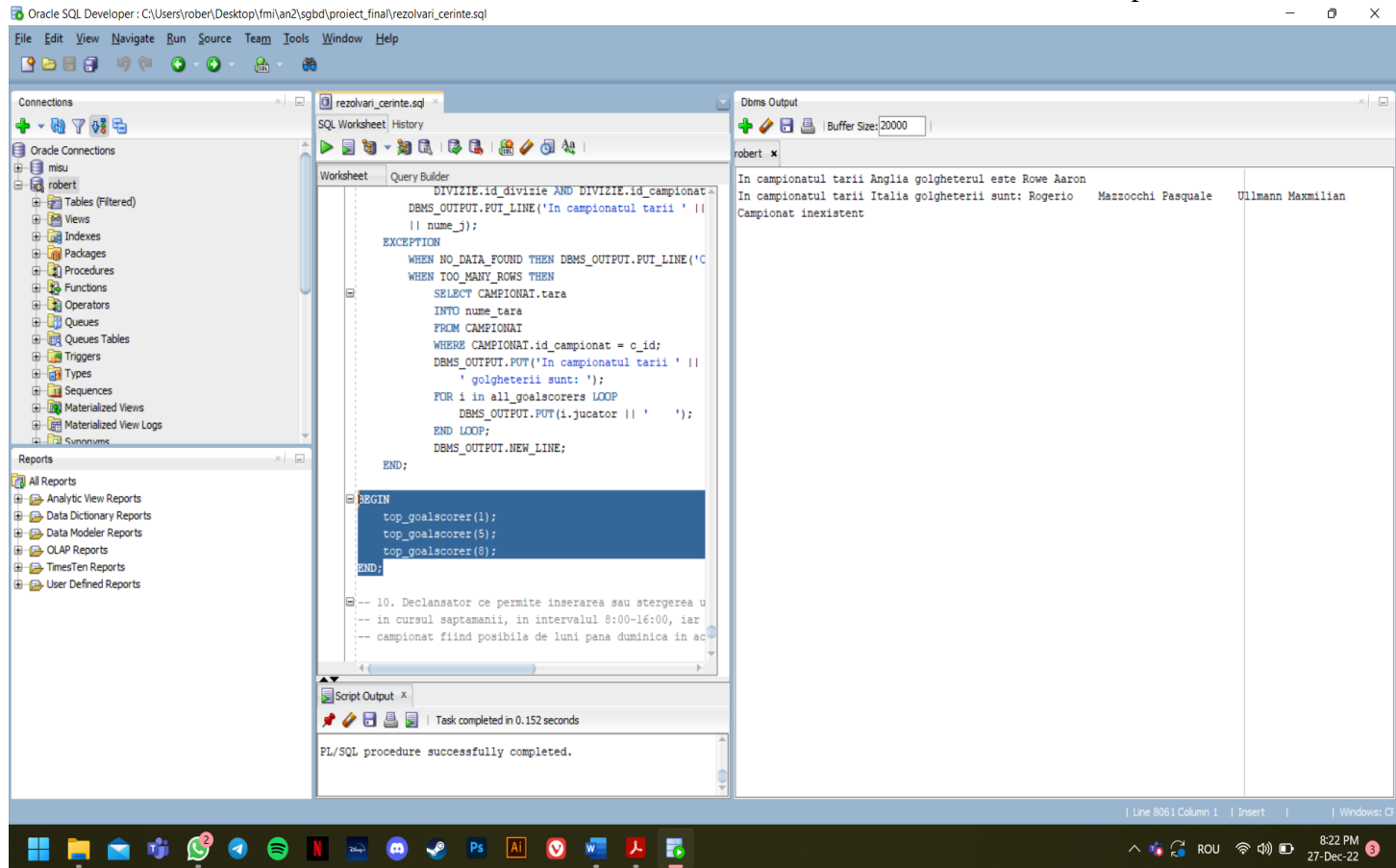
BEGIN

top_goalscorer(1);

top_goalscorer(5);

top_goalscorer(8);

END;
```



10. Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

-- 10. Declansator ce permite inserarea sau stergerea unui nou campionat doar
-- in cursul saptamanii, in intervalul 8:00-16:00, iar modificarea unui
-- campionat fiind posibila de luni pana duminica in acelasi interval orar

CREATE OR REPLACE TRIGGER campionat_scheduler

BEFORE DELETE OR UPDATE OR INSERT ON CAMPIONAT

BEGIN

IF (INSERTING OR DELETING) AND ((TO_CHAR(SYSDATE,'D') = 7 OR

TO_CHAR(SYSDATE,'D') = 1) OR TO_CHAR(SYSDATE,'HH24')

NOT BETWEEN 8 AND 16) THEN

RAISE_APPLICATION_ERROR(-20001, 'Nu se pot insera/sterge elemente

din tabela in aceasta perioada. Consultati orarul.');

ELSIF UPDATING AND (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 8 AND 16) THEN

```
        RAISE_APPLICATION_ERROR(-20002, 'Nu se pot modifica date din tabela in  
        aceasta perioada. Consultati orarul.');
```

END IF;

END;

BEGIN

DELETE FROM CAMPIONAT WHERE id_campionat = 1;

INSERT INTO CAMPIONAT(id_campionat, tara, an_infiintare_campionat, rang) VALUES(8, 'China', 1954, 56);

UPDATE CAMPIONAT SET an_infiintare_campionat = 1950 WHERE id_campionat = 1;

END;

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Script' tab. The script contains the following code:

```
END IF;  
  
BEGIN  
DELETE FROM CAMPIONAT WHERE id_campionat = 1;  
END;  
  
BEGIN  
INSERT INTO CAMPIONAT(id_campionat, tara, an_infiintare_campionat, rang) VALUES(8, 'China', 1954, 56)  
END;  
  
BEGIN  
UPDATE CAMPIONAT SET an_infiintare_campionat = 1950 WHERE id_campionat = 1;  
END;  
  
-- 11. Trigger care nu permite schimbarea echipei unui portar daca actuala  
-- echipa nu are un alt portar sa il inlocuiasca.
```

The 'Script Output' window shows the following error message:

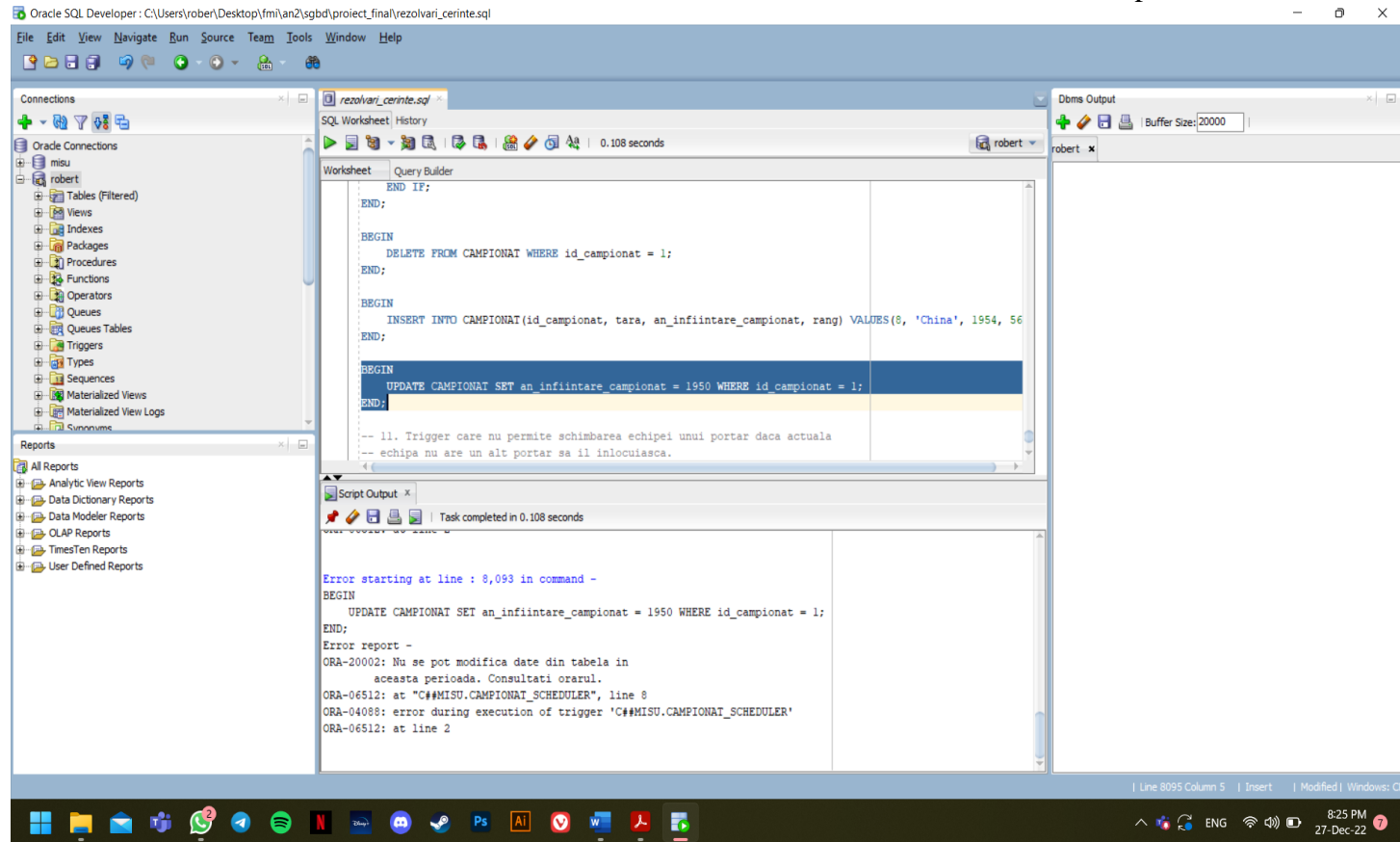
```
Error starting at line : 8,088 in command -  
BEGIN  
DELETE FROM CAMPIONAT WHERE id_campionat = 1;  
END;  
Error report -  
ORA-20001: Nu se pot insera/sterge elemente  
din tabela in aceasta perioada. Consultati orarul.  
ORA-06512: at "C##MISU.CAMPIONAT_SCHEDULER", line 5  
ORA-04088: error during execution of trigger 'C##MISU.CAMPIONAT_SCHEDULER'  
ORA-06512: at line 2
```

The screenshot shows the Oracle SQL Developer interface. The main window displays a PL/SQL script in the 'Script' tab. The script contains the following code:

```
END IF;  
  
BEGIN  
DELETE FROM CAMPIONAT WHERE id_campionat = 1;  
END;  
  
BEGIN  
INSERT INTO CAMPIONAT(id_campionat, tara, an_infiintare_campionat, rang) VALUES(8, 'China', 1954, 56)  
END;  
  
BEGIN  
UPDATE CAMPIONAT SET an_infiintare_campionat = 1950 WHERE id_campionat = 1;  
END;  
  
-- 11. Trigger care nu permite schimbarea echipei unui portar daca actuala  
-- echipa nu are un alt portar sa il inlocuiasca.
```

The 'Script Output' window shows the following error message:

```
Error starting at line : 8,089 in command -  
BEGIN  
INSERT INTO CAMPIONAT(id_campionat, tara, an_infiintare_campionat, rang) VALUES(8, 'China', 1954, 56);  
END;  
Error report -  
ORA-20001: Nu se pot insera/sterge elemente  
din tabela in aceasta perioada. Consultati orarul.  
ORA-06512: at "C##MISU.CAMPIONAT_SCHEDULER", line 5  
ORA-04088: error during execution of trigger 'C##MISU.CAMPIONAT_SCHEDULER'  
ORA-06512: at line 2
```



11. Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

-- 11. Trigger care nu permite schimbarea echipei unui portar daca actuala

-- echipa nu are un alt portar sa il inlocuiasca.

CREATE OR REPLACE TRIGGER gk_transfer

BEFORE UPDATE OF id echipa ON JUCATOR

FOR EACH ROW

WHEN (OLD.pozitie = 'portar')

DECLARE

pragma autonomous_transaction;

x NUMBER;

BEGIN

SELECT count(id_jucator)

INTO x

FROM JUCATOR

```
WHERE id echipa = :OLD.id echipa AND pozitie = 'portar';
```

```
IF x = 1 THEN
```

```
RAISE_APPLICATION_ERROR(-20003, 'Transferul nu poate fi  
realizat deoarece echipa ar ramane fara portar.');
```

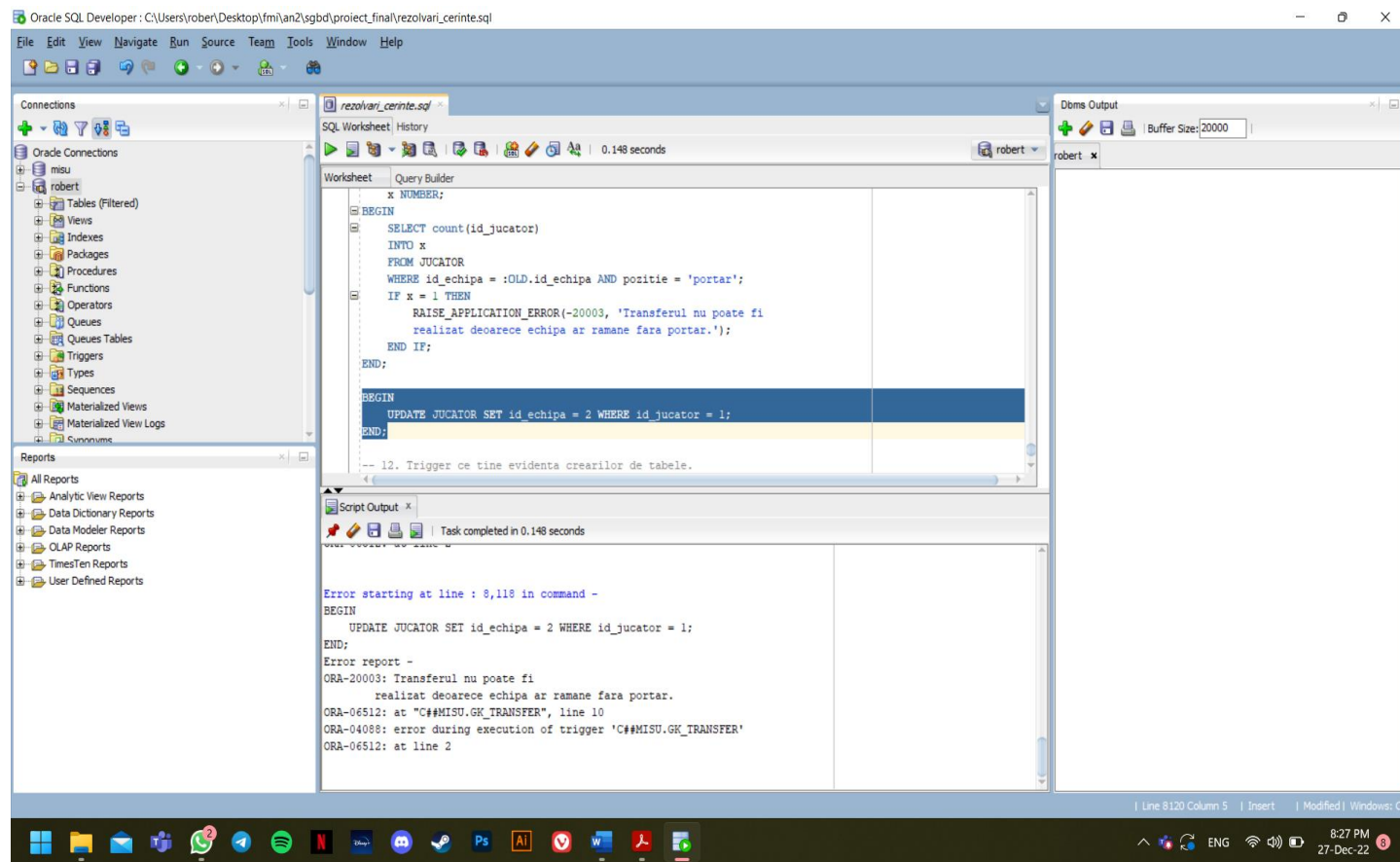
```
END IF;
```

```
END;
```

```
BEGIN
```

```
UPDATE JUCATOR SET id echipa = 2 WHERE id jucator = 1;
```

```
END;
```



12. Definiți un trigger de tip LDD. Declanșați trigger-ul.

-- 12. Trigger ce tine evidenta crearii de tabele.

```
create table table_creation_log(
```

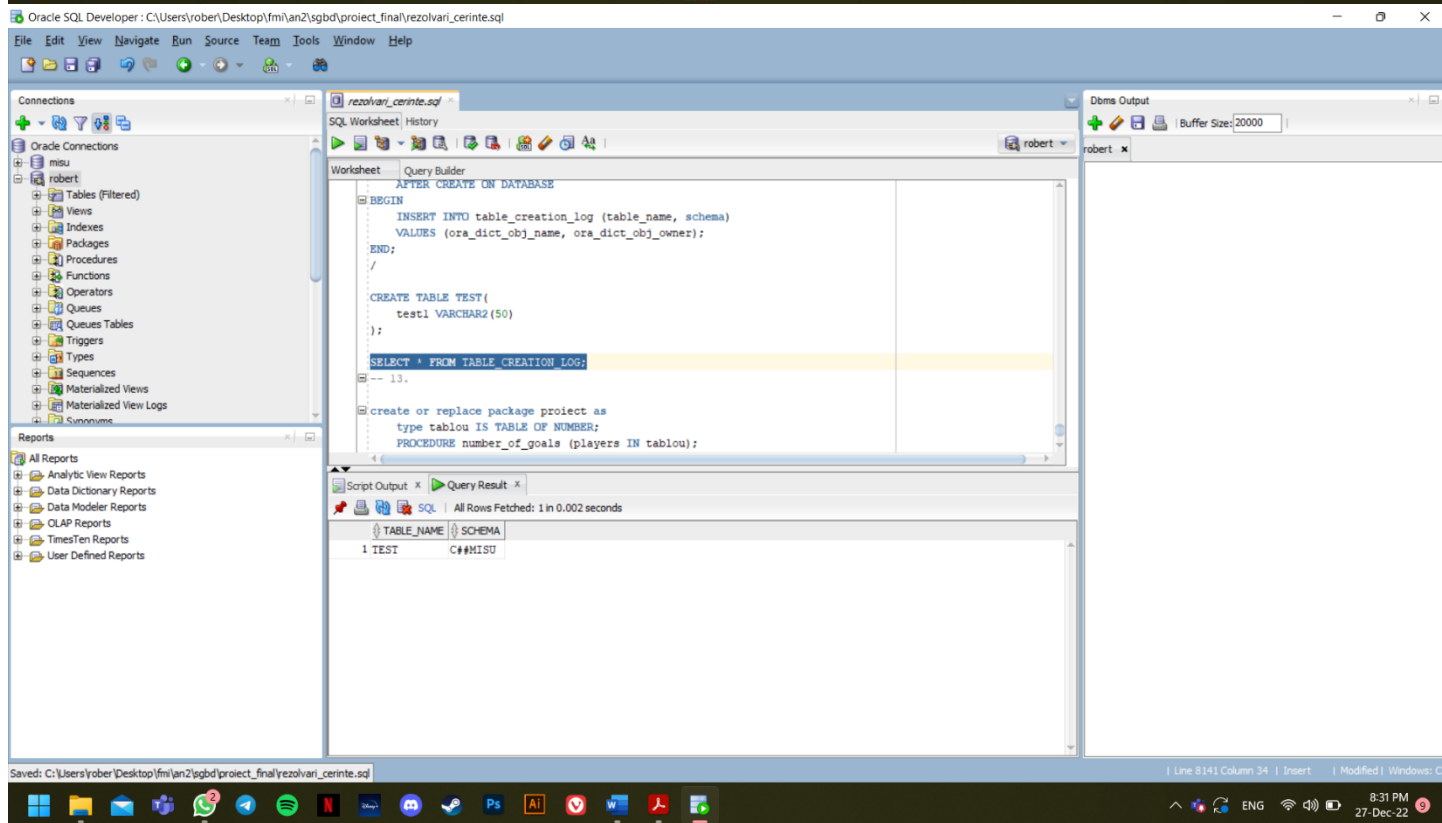
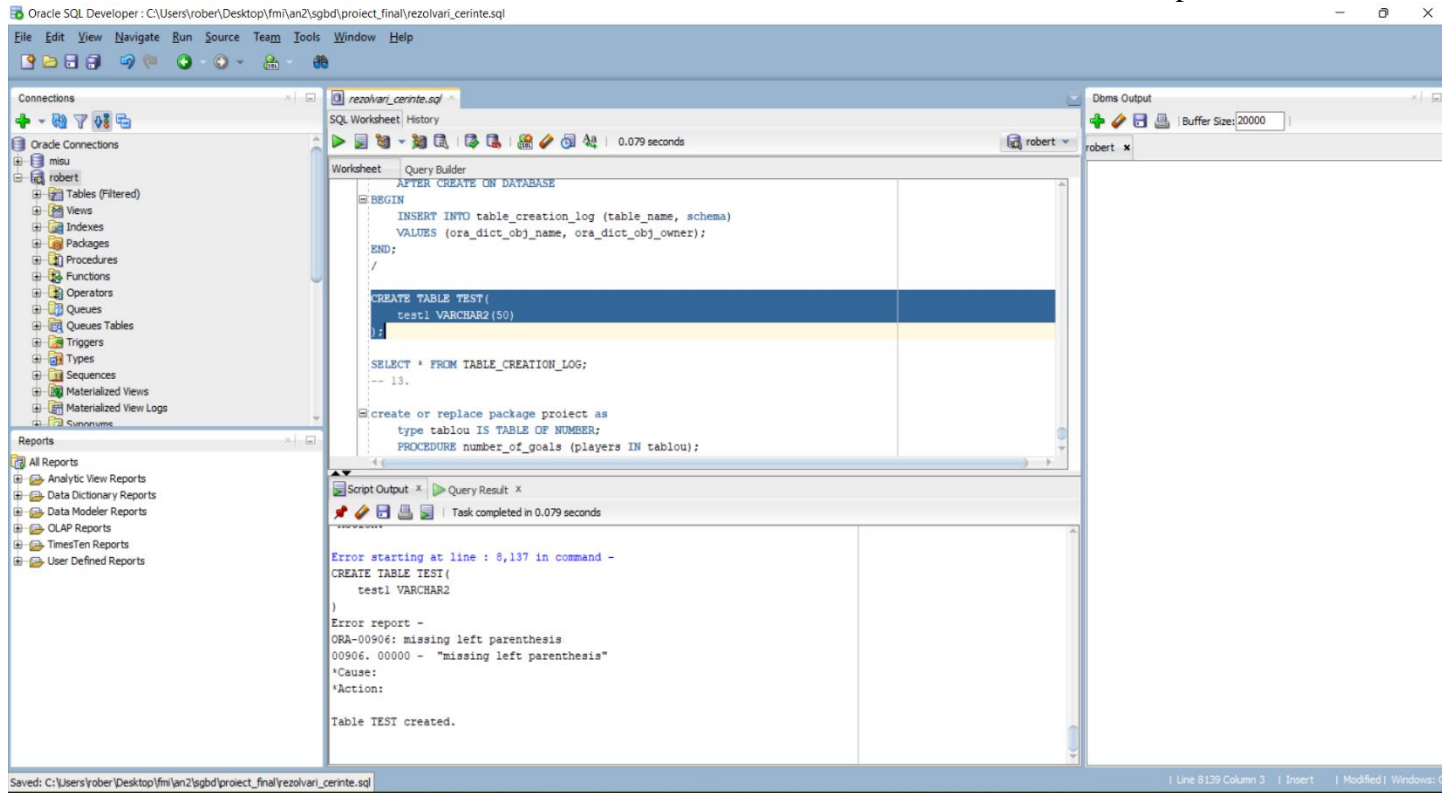
```
table_name VARCHAR2(250),
```

);

```
CREATE OR REPLACE TRIGGER log_alter_campionat
  AFTER CREATE ON DATABASE
BEGIN
  INSERT INTO table_creation_log (table_name, schema)
  VALUES (ora_dict_obj_name, ora_dict_obj_owner);
END;
/
```

```
CREATE TABLE TEST(
  test1 VARCHAR2
);
```

```
SELECT * FROM TABLE_CREATION_LOG;
```



13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

create or replace package proiect as

```
type tablou IS TABLE OF NUMBER;  
PROCEDURE number_of_goals (players IN tablou);  
PROCEDURE contributii;  
FUNCTION match_max_goals(a_id NUMBER) RETURN VARCHAR2;  
PROCEDURE top_goalscorer(c_id NUMBER);
```

END proiect;

/

create or replace package body proiect as

```
PROCEDURE number_of_goals (players IN tablou) IS  
    gol NUMBER;  
    type vector is TABLE OF NUMBER INDEX BY PLS_INTEGER;  
    goals vector;  
BEGIN  
    FOR i in players.FIRST..players.LAST LOOP  
        SELECT STATISTICI_INDIVIDUALE.numar_goluri  
        INTO gol  
        FROM STATISTICI_INDIVIDUALE, JUCATOR  
        WHERE JUCATOR.id_jucator = players(i) AND  
        STATISTICI_INDIVIDUALE.id_statistici = JUCATOR.id_statistici;  
        goals(i) := gol;  
    END LOOP;  
    FOR i in goals.FIRST..goals.LAST LOOP  
        DBMS_OUTPUT.PUT_LINE('Jucatorul cu ID ' || players(i) || ' a inscris '  
        || goals(i) || ' goluri.');    END LOOP;  
END number_of_goals;
```

```
PROCEDURE contributii IS  
    CURSOR prem_players IS
```

```
SELECT JUCATOR.id_jucator id
FROM JUCATOR, ECHIPA
WHERE JUCATOR.id_echipa = ECHIPA.id_echipa AND ECHIPA.id_divizie = 1;

CURSOR contributie(j_id NUMBER) IS

SELECT STATISTICI_INDIVIDUALE.numar_goluri + STATISTICI_INDIVIDUALE.numar_assisturi
FROM STATISTICI_INDIVIDUALE, JUCATOR
WHERE STATISTICI_INDIVIDUALE.id_statistici = JUCATOR.id_statistici AND
      JUCATOR.id_jucator = j_id;

g_a NUMBER;

BEGIN

FOR p_id IN prem_players LOOP

OPEN contributie(p_id.id);

LOOP

FETCH contributie INTO g_a;

EXIT WHEN contributie%NOTFOUND;

DBMS_OUTPUT.PUT_LINE(p_id.id || ' cu contributia ' || g_a);

END LOOP;

CLOSE contributie;

END LOOP;

END contributii;
```

```
FUNCTION match_max_goals(a_id NUMBER) RETURN VARCHAR2 IS

stadium VARCHAR2(50);

BEGIN

SELECT STADION.ume_stadion
INTO stadium
FROM STADION, ARBITRU, PARTIDA, ARBITRU_PARTIDA
WHERE ARBITRU.id_arbitru = a_id AND ARBITRU_PARTIDA.id_partida =
PARTIDA.id_partida AND ARBITRU.id_arbitru = ARBITRU_PARTIDA.id_arbitru
AND STADION.id_stadion = PARTIDA.id_stadion AND
PARTIDA.nr_goluri_gazda + PARTIDA.nr_goluri_oaspeti = (
SELECT max(PARTIDA.nr_goluri_gazda + PARTIDA.nr_goluri_oaspeti)
FROM PARTIDA, ARBITRU_PARTIDA, ARBITRU
WHERE ARBITRU.id_arbitru = a_id AND PARTIDA.id_partida =
```

```
        ARBITRU_PARTIDA.id_partida AND ARBITRU.id_arbitru =  
        ARBITRU_PARTIDA.id_arbitru);  
  
    RETURN stadium;  
  
EXCEPTION  
  
    WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Nici o partida arbitrata'); return null;  
  
    WHEN TOO_MANY_ROWS THEN DBMS_OUTPUT.PUT_LINE('Mai multe meciuri cu aceleasi goluri'); return null;  
  
END match_max_goals;  
  
  
PROCEDURE top_goalscorer(c_id NUMBER) IS  
  
    CURSOR all_goalscorers IS  
  
        SELECT JUCATOR.ume_jucator jucator  
  
            FROM JUCATOR, ECHIPA, DIVIZIE, STATISTICI_INDIVIDUALE  
  
            WHERE JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =  
  
            DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id AND  
  
            JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici  
  
            AND STATISTICI_INDIVIDUALE.numar_goluri =  
  
                (SELECT MAX(STATISTICI_INDIVIDUALE.numar_goluri)  
  
                 FROM STATISTICI_INDIVIDUALE, JUCATOR, ECHIPA, DIVIZIE  
  
                 WHERE JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici  
  
                 AND JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =  
  
                 DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id);  
  
    ume_j JUCATOR.ume_jucator%TYPE;  
  
    ume_tara CAMPIONAT.tara%TYPE;  
  
BEGIN  
  
    SELECT JUCATOR.ume_jucator, CAMPIONAT.tara  
  
    INTO ume_j, ume_tara  
  
    FROM JUCATOR, ECHIPA, DIVIZIE, CAMPIONAT, STATISTICI_INDIVIDUALE  
  
    WHERE JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =  
  
    DIVIZIE.id divizie AND DIVIZIE.id campionat = CAMPIONAT.id campionat AND  
  
    CAMPIONAT.id campionat = c_id AND  
  
    JUCATOR.id statistici = STATISTICI_INDIVIDUALE.id statistici  
  
    AND STATISTICI_INDIVIDUALE.numar_goluri =  
  
        (SELECT MAX(STATISTICI_INDIVIDUALE.numar_goluri)  
  
         FROM STATISTICI_INDIVIDUALE, JUCATOR, ECHIPA, DIVIZIE
```

```
WHERE JUCATOR.id_statistici = STATISTICI_INDIVIDUALE.id_statistici

AND JUCATOR.id echipa = ECHIPA.id echipa AND ECHIPA.id divizie =

DIVIZIE.id divizie AND DIVIZIE.id campionat = c_id);

DBMS_OUTPUT.PUT_LINE('In campionatul tarii ' || nume_tara || ' golgheterul este '

|| nume_j);

EXCEPTION

WHEN NO_DATA_FOUND THEN DBMS_OUTPUT.PUT_LINE('Campionat inexistent');

WHEN TOO_MANY_ROWS THEN

SELECT CAMPIONAT.tara

INTO nume_tara

FROM CAMPIONAT

WHERE CAMPIONAT.id_campionat = c_id;

DBMS_OUTPUT.PUT('In campionatul tarii ' || nume_tara ||

' golgheterii sunt: ');

FOR i in all_goalscorers LOOP

DBMS_OUTPUT.PUT(i.jucator || ' ');

END LOOP;

DBMS_OUTPUT.NEW_LINE;

END top_goalscorer;

END proiect;

BEGIN

proiect.contributii();

END;
```

Oracle SQL Developer : C:\Users\rober\Desktop\fm\an2\sgbd\proiect_final\rezolvari_cerinte.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

- misu
- robert
- Tables (Filtered)
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queue Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Synonyms

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

rezolvari_cerinte.sql

SQL Worksheet History

0.073 seconds

robert

Worksheet

```
WHEN TOO_MANY_ROWS THEN
SELECT CAMPIONAT.tara
INTO nume_tara
FROM CAMPIONAT
WHERE CAMPIONAT.id_campionat = c_id;
DBMS_OUTPUT.PUT('In campionatul tarii ' || nume_tara ||
' golgheterii sunt: ');
FOR i in all_goalscorers LOOP
DBMS_OUTPUT.PUT(i.jucator || ' ');
END LOOP;
DBMS_OUTPUT.NEW_LINE;
END top_goalscorer;
END proiect;

BEGIN
proiect.contributii();
END;
```

Script Output

Task completed in 0.073 seconds

```
CREATE TABLE TEST (
test1 VARCHAR2
)
Error report -
ORA-00906: missing left parenthesis
00906. 00000 - "missing left parenthesis"
*Cause:
*Action:

Table TEST created.

PL/SQL procedure successfully completed.
```

DBMS Output

robert

Buffer Size: 20000

1	cu	contributia	3
2	cu	contributia	29
3	cu	contributia	71
4	cu	contributia	66
5	cu	contributia	0
6	cu	contributia	42
7	cu	contributia	32
8	cu	contributia	15
9	cu	contributia	5
10	cu	contributia	52
11	cu	contributia	31
12	cu	contributia	16
13	cu	contributia	1
14	cu	contributia	44
15	cu	contributia	38
16	cu	contributia	19
17	cu	contributia	5
18	cu	contributia	35
19	cu	contributia	42
20	cu	contributia	63
21	cu	contributia	4
22	cu	contributia	37
23	cu	contributia	61
24	cu	contributia	32
25	cu	contributia	3
26	cu	contributia	22
27	cu	contributia	0
28	cu	contributia	62
29	cu	contributia	4
30	cu	contributia	20
31	cu	contributia	41
32	cu	contributia	43
33	cu	contributia	5
34	cu	contributia	45
35	cu	contributia	15
36	cu	contributia	33
37	cu	contributia	2

Saved: C:\Users\rober\Desktop\fm\an2\sgbd\proiect_final\rezolvari_cerinte.sql

Line 8266 Column 5 | Insert | Windows: C

8:33 PM
27-Dec-22