Sisteme de Gestiune a Bazelor de Date

# Gestiunea turneului de fotbal al ASMI

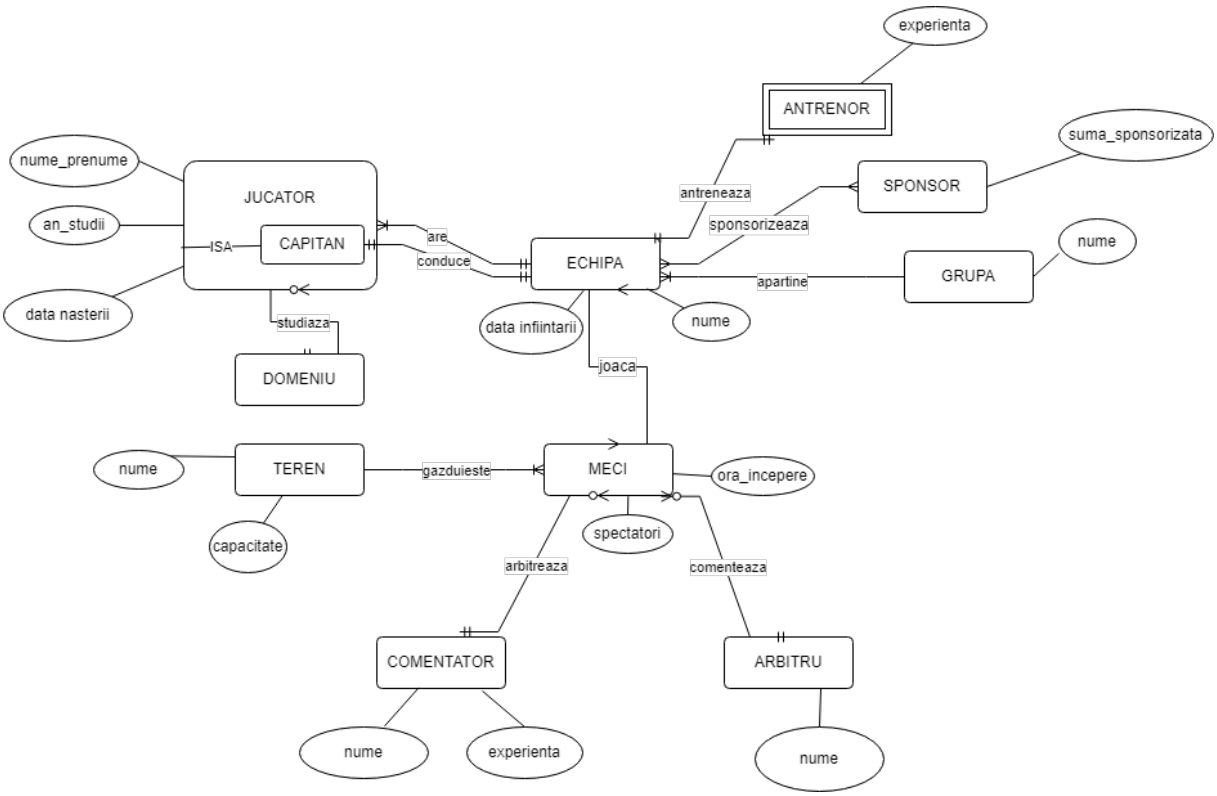
Pascu Ionuț Alexandru

Grupa 252

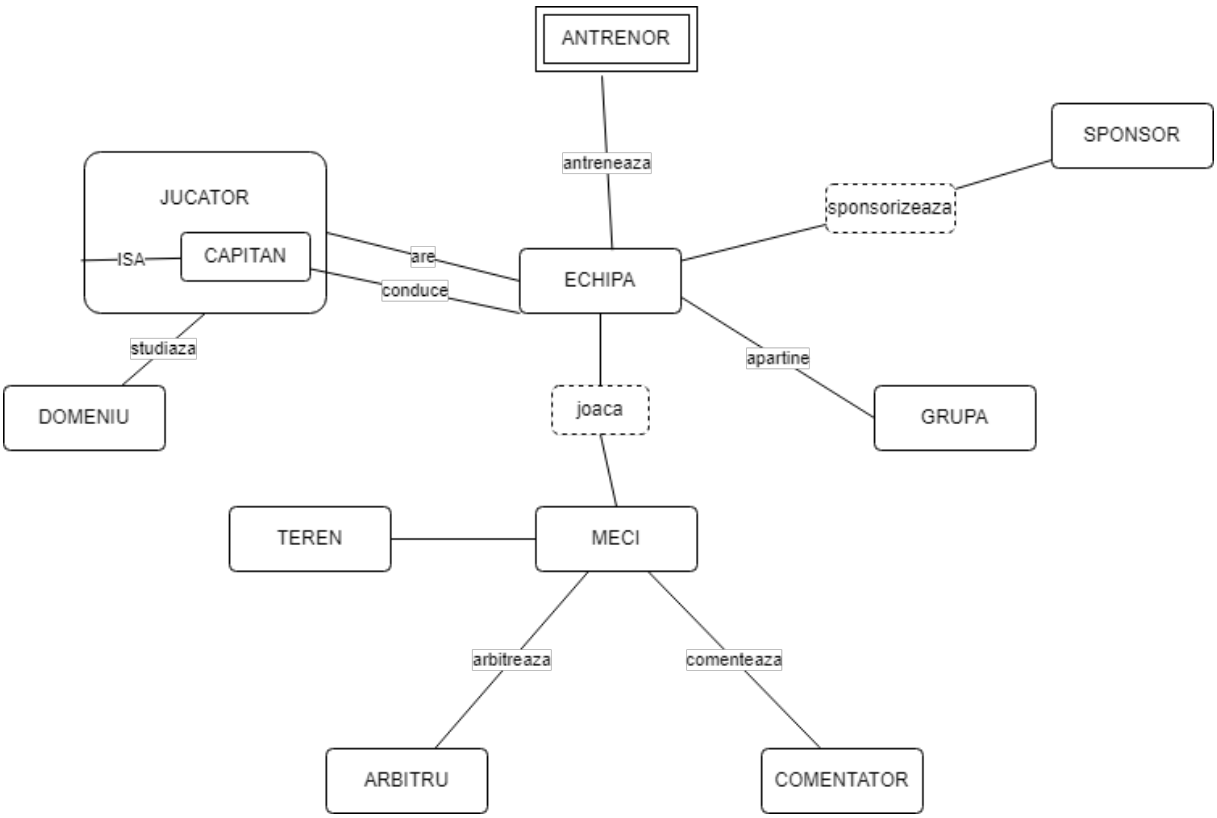
1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.

În fiecare an, Asociația Studenților la Matematică și Informatică organizează un turneu de fotbal la care se pot înscrie studenți din toată Facultatea de Matematică-Informatică. La turneu participă mai multe echipe formate din studenți (jucători). Echipele sunt împărțite în mai multe grupe. În fiecare grupă se desfășoară mai multe meciuri între echipele din acea grupă. Meciurile se joacă pe mai multe terenuri și au o oră de început. Fiecare meci este arbitrat de un arbitru și are un comentator. Fiecare echipă are un antrenor și un căpitan. Fiecare echipă poate avea mai mulți sponsori. Fiecare jucător este student la o anumit domeniu de licență.

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, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```
DROP TABLE echipa;
CREATE TABLE echipa(
    team_id number(5) PRIMARY KEY,
    nume_echipa varchar2(30) NOT NULL,
    data_infiintare date default '01-01-2022',
    group_id number(5) NOT NULL
);

DROP TABLE sponsor;
CREATE TABLE sponsor(
    sponsor_id number(5) PRIMARY KEY,
    nume_sponsor varchar2(30) NOT NULL,
    suma_sponsorizata number(8) NOT NULL
);

DROP TABLE jucator;
CREATE TABLE jucator(
    player_id number(5) PRIMARY KEY,
    nume_prenume varchar2(30) NOT NULL,
    an_studii number(3),
    domeniu_id number(5),
    data_nasterii date NOT NULL,
    team_id number(5) NOT NULL
```

```
);

DROP TABLE sponsorizeaza;
CREATE TABLE sponsorizeaza(
    team_id number(5) NOT NULL,
    sponsor_id number(5) NOT NULL,
    CONSTRAINT sponsorizeaza_id PRIMARY KEY(team_id, sponsor_id)
);


DROP TABLE joaca;
CREATE TABLE joaca(
    team_id number(5) NOT NULL,
    match_id number(5) NOT NULL,
    CONSTRAINT joaca_id PRIMARY KEY(team_id, match_id)
);

DROP TABLE antrenor;
CREATE TABLE antrenor(
    coach_id number(5) PRIMARY KEY,
    team_id number(5) NOT NULL,
    experienta number(3) NOT NULL,
    nume varchar2(30) NOT NULL,
    CONSTRAINT team_fk FOREIGN KEY (team_id) REFERENCES echipa(team_id)
);

DROP TABLE meci;
CREATE TABLE meci(
    match_id number(5) PRIMARY KEY,
    ora_incepere varchar(10) NOT NULL,
    spectatori number(10) NOT NULL,
    commentator_id number(5) NOT NULL,
    referee_id number(5) NOT NULL,
    field_id number(5) NOT NULL
);

DROP TABLE arbitru;
CREATE TABLE arbitru(
    referee_id number(5) PRIMARY KEY,
    nume varchar2(30) NOT NULL,
    varsta number(3) NOT NULL
);

DROP TABLE comentator;
CREATE TABLE comentator(
    commentator_id number(5) PRIMARY KEY,
    nume varchar2(30) NOT NULL,
    experienta number(3) NOT NULL
);

DROP TABLE teren;
CREATE TABLE teren(
    field_id number(5) PRIMARY KEY,
    nume varchar2(30) NOT NULL,
    capacitate number(10) NOT NULL
);

DROP TABLE grupa;
CREATE TABLE grupa(
```

```
    group_id number(5) PRIMARY KEY,
    nume varchar2(30) NOT NULL
);

DROP TABLE domeniu;
CREATE TABLE domeniu(
    domeniu_id number(5) PRIMARY KEY,
    nume varchar2(30) NOT NULL
);

DROP TABLE capitan;
CREATE TABLE capitan(
    player_id number(5) NOT NULL REFERENCES jucator(player_id),
    team_id number(5) not null
);
```

5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```
INSERT INTO GRUPA VALUES (1, 'A');
INSERT INTO GRUPA VALUES (2, 'B');
INSERT INTO GRUPA VALUES (3, 'C');
INSERT INTO GRUPA VALUES (4, 'D');
INSERT INTO GRUPA VALUES (5, 'E');
-- select * from GRUPA;


CREATE SEQUENCE id_echipa_seq
START WITH 1
INCREMENT BY 1;

INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'FMI United', TO_DATE('06-09-
2019'), 1);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'Warriors FC', TO_DATE('30-03-
2020'), 2);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'Winners Club', TO_DATE('04-04-
2021'), 1);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'FC ASMI', TO_DATE('01-01-2020'),
2);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'Real FMI', TO_DATE('24-10-
2020'), 1);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'Stiinta FMI', TO_DATE('15-09-
2018'), 2);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'UniBuc FC', TO_DATE('01-10-
2017'), 1);
INSERT INTO echipa VALUES(id_echipa_seq.nextval, 'FC FMI', TO_DATE('01-10-2017'),
2);
-- SELECT * FROM echipa ORDER BY GROUP_ID;


INSERT INTO sponsor VALUES(1, 'Red Bull', 1000);
INSERT INTO sponsor VALUES(2, 'PlayStation', 2500);
INSERT INTO sponsor VALUES(3, 'Coca Cola', 1500);
INSERT INTO sponsor VALUES(4, 'Tesla', 1200);
INSERT INTO sponsor VALUES(5, 'Audi', 2000);
INSERT INTO SPONSOR VALUES (10, 'Samsung', 1000);
-- SELECT * FROM sponsor ORDER BY suma_sponsorizata;
```

```sql
INSERT INTO DOMENIU VALUES(1, 'Informatica');
INSERT INTO DOMENIU VALUES(2, 'Matematica');
INSERT INTO DOMENIU VALUES(3, 'Mate-info');
INSERT INTO DOMENIU VALUES(4, 'Fizica');
INSERT INTO DOMENIU VALUES(5, 'Chimie');
-- SELECT * FROM DOMENIU;


INSERT INTO jucator VALUES(1, 'Alex Pascu', 2, 1, TO_DATE('30-03-2002'), 1);
INSERT INTO jucator VALUES(2, 'Cristiano Ronaldo', 3, 1, TO_DATE('05-02-1985'),
1);
INSERT INTO jucator VALUES(3, 'Lionel Messi', 2, 3, TO_DATE('24-06-1987'), 2);
INSERT INTO jucator VALUES(4, 'Karim Benzema', 2, 1, TO_DATE('19-12-1987'), 3);
INSERT INTO jucator VALUES(5, 'Kylian Mbappe', 2, 1, TO_DATE('20-12-1998'), 4);
INSERT INTO jucator VALUES(6, 'Toni Kroos', 1, 2, TO_DATE('12-03-1994'), 2);
INSERT INTO jucator VALUES(7, 'Neymar Jr', 3, 2, TO_DATE('05-02-1992'), 4);
INSERT INTO jucator VALUES(8, 'Luka Modric', 2, 3, TO_DATE('10-06-2000'), 6);
INSERT INTO jucator VALUES(9, 'Vinicius Jr', 1, 2, TO_DATE('22-11-1999'), 7);
INSERT INTO jucator VALUES(10, 'Andrei Ivan', 3, 2, TO_DATE('08-01-1994'), 5);
INSERT INTO jucator VALUES(11, 'Robert Lewandowski', 2, 2, TO_DATE('21-08-1988'),
8);
INSERT INTO jucator VALUES(12, 'Paul Pogba', 1, 3, TO_DATE('15-03-1993'), 7);
INSERT INTO jucator VALUES(13, 'Sergio Ramos', 1, 1, TO_DATE('30-03-1986'), 5);
INSERT INTO jucator VALUES(14, 'Eden Hazard', 1, 2, TO_DATE('07-01-1991'), 8);
INSERT INTO jucator VALUES(16, 'Luis Suarez', 1, 1, TO_DATE('24-01-1987'), 7);
INSERT INTO jucator VALUES(17, 'Antoine Griezmann', 2, 3, TO_DATE('21-03-1991'),
6);
INSERT INTO jucator VALUES(18, 'Kevin De Bruyne', 1, 3, TO_DATE('28-06-1991'), 2);
INSERT INTO jucator VALUES(19, 'Harry Kane', 2, 2, TO_DATE('28-07-1993'), 4);
INSERT INTO jucator VALUES(20, 'Sadio Mane', 3, 2, TO_DATE('10-04-1992'), 3);
INSERT INTO jucator VALUES(21, 'Robert Trifan', 2, 1, TO_DATE('15-06-1992'), 1);
INSERT INTO jucator VALUES(22, 'Andrei Murica', 2, 1, TO_DATE('08-12-1994'), 1);
-- SELECT * FROM jucator ORDER BY TEAM_ID;


INSERT INTO CAPITAN VALUES(1, 1);
INSERT INTO CAPITAN VALUES(3, 2);
INSERT INTO CAPITAN VALUES(4, 3);
INSERT INTO CAPITAN VALUES(7, 4);
INSERT INTO CAPITAN VALUES(13, 5);
INSERT INTO CAPITAN VALUES(8, 6);
INSERT INTO CAPITAN VALUES(12, 7);
INSERT INTO CAPITAN VALUES(11, 8);
-- SELECT * FROM CAPITAN ORDER BY TEAM_ID;

INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(1, 5);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(1, 2);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(2, 1);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(2, 4);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(3, 5);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(3, 2);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(4, 2);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(4, 1);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(5, 2);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(5, 4);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(6, 1);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(6, 3);
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(7, 3);
```

```sql
INSERT INTO sponsorizeaza (team_id, sponsor_id) VALUES(7, 4);
-- SELECT * FROM sponsorizeaza;

INSERT INTO COMENTATOR VALUES(1, 'Alexandru Popescu', 1);
INSERT INTO COMENTATOR VALUES(2, 'Mihai Stoica', 2);
INSERT INTO COMENTATOR VALUES(3, 'Andrei Ionescu', 3);
INSERT INTO COMENTATOR VALUES(4, 'Cristian Ionescu', 3);
INSERT INTO COMENTATOR VALUES(5, 'Mihai Popescu', 1);
-- SELECT * FROM comentator;

INSERT INTO ARBITRU VALUES(1, 'Mihai Manea', 30);
INSERT INTO ARBITRU VALUES(2, 'Andrei Stamate', 23);
INSERT INTO ARBITRU VALUES(3, 'Cristian Ciobanu', 34);
INSERT INTO ARBITRU VALUES(4, 'Alexandru Gheorghe', 27);
INSERT INTO ARBITRU VALUES(5, 'Iustin Stoica', 20);
-- SELECT * FROM arbitru;

INSERT INTO TEREN VALUES(1, 'Stadionul National', 5000);
INSERT INTO TEREN VALUES(2, 'Stadionul Olimpic', 4000);
INSERT INTO TEREN VALUES(3, 'Arena Leilor', 8000);
INSERT INTO TEREN VALUES(4, 'Herastrau Park', 6400);
INSERT INTO TEREN VALUES(5, 'Cotroceni Stadium', 7500);
-- SELECT * FROM teren ORDER BY capacitate;

INSERT into MECI VALUES(1, '16:00', 7200, 1, 3, 3);
INSERT into MECI VALUES(2, '18:30', 4500, 4, 2, 4);
INSERT into MECI VALUES(3, '15:00', 5700, 5, 3, 5);
INSERT into MECI VALUES(4, '19:00', 5000, 4, 2, 1);
INSERT into MECI VALUES(5, '17:00', 3700, 3, 1, 2);
INSERT into MECI VALUES(6, '20:00', 5000, 2, 4, 1);
INSERT into MECI VALUES(7, '16:30', 3800, 2, 5, 4);
INSERT into MECI VALUES(8, '18:00', 7000, 1, 4, 5);
-- SELECT * FROM meci ORDER BY ora_inceput;

INSERT INTO joaca (match_id, team_id) VALUES(1, 1);
INSERT INTO joaca (match_id, team_id) VALUES(1, 3);
INSERT INTO joaca (match_id, team_id) VALUES(2, 4);
INSERT INTO joaca (match_id, team_id) VALUES(2, 2);
INSERT INTO joaca (match_id, team_id) VALUES(3, 5);
INSERT INTO joaca (match_id, team_id) VALUES(3, 7);
INSERT INTO joaca (match_id, team_id) VALUES(4, 8);
INSERT INTO joaca (match_id, team_id) VALUES(4, 6);
INSERT INTO joaca (match_id, team_id) VALUES(5, 7);
INSERT INTO joaca (match_id, team_id) VALUES(5, 1);
INSERT INTO joaca (match_id, team_id) VALUES(6, 3);
INSERT INTO joaca (match_id, team_id) VALUES(6, 5);
INSERT INTO joaca (match_id, team_id) VALUES(7, 2);
INSERT INTO joaca (match_id, team_id) VALUES(7, 8);
INSERT INTO joaca (match_id, team_id) VALUES(8, 6);
INSERT INTO joaca (match_id, team_id) VALUES(8, 4);
-- SELECT * FROM joaca ORDER BY match_id;

INSERT INTO antrenor VALUES(1, 1, 3, 'Carlo Ancelotti');
INSERT INTO antrenor VALUES(2, 2, 1, 'Pep Guardiola');
INSERT INTO antrenor VALUES(3, 3, 2, 'Dan Petrescu');
INSERT INTO antrenor VALUES(4, 4, 2, 'Zinedine Zidane');
INSERT INTO antrenor VALUES(5, 5, 2, 'Laurentiu Reghecampf');
INSERT INTO antrenor VALUES(6, 6, 3, 'Gigi Becali');
INSERT INTO antrenor VALUES(7, 7, 1, 'Mihai Rotaru');
```

```
INSERT INTO antrenor VALUES(8, 8, 2, 'Andrei Pavel');
-- SELECT * FROM antrenor;
COMMIT;
ROLLBACK;
```

| | GROUP_ID | NUME |
|---|---|---|
| 1 | 3 | C |
| 2 | 4 | D |
| 3 | 5 | E |
| 4 | 1 | A |
| 5 | 2 | B |

| | TEAM_ID | NUME_ECHIPA | DATA_INFIINTARE | GROUP_ID |
|---|---|---|---|---|
| 1 | 1 | FMI United | 2019-09-06 | 1 |
| 2 | 2 | Warriors FC | 2020-03-30 | 2 |
| 3 | 3 | Winners Club | 2021-04-04 | 1 |
| 4 | 4 | FC ASMI | 2020-01-01 | 2 |
| 5 | 5 | Real FMI | 2020-10-24 | 1 |
| 6 | 6 | Stiinta FMI | 2018-09-15 | 2 |
| 7 | 7 | UniBuc FC | 2017-10-01 | 1 |
| 8 | 8 | FC FMI | 2017-10-01 | 2 |

| | SPONSOR_ID | NUME_SPONSOR | SUMA_SPONSORIZATA |
|---|---|---|---|
| 1 | 1 | Red Bull | 1000 |
| 2 | 2 | PlayStation | 2500 |
| 3 | 3 | Coca Cola | 1500 |
| 4 | 4 | Tesla | 1200 |
| 5 | 5 | Audi | 2000 |
| 6 | 10 | Samsung | 1000 |

| | DOMENIU_ID | NUME |
|---|---|---|
| 1 | 1 | Informatica |
| 2 | 2 | Matematica |
| 3 | 3 | Mate-info |
| 4 | 4 | Fizica |
| 5 | 5 | Chimie |

| | PLAYER_ID | NUME_PRENUME | AN_STUDII | DOMENIU_ID | DATA_NASTERII | TEAM_ID |
|---|---|---|---|---|---|---|
| 1 | 1 | Alex Pascu | 2 | 1 | 2002-03-30 | 1 |
| 2 | 22 | Andrei Murica | 2 | 1 | 1994-12-08 | 1 |
| 3 | 2 | Cristiano Ronaldo | 3 | 1 | 1985-02-05 | 1 |
| 4 | 21 | Robert Trifan | 2 | 1 | 1992-06-15 | 1 |
| 5 | 3 | Lionel Messi | 2 | 3 | 1987-06-24 | 2 |
| 6 | 6 | Toni Kroos | 1 | 2 | 1994-03-12 | 2 |
| 7 | 18 | Kevin De Bruyne | 1 | 3 | 1991-06-28 | 2 |
| 8 | 4 | Karim Benzema | 2 | 1 | 1987-12-19 | 3 |
| 9 | 20 | Sadio Mane | 3 | 2 | 1992-04-10 | 3 |
| 10 | 7 | Neymar Jr | 3 | 2 | 1992-02-05 | 4 |
| 11 | 5 | Kylian Mbappe | 2 | 1 | 1998-12-20 | 4 |
| 12 | 19 | Harry Kane | 2 | 2 | 1993-07-28 | 4 |
| 13 | 10 | Andrei Ivan | 3 | 2 | 1994-01-08 | 5 |
| 14 | 13 | Sergio Ramos | 1 | 1 | 1986-03-30 | 5 |
| 15 | 8 | Luka Modric | 2 | 3 | 2000-06-10 | 6 |
| 16 | 17 | Antoine Griezmann | 2 | 3 | 1991-03-21 | 6 |
| 17 | 12 | Paul Pogba | 1 | 3 | 1993-03-15 | 7 |
| 18 | 16 | Luis Suarez | 1 | 1 | 1987-01-24 | 7 |
| 19 | 9 | Vinicius Jr | 1 | 2 | 1999-11-22 | 7 |
| 20 | 11 | Robert Lewandowski | 2 | 2 | 1988-08-21 | 8 |
| 21 | 14 | Eden Hazard | 1 | 2 | 1991-01-07 | 8 |

| | PLAYER_ID | TEAM_ID |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 3 | 2 |
| 3 | 4 | 3 |
| 4 | 7 | 4 |
| 5 | 13 | 5 |
| 6 | 8 | 6 |
| 7 | 12 | 7 |
| 8 | 11 | 8 |

| | TEAM_ID | SPONSOR_ID |
|---|---|---|
| 1 | 1 | 2 |
| 2 | 1 | 5 |
| 3 | 2 | 1 |
| 4 | 2 | 4 |
| 5 | 3 | 2 |
| 6 | 3 | 5 |
| 7 | 4 | 1 |
| 8 | 4 | 2 |
| 9 | 5 | 2 |
| 10 | 5 | 4 |
| 11 | 6 | 1 |
| 12 | 6 | 3 |
| 13 | 7 | 3 |
| 14 | 7 | 4 |

| | COMMENTATOR_ID | NUME | EXPERIENTA |
|---|---|---|---|
| 1 | 1 | Alexandru Popescu | 1 |
| 2 | 2 | Mihai Stoica | 2 |
| 3 | 3 | Andrei Ionescu | 3 |
| 4 | 4 | Cristian Ionescu | 3 |
| 5 | 5 | Mihai Popescu | 1 |

| | REFEREE_ID | NUME | VARSTA |
|---|---|---|---|
| 1 | 1 | Mihai Manea | 30 |
| 2 | 2 | Andrei Stamate | 23 |
| 3 | 3 | Cristian Ciobanu | 34 |
| 4 | 4 | Alexandru Gheorghe | 27 |
| 5 | 5 | Iustin Stoica | 20 |

| | FIELD_ID | NUME | CAPACITATE |
|---|---|---|---|
| 1 | 1 | Stadionul National | 5000 |
| 2 | 2 | Stadionul Olimpic | 4000 |
| 3 | 3 | Arena Leilor | 8000 |
| 4 | 4 | Herastrau Park | 6400 |
| 5 | 5 | Cotroceni Stadium | 7500 |

| | MATCH_ID | ORA_INCEPUT | SPECTATORI | COMMENTATOR_ID | REFEREE_ID | FIELD_ID |
|---|---|---|---|---|---|---|
| 1 | 1 | 16:00 | 7200 | 1 | 3 | 3 |
| 2 | 2 | 18:30 | 4500 | 4 | 2 | 4 |
| 3 | 3 | 15:00 | 5700 | 5 | 3 | 5 |
| 4 | 4 | 19:00 | 5000 | 4 | 2 | 1 |
| 5 | 5 | 17:00 | 3700 | 3 | 1 | 2 |
| 6 | 6 | 20:00 | 5000 | 2 | 4 | 1 |
| 7 | 7 | 16:30 | 3800 | 2 | 5 | 4 |
| 8 | 8 | 18:00 | 7000 | 1 | 4 | 5 |

| | TEAM_ID | MATCH_ID |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 3 | 1 |
| 3 | 2 | 2 |
| 4 | 4 | 2 |
| 5 | 7 | 3 |
| 6 | 5 | 3 |
| 7 | 8 | 4 |
| 8 | 6 | 4 |
| 9 | 1 | 5 |
| 10 | 7 | 5 |
| 11 | 3 | 6 |
| 12 | 5 | 6 |
| 13 | 2 | 7 |
| 14 | 8 | 7 |
| 15 | 6 | 8 |
| 16 | 4 | 8 |

| | COACH_ID | TEAM_ID | EXPERIENTA | NUME |
|---|---|---|---|---|
| 1 | 1 | 1 | 3 | Carlo Ancelotti |
| 2 | 2 | 2 | 1 | Pep Guardiola |
| 3 | 3 | 3 | 2 | Dan Petrescu |
| 4 | 4 | 4 | 2 | Zinedine Zidane |
| 5 | 5 | 5 | 2 | Laurentiu Reghecampf |
| 6 | 6 | 6 | 3 | Gigi Becali |
| 7 | 7 | 7 | 1 | Mihai Rotaru |
| 8 | 8 | 8 | 2 | Andrei Pavel |

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

```
-- exercitiul 6
-- Afisati toti jucatorii care participa la turneu, apoi doar jucatorii care
-- fac parte din echipa cu un id dat
CREATE OR REPLACE PROCEDURE get_players_from_team(id_echipa IN NUMBER)
IS
    TYPE jucatori is varray(100) of jucator%rowtype;
    v_jucatori jucatori;
    v_jucator jucator%rowtype;
    TYPE toti_jucatorii is table of jucator%rowtype;
    v_toti_jucatorii toti_jucatorii;
    ECHIPA_NEGASITA EXCEPTION;
BEGIN
    SELECT * BULK COLLECT INTO v_toti_jucatorii from JUCATOR;
    DBMS_OUTPUT.PUT_LINE('Toti jucatorii:');
    DBMS_OUTPUT.PUT_LINE('----------------');
```

```
    FOR i in 1..v_toti_jucatorii.COUNT loop
        v_jucator := v_toti_jucatorii(i);
        DBMS_OUTPUT.PUT_LINE(v_jucator.NUME_PRENUME);
    end loop;
    SELECT * bulk collect into v_jucatori FROM JUCATOR WHERE TEAM_ID = id_echipa;
    IF v_jucatori.COUNT = 0 THEN
        RAISE ECHIPA_NEGASITA;
    end if;
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE('Jucatorii din echipa cu id-ul ' || id_echipa || ':');
    DBMS_OUTPUT.PUT_LINE('----------------');
    FOR i IN 1 .. v_jucatori.COUNT LOOP
        v_jucator := v_jucatori(i);
        DBMS_OUTPUT.PUT_LINE(v_jucator.NUME_PRENUME);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('----------------');

    EXCEPTION
        WHEN ECHIPA_NEGASITA THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista echipa cu id-ul ' || id_echipa);

END;
-- apelarea procedurii
BEGIN
  get_players_from_team(1);
  get_players_from_team(152);
END;
```

```
Toti jucatorii:
---------------
Alex Pascu
Cristiano Ronaldo
Lionel Messi
Karim Benzema
Kylian Mbappe
Toni Kroos
Neymar Jr
Luka Modric
Vinicius Jr
Andrei Ivan
Robert Lewandowski
Paul Pogba
Sergio Ramos
Eden Hazard
Luis Suarez
Antoine Griezmann
Kevin De Bruyne
Harry Kane
Sadio Mane
Robert Trifan
Andrei Murica
---------------
Jucatorii din echipa cu id-ul 1:
---------------
Alex Pascu
Cristiano Ronaldo
Robert Trifan
Andrei Murica
```

```
Toti jucatorii:
---------------
Alex Pascu
Cristiano Ronaldo
Lionel Messi
Karim Benzema
Kylian Mbappe
Toni Kroos
Neymar Jr
Luka Modric
Vinicius Jr
Andrei Ivan
Robert Lewandowski
Paul Pogba
Sergio Ramos
Eden Hazard
Luis Suarez
Antoine Griezmann
Kevin De Bruyne
Harry Kane
Sadio Mane
Robert Trifan
Andrei Murica
Nu exista echipa cu id-ul 152
```

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

```
-- exercitiul 7
-- Afisati echipele dintr-o grupa data, apoi capitanii tuturor echipelor
-- apoi afisati antrenorii echipelor din grupa respectiva cu mai mult/mai putin de
2 ani experienta (daca p_cursor = 1, mai mult, daca p_cursor = 2, mai putin)
CREATE OR REPLACE PROCEDURE get_teams_captains_coaches(p_grupa IN GRUPA.NUME%type,
p_cursor IN NUMBER)
AS
  cursor c_teams (p_grupa IN GRUPA.NUME%type) is
    select * from echipa e
    join grupa g on e.GROUP_ID = g.GROUP_ID
```

```
    where g.nume = p_grupa;
   cursor c_captains is
    select * from jucator j
    join capitan c on c.PLAYER_ID = j.PLAYER_ID;

    v_echipa c_teams%rowtype;
    v_capitan c_captains%rowtype;
    TYPE tip_cursor IS REF CURSOR RETURN antrenor%ROWTYPE;
    v_coach_cursor tip_cursor;

    v_coach ANTRENOR%rowtype;
    INVALID_INPUT EXCEPTION;
BEGIN
    IF p_cursor > 2 OR p_cursor < 1 THEN
        RAISE INVALID_INPUT;
    end if;
    DBMS_OUTPUT.PUT_LINE('Echipele din grupa ' || p_grupa || ':');
    DBMS_OUTPUT.PUT_LINE('----------------');
    OPEN c_teams(p_grupa);
    LOOP
        FETCH c_teams INTO v_echipa;
        EXIT WHEN c_teams%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_echipa.NUME_ECHIPA);
    END LOOP;
    CLOSE c_teams;
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE('Capitanii tuturor echipelor:');
    DBMS_OUTPUT.PUT_LINE('----------------');
    OPEN c_captains;
    LOOP
        FETCH c_captains INTO v_capitan;
        EXIT WHEN c_captains%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_capitan.nume_prenume);
    END LOOP;
    CLOSE c_captains;

     IF p_cursor = 1 THEN
        OPEN v_coach_cursor FOR
            SELECT * FROM ANTRENOR WHERE EXPERIENTA > 2;
    ELSIF p_cursor = 2 THEN
        OPEN v_coach_cursor FOR
            SELECT * FROM ANTRENOR WHERE EXPERIENTA <= 2;
    END IF;

    DBMS_OUTPUT.PUT_LINE('----------------');
    if p_cursor = 1 then
        DBMS_OUTPUT.PUT_LINE('Antrenorii echipelor cu experienta mai mare de 2
ani:');
    else
        DBMS_OUTPUT.PUT_LINE('Antrenorii echipelor cu experienta mai mica sau
egala cu 2 ani:');
    end if;
    DBMS_OUTPUT.PUT_LINE('----------------');
    LOOP
        FETCH v_coach_cursor INTO v_coach;
        EXIT WHEN v_coach_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_coach.nume);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('----------------');
```

```
    EXCEPTION
        WHEN INVALID_INPUT THEN
            DBMS_OUTPUT.PUT_LINE('Parametri invalizi!');

END;
-- apelarea procedurii
BEGIN
    get_teams_captains_coaches('B', 2);
    get_teams_captains_coaches('A', 1);
    get_teams_captains_coaches('A', 3);
    get_teams_captains_coaches('C', -3);
END;
```

```
Echipele din grupa B:
----------------
Warriors FC
FC ASMI
Stiinta FMI
FC FMI
----------------
Capitanii tuturor echipelor:
----------------
Alex Pascu
Lionel Messi
Karim Benzema
Neymar Jr
Luka Modric
Robert Lewandowski
Paul Pogba
Sergio Ramos
----------------
Antrenorii echipelor cu experienta mai mica sau egala cu 2 ani:
----------------
Pep Guardiola
Dan Petrescu
Zinedine Zidane
Laurentiu Reghecampf
Mihai Rotaru
Andrei Pavel
```

```
Echipele din grupa A:
----------------
FMI United
Winners Club
Real FMI
UniBuc FC
----------------
Capitanii tuturor echipelor:
----------------
Alex Pascu
Lionel Messi
Karim Benzema
Neymar Jr
Luka Modric
Robert Lewandowski
Paul Pogba
Sergio Ramos
----------------
Antrenorii echipelor cu experienta mai mare de 2 ani:
----------------
Carlo Ancelotti
Gigi Becali
----------------
Parametri invalizi!
Parametri invalizi!
```

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.

```sql
-- exercitiul 8
-- Afisati echipa care primeste cei mai multi bani din sponsorizari
-- avand un sponsor dat
CREATE OR REPLACE FUNCTION get_team_with_most_money(p_sponsor_id IN NUMBER) RETURN
VARCHAR2 AS
    v_nume_echipa ECHIPA.nume_echipa%type;
    v_suma NUMBER;
    v_s NUMBER;
    v_cnt NUMBER;
    v_sol ECHIPA.nume_echipa%type;
    v_sponsor_id NUMBER;
    v_cursor SYS_REFCURSOR;
    NUMAR_NEGATIV EXCEPTION;
    FARA_ECHIPE EXCEPTION;
    PREA_MULTE_ECHIPE EXCEPTION;
```

```
BEGIN
    IF p_sponsor_id < 0 THEN
        RAISE NUMAR_NEGATIV;
    END IF;
    SELECT SPONSOR_ID
    INTO v_sponsor_id
    FROM SPONSOR
    WHERE SPONSOR_ID = p_sponsor_id;
    OPEN v_cursor FOR
        SELECT e.NUME_ECHIPA
        FROM ECHIPA e
        JOIN SPONSORIZEAZA ON e.TEAM_ID = SPONSORIZEAZA.TEAM_ID
        JOIN SPONSOR s ON SPONSORIZEAZA.SPONSOR_ID = s.SPONSOR_ID
        WHERE s.SPONSOR_ID = p_sponsor_id;
    v_s := 0;
    v_suma := 0;
    v_cnt := 0;
    LOOP
        FETCH v_cursor INTO v_nume_echipa;
        EXIT WHEN v_cursor%NOTFOUND;
        SELECT SUM(SUMA_SPONSORIZATA)
        INTO v_s
        FROM SPONSORIZEAZA
        JOIN SPONSOR s ON SPONSORIZEAZA.SPONSOR_ID = s.SPONSOR_ID
        WHERE TEAM_ID = (SELECT TEAM_ID FROM ECHIPA WHERE NUME_ECHIPA =
v_nume_echipa);
        IF v_s > v_suma THEN
            v_suma := v_s;
            v_sol := v_nume_echipa;
            v_cnt := 1;
        ELSIF v_s = v_suma THEN
            v_cnt := v_cnt + 1;
        END IF;
    END LOOP;
    close v_cursor;
    IF v_sol IS NULL THEN
        RAISE FARA_ECHIPE;
    END IF;
    IF v_cnt > 1 THEN
        RAISE PREA_MULTE_ECHIPE;
    end if;
    RETURN v_sol;

    EXCEPTION
        WHEN NUMAR_NEGATIV THEN
            DBMS_OUTPUT.PUT_LINE('ID-ul sponsorului nu poate fi negativ!');
            RETURN NULL;
        WHEN FARA_ECHIPE THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista nicio echipa care sa aiba sponsorul cu
ID-ul ' || p_sponsor_id);
            RETURN NULL;
        WHEN PREA_MULTE_ECHIPE THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multe echipe care au acelasi numar de
bani din sponsorizari!');
            RETURN NULL;
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista sponsor cu acest ID!');
            RETURN NULL;
```

```
END;
-- apelarea functiei
BEGIN
    DBMS_OUTPUT.PUT_LINE(get_team_with_most_money(1));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(get_team_with_most_money(2));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(get_team_with_most_money(-3));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(get_team_with_most_money(123));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(get_team_with_most_money(10));
END;
```

```
FC ASMI
----------------
Exista mai multe echipe care au acelasi numar de bani din sponsorizari!
----------------
ID-ul sponsorului nu poate fi negativ!
----------------
Nu exista sponsor cu acest ID!
----------------
Nu exista nicio echipa care sa aiba sponsorul cu ID-ul 10
```

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.

```
-- exercitiul 9
-- Pentru un jucator al carui prenume este dat, afisati terenul cu cea mai mare
capacitate
-- in care a jucat
CREATE OR REPLACE PROCEDURE get_field_with_most_capacity(p_nume_jucator IN
VARCHAR2) AS
    v_nume_teren TEREN.NUME%type;
    v_capacitate TEREN.capacitate%type;
    aux_capacitate TEREN.capacitate%type;
    aux_nume_teren TEREN.NUME%type;
    v_playerid JUCATOR.player_id%type;
    v_init NUMBER := -1;
    CURSOR c_teren IS
        SELECT t.CAPACITATE, t.NUME, JUCATOR.PLAYER_ID
        FROM teren t
        JOIN meci m on m.FIELD_ID = t.FIELD_ID
        JOIN joaca on joaca.MATCH_ID = m.MATCH_ID
        JOIN ECHIPA on ECHIPA.TEAM_ID = joaca.TEAM_ID
        join JUCATOR on JUCATOR.TEAM_ID = ECHIPA.TEAM_ID
        WHERE UPPER(JUCATOR.NUME_PRENUME) LIKE '%' || UPPER(p_nume_jucator) ||
'%';
```

```
    INVALID_INPUT EXCEPTION;
    TYPE_MISMATCH EXCEPTION;
BEGIN
    IF p_nume_jucator IS NULL THEN
        RAISE INVALID_INPUT;
    END IF;
    IF regexp_like(p_nume_jucator, '[0-9]') THEN
        RAISE TYPE_MISMATCH;
    END IF;
   OPEN c_teren;
    v_capacitate := -1;
    LOOP
        FETCH c_teren INTO aux_capacitate, aux_nume_teren, v_playerid;
        EXIT WHEN c_teren%NOTFOUND;
        if v_playerid != v_init and v_init != -1 then
            raise too_many_rows;
        end if;
        if aux_capacitate > v_capacitate then
            v_capacitate := aux_capacitate;
            v_nume_teren := aux_nume_teren;
        end if;
        v_init := v_playerid;
    end loop;
    CLOSE c_teren;
    IF v_capacitate = -1 THEN
        raise no_data_found;
    end if;
    DBMS_OUTPUT.PUT_LINE('Terenul cu cea mai mare capacitate in care a jucat ' ||
p_nume_jucator || ' este ' || v_nume_teren || ' cu o capacitate de ' ||
v_capacitate);

    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista niciun jucator cu numele ' ||
p_nume_jucator);
            RETURN;
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai mult de un jucator cu numele ' ||
p_nume_jucator);
            RETURN;
        WHEN TYPE_MISMATCH THEN
            DBMS_OUTPUT.PUT_LINE('Numele jucatorului trebuie sa fie un sir de
caractere fara cifre!');
            RETURN;
        WHEN INVALID_INPUT THEN
            DBMS_OUTPUT.PUT_LINE('Numele jucatorului nu poate fi NULL!');
            RETURN;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Alta eroare!');

END;
-- apelarea procedurii
BEGIN
    get_field_with_most_capacity('Alex');
    get_field_with_most_capacity('Andrei');
    get_field_with_most_capacity('Obama');
    get_field_with_most_capacity(6);
    get_field_with_most_capacity('ale32');
```

```
    get_field_with_most_capacity('');
END;
```

```
Terenul cu cea mai mare capacitate in care a jucat Alex este Arena Leilor cu o capacitate de 8000
Exista mai mult de un jucator cu numele Andrei
Nu exista niciun jucator cu numele Obama
Numele jucatorului trebuie sa fie un sir de caractere fara cifre!
Numele jucatorului trebuie sa fie un sir de caractere fara cifre!
Numele jucatorului nu poate fi NULL!
```

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

```
-- exercitiul 10
-- Creati un trigger care sa nu permita sa fie mai mult de 16 echipe in turneu
-- intrucat organizarea nu permite mai multe echipe
CREATE OR REPLACE TRIGGER trigger_echipa
    AFTER INSERT ON ECHIPA
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count FROM ECHIPA;
    IF v_count > 16 THEN
        RAISE_APPLICATION_ERROR(-20004, 'Nu se poate insera o noua echipa!');
    END IF;
END;
-- declansarea triggerului
BEGIN
    FOR i in 1..10 LOOP
        INSERT INTO ECHIPA(team_id, nume_echipa, group_id) VALUES(30+i, 'test ' ||
i, 1);
    end loop;
    delete from ECHIPA where DATA_INFIINTARE = '01-01-2022';
END;
```

```
[2023-01-11 23:32:23] [72000][20000]
[2023-01-11 23:32:23]   ORA-20000: Nu se poate insera o noua echipa!
[2023-01-11 23:32:23]   ORA-06512: la "ALEX.TRIGGER_ECHIPA", linia 6
[2023-01-11 23:32:23]   ORA-04088: eroare în timpul execuţiei triggerului 'ALEX.TRIGGER_ECHIPA'
[2023-01-11 23:32:23]   ORA-06512: la linia 3
[2023-01-11 23:32:23] Position: 0
```

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

```
-- exercitiul 11
-- Creati un trigger care sa nu permita modificarea datei de infiintare a unei
echipe
-- (acesta este un camp ce nu trebuie modificat)
CREATE OR REPLACE TRIGGER trigger_data_infiintare
    BEFORE UPDATE OF DATA_INFIINTARE ON ECHIPA
    FOR EACH ROW
    WHEN (NEW.DATA_INFIINTARE != OLD.DATA_INFIINTARE)
BEGIN
    RAISE_APPLICATION_ERROR(-20005, 'Nu se poate modifica data infiintarii!');
```

```
END;
-- declansarea triggerului
BEGIN
    UPDATE ECHIPA SET DATA_INFIINTARE = '01-01-1990' WHERE ECHIPA.TEAM_ID < 3;
END;
```

```
[2023-01-11 23:33:19] [72000][20005]
[2023-01-11 23:33:19]   ORA-20005: Nu se poate modifica data infiintarii!
[2023-01-11 23:33:19]   ORA-06512: la "ALEX.TRIGGER_DATA_INFIINTARE", linia 2
[2023-01-11 23:33:19]   ORA-04088: eroare în timpul execuţiei triggerului 'ALEX.TRIGGER_DATA_INFIINTARE'
[2023-01-11 23:33:19]   ORA-06512: la linia 2
[2023-01-11 23:33:19] Position: 0
```

Pentru exercițiile 10 și 11 am făcut și un trigger compus care rezolvă ambele exerciții deodată.

```
-- exercitiile 10 si 11 combinate
CREATE OR REPLACE TRIGGER trigger_jucator
    FOR INSERT OR UPDATE OR DELETE ON JUCATOR
    COMPOUND TRIGGER
    BEFORE STATEMENT IS
    BEGIN
        IF TO_CHAR(SYSDATE, 'HH24') < '08' OR TO_CHAR(SYSDATE, 'HH24') > '20' THEN
           RAISE_APPLICATION_ERROR(-20002, 'Nu se poate face operatia in afara
intervalului orar 08:00 - 20:00');
       END IF;
    END BEFORE STATEMENT;
    BEFORE EACH ROW IS
    BEGIN
        IF UPDATING THEN
            IF :OLD.NUME_PRENUME != :NEW.NUME_PRENUME THEN
                RAISE_APPLICATION_ERROR(-20003, 'Nu se poate modifica numele
jucatorului!');
            END IF;
        END IF;
    END BEFORE EACH ROW;
END;
-- apelarea triggerului
BEGIN
    INSERT INTO JUCATOR VALUES(50, 'John Cena', 2, 1, '1990-01-01', 1);
    UPDATE JUCATOR SET NUME_PRENUME = 'John Cena' WHERE PLAYER_ID = 3;
END;
```

```
[2023-01-11 23:35:21] [72000][20000]
[2023-01-11 23:35:21]   ORA-20000: Nu se poate face operatia in afara intervalului orar 08:00 - 20:00
[2023-01-11 23:35:21]   ORA-06512: la "ALEX.TRIGGER_JUCATOR", linia 5
[2023-01-11 23:35:21]   ORA-04088: eroare în timpul execuţiei triggerului 'ALEX.TRIGGER_JUCATOR'
[2023-01-11 23:35:21] Position: 36
```

```
[2023-01-11 12:42:11] [72000][20003]
[2023-01-11 12:42:11]   ORA-20003: Nu se poate modifica numele jucatorului!
[2023-01-11 12:42:11]   ORA-06512: la "ALEX.TRIGGER_JUCATOR", linia 12
[2023-01-11 12:42:11]   ORA-04088: eroare în timpul execuţiei triggerului 'ALEX.TRIGGER_JUCATOR'
[2023-01-11 12:42:11] Position: 7
```

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

```
-- exercitiul 12
-- Creati un trigger care sa permita modificarea schemei doar de utilizatorul alex
-- si salvati modificarile facute asupra schemei intr-un tabel
CREATE TABLE log_history
(
    username VARCHAR2(20),
    log_date DATE,
    db_name VARCHAR2(20),
    event VARCHAR2(100),
    obj_name VARCHAR2(100)
);

CREATE OR REPLACE TRIGGER trigger_schema
    AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
   IF USER != 'ALEX' THEN
       RAISE_APPLICATION_ERROR(-20910, 'Numai administratorul bazei de date poate
realiza operatia!');
   END IF;
    INSERT INTO log_history VALUES(SYS.LOGIN_USER, SYSDATE, SYS.DATABASE_NAME,
SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME);
END;

-- declansarea triggerului
CREATE TABLE test (id NUMBER);
DROP TABLE test;
ALTER TABLE JUCATOR ADD (test VARCHAR2(20));
ALTER TABLE JUCATOR DROP COLUMN test;
-- vizualizarea modificarilor
SELECT * FROM log_history;
ROLLBACK;
```

| | | | | | |
|---|---|---|---|---|---|
| 1 | ALEX | 2023-01-10 22:46:24 | XE | CREATE | TEST |
| 2 | ALEX | 2023-01-10 22:46:25 | XE | DROP | TEST |
| 3 | ALEX | 2023-01-10 22:46:25 | XE | ALTER | JUCATOR |
| 4 | ALEX | 2023-01-10 22:46:25 | XE | ALTER | JUCATOR |

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

```
-- exercitiul 13
-- Definiti un pachet care sa contina toate obiectele definite in cadrul
proiectului
CREATE OR REPLACE PACKAGE proiect_AXP AS
    PROCEDURE get_players_from_team(id_echipa IN NUMBER);
    PROCEDURE get_teams_captains_coaches(p_grupa IN GRUPA.NUME%type, p_cursor IN
NUMBER);
    FUNCTION get_team_with_most_money(p_sponsor_id IN NUMBER) RETURN VARCHAR2;
    PROCEDURE get_field_with_most_capacity(p_nume_jucator IN VARCHAR2);
END proiect_AXP;
CREATE OR REPLACE PACKAGE BODY proiect_AXP AS
    -- exercitiul 6
    -- Afisati toti jucatorii care participa la turneu, apoi doar jucatorii care
    -- fac parte din echipa cu un id dat
    PROCEDURE get_players_from_team(id_echipa IN NUMBER)
IS
```

```
    TYPE jucatori is varray(100) of jucator%rowtype;
    v_jucatori jucatori;
    v_jucator jucator%rowtype;
    TYPE toti_jucatorii is table of jucator%rowtype;
    v_toti_jucatorii toti_jucatorii;
    ECHIPA_NEGASITA EXCEPTION;
BEGIN
    SELECT * BULK COLLECT INTO v_toti_jucatorii from JUCATOR;
    DBMS_OUTPUT.PUT_LINE('Toti jucatorii:');
    DBMS_OUTPUT.PUT_LINE('----------------');
    FOR i in 1..v_toti_jucatorii.COUNT loop
        v_jucator := v_toti_jucatorii(i);
        DBMS_OUTPUT.PUT_LINE(v_jucator.NUME_PRENUME);
    end loop;
    SELECT * bulk collect into v_jucatori FROM JUCATOR WHERE TEAM_ID = id_echipa;
    IF v_jucatori.COUNT = 0 THEN
        RAISE ECHIPA_NEGASITA;
    end if;
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE('Jucatorii din echipa cu id-ul ' || id_echipa || ':');
    DBMS_OUTPUT.PUT_LINE('----------------');
    FOR i IN 1 .. v_jucatori.COUNT LOOP
        v_jucator := v_jucatori(i);
        DBMS_OUTPUT.PUT_LINE(v_jucator.NUME_PRENUME);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('----------------');

    EXCEPTION
        WHEN ECHIPA_NEGASITA THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista echipa cu id-ul ' || id_echipa);

END;

    -- exercitiul 7
    -- Afisati echipele dintr-o grupa data, apoi capitanii tuturor echipelor
    -- apoi afisati antrenorii echipelor din grupa respectiva cu mai mult/mai
putin de 2 ani experienta
    PROCEDURE get_teams_captains_coaches(p_grupa IN GRUPA.NUME%type, p_cursor IN
NUMBER)
AS
  cursor c_teams (p_grupa IN GRUPA.NUME%type) is
    select * from echipa e
    join grupa g on e.GROUP_ID = g.GROUP_ID
    where g.nume = p_grupa;
   cursor c_captains is
    select * from jucator j
    join capitan c on c.PLAYER_ID = j.PLAYER_ID;

    v_echipa c_teams%rowtype;
    v_capitan c_captains%rowtype;
    TYPE tip_cursor IS REF CURSOR RETURN antrenor%ROWTYPE;
    v_coach_cursor tip_cursor;

    v_coach ANTRENOR%rowtype;
    INVALID_INPUT EXCEPTION;
BEGIN
    IF p_cursor > 2 OR p_cursor < 1 THEN
        RAISE INVALID_INPUT;
    end if;
```

```
    DBMS_OUTPUT.PUT_LINE('Echipele din grupa ' || p_grupa || ':');
    DBMS_OUTPUT.PUT_LINE('----------------');
    OPEN c_teams(p_grupa);
    LOOP
        FETCH c_teams INTO v_echipa;
        EXIT WHEN c_teams%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_echipa.NUME_ECHIPA);
    END LOOP;
    CLOSE c_teams;
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE('Capitanii tuturor echipelor:');
    DBMS_OUTPUT.PUT_LINE('----------------');
    OPEN c_captains;
    LOOP
        FETCH c_captains INTO v_capitan;
        EXIT WHEN c_captains%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_capitan.nume_prenume);
    END LOOP;
    CLOSE c_captains;

     IF p_cursor = 1 THEN
        OPEN v_coach_cursor FOR
            SELECT * FROM ANTRENOR WHERE EXPERIENTA > 2;
    ELSIF p_cursor = 2 THEN
        OPEN v_coach_cursor FOR
            SELECT * FROM ANTRENOR WHERE EXPERIENTA <= 2;
    END IF;

    DBMS_OUTPUT.PUT_LINE('----------------');
    if p_cursor = 1 then
        DBMS_OUTPUT.PUT_LINE('Antrenorii echipelor cu experienta mai mare de 2
ani:');
    else
        DBMS_OUTPUT.PUT_LINE('Antrenorii echipelor cu experienta mai mica sau
egala cu 2 ani:');
    end if;
    DBMS_OUTPUT.PUT_LINE('----------------');
    LOOP
        FETCH v_coach_cursor INTO v_coach;
        EXIT WHEN v_coach_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(v_coach.nume);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('----------------');

    EXCEPTION
        WHEN INVALID_INPUT THEN
            DBMS_OUTPUT.PUT_LINE('Parametri invalizi!');

END;

    -- exercitiul 8
    -- Afisati echipa care primeste cei mai multi bani din sponsorizari
    -- avand un sponsor dat
    FUNCTION get_team_with_most_money(p_sponsor_id IN NUMBER) RETURN VARCHAR2 AS
    v_nume_echipa ECHIPA.nume_echipa%type;
    v_suma NUMBER;
    v_s NUMBER;
    v_cnt NUMBER;
    v_sol ECHIPA.nume_echipa%type;
```

```
    v_sponsor_id NUMBER;
    v_cursor SYS_REFCURSOR;
    NUMAR_NEGATIV EXCEPTION;
    FARA_ECHIPE EXCEPTION;
    PREA_MULTE_ECHIPE EXCEPTION;
BEGIN
    IF p_sponsor_id < 0 THEN
        RAISE NUMAR_NEGATIV;
    END IF;
    SELECT SPONSOR_ID
    INTO v_sponsor_id
    FROM SPONSOR
    WHERE SPONSOR_ID = p_sponsor_id;
    OPEN v_cursor FOR
        SELECT e.NUME_ECHIPA
        FROM ECHIPA e
        JOIN SPONSORIZEAZA ON e.TEAM_ID = SPONSORIZEAZA.TEAM_ID
        JOIN SPONSOR s ON SPONSORIZEAZA.SPONSOR_ID = s.SPONSOR_ID
        WHERE s.SPONSOR_ID = p_sponsor_id;
    v_s := 0;
    v_suma := 0;
    v_cnt := 0;
    LOOP
        FETCH v_cursor INTO v_nume_echipa;
        EXIT WHEN v_cursor%NOTFOUND;
        SELECT SUM(SUMA_SPONSORIZATA)
        INTO v_s
        FROM SPONSORIZEAZA
        JOIN SPONSOR s ON SPONSORIZEAZA.SPONSOR_ID = s.SPONSOR_ID
        WHERE TEAM_ID = (SELECT TEAM_ID FROM ECHIPA WHERE NUME_ECHIPA =
v_nume_echipa);
        IF v_s > v_suma THEN
            v_suma := v_s;
            v_sol := v_nume_echipa;
            v_cnt := 1;
        ELSIF v_s = v_suma THEN
            v_cnt := v_cnt + 1;
        END IF;
    END LOOP;
    close v_cursor;
    IF v_sol IS NULL THEN
        RAISE FARA_ECHIPE;
    END IF;
    IF v_cnt > 1 THEN
        RAISE PREA_MULTE_ECHIPE;
    end if;
    RETURN v_sol;

    EXCEPTION
        WHEN NUMAR_NEGATIV THEN
            DBMS_OUTPUT.PUT_LINE('ID-ul sponsorului nu poate fi negativ!');
            RETURN NULL;
        WHEN FARA_ECHIPE THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista nicio echipa care sa aiba sponsorul cu
ID-ul ' || p_sponsor_id);
            RETURN NULL;
        WHEN PREA_MULTE_ECHIPE THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multe echipe care au acelasi numar de
bani din sponsorizari!');
```

```
            RETURN NULL;
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista sponsor cu acest ID!');
            RETURN NULL;

END;

    -- exercitiul 9
    -- Pentru un jucator al carui prenume este dat, afisati terenul cu cea mai
mare capacitate
    -- in care a jucat
    PROCEDURE get_field_with_most_capacity(p_nume_jucator IN VARCHAR2) AS
    v_nume_teren TEREN.NUME%type;
    v_capacitate TEREN.capacitate%type;
    aux_capacitate TEREN.capacitate%type;
    aux_nume_teren TEREN.NUME%type;
    v_playerid JUCATOR.player_id%type;
    v_init NUMBER := -1;
    CURSOR c_teren IS
        SELECT t.CAPACITATE, t.NUME, JUCATOR.PLAYER_ID
        FROM teren t
        JOIN meci m on m.FIELD_ID = t.FIELD_ID
        JOIN joaca on joaca.MATCH_ID = m.MATCH_ID
        JOIN ECHIPA on ECHIPA.TEAM_ID = joaca.TEAM_ID
        join JUCATOR on JUCATOR.TEAM_ID = ECHIPA.TEAM_ID
        WHERE UPPER(JUCATOR.NUME_PRENUME) LIKE '%' || UPPER(p_nume_jucator) ||
'%';
    INVALID_INPUT EXCEPTION;
    TYPE_MISMATCH EXCEPTION;
BEGIN
    IF p_nume_jucator IS NULL THEN
        RAISE INVALID_INPUT;
    END IF;
    IF regexp_like(p_nume_jucator, '[0-9]') THEN
        RAISE TYPE_MISMATCH;
    END IF;
   OPEN c_teren;
    v_capacitate := -1;
    LOOP
        FETCH c_teren INTO aux_capacitate, aux_nume_teren, v_playerid;
        EXIT WHEN c_teren%NOTFOUND;
        if v_playerid != v_init and v_init != -1 then
            raise too_many_rows;
        end if;
        if aux_capacitate > v_capacitate then
            v_capacitate := aux_capacitate;
            v_nume_teren := aux_nume_teren;
        end if;
        v_init := v_playerid;
    end loop;
    CLOSE c_teren;
    IF v_capacitate = -1 THEN
        raise no_data_found;
    end if;
    DBMS_OUTPUT.PUT_LINE('Terenul cu cea mai mare capacitate in care a jucat ' ||
p_nume_jucator || ' este ' || v_nume_teren || ' cu o capacitate de ' ||
v_capacitate);

    EXCEPTION
```

```
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista niciun jucator cu numele ' ||
p_nume_jucator);
            RETURN;
        WHEN TOO_MANY_ROWS THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai mult de un jucator cu numele ' ||
p_nume_jucator);
            RETURN;
        WHEN TYPE_MISMATCH THEN
            DBMS_OUTPUT.PUT_LINE('Numele jucatorului trebuie sa fie un sir de
caractere fara cifre!');
            RETURN;
        WHEN INVALID_INPUT THEN
            DBMS_OUTPUT.PUT_LINE('Numele jucatorului nu poate fi NULL!');
            RETURN;
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Alta eroare!');

END;
END proiect_AXP;

-- testare pachet
BEGIN
    proiect_axp.get_players_from_team(1);
    proiect_axp.get_players_from_team(152);
    DBMS_OUTPUT.PUT_LINE('###############');
    proiect_axp.get_teams_captains_coaches('B', 2);
    proiect_axp.get_teams_captains_coaches('A', 1);
    proiect_axp.get_teams_captains_coaches('A', 3);
    proiect_axp.get_teams_captains_coaches('C', -3);
    DBMS_OUTPUT.PUT_LINE('################');
    DBMS_OUTPUT.PUT_LINE(proiect_axp.get_team_with_most_money(1));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(proiect_axp.get_team_with_most_money(2));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(proiect_axp.get_team_with_most_money(-3));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(proiect_axp.get_team_with_most_money(123));
    DBMS_OUTPUT.PUT_LINE('----------------');
    DBMS_OUTPUT.PUT_LINE(proiect_axp.get_team_with_most_money(10));
    DBMS_OUTPUT.PUT_LINE('################');
    proiect_axp.get_field_with_most_capacity('Alex');
    proiect_axp.get_field_with_most_capacity('Andrei');
    proiect_axp.get_field_with_most_capacity('Obama');
    proiect_axp.get_field_with_most_capacity(6);
    proiect_axp.get_field_with_most_capacity('ale32');
    proiect_axp.get_field_with_most_capacity('');
end;
```

```
Toti jucatorii:
----------------
Alex Pascu
Cristiano Ronaldo
Lionel Messi
Karim Benzema
Kylian Mbappe
Toni Kroos
Neymar Jr
Luka Modric
Vinicius Jr
Andrei Ivan
Robert Lewandowski
Paul Pogba
Sergio Ramos
Eden Hazard
Luis Suarez
Antoine Griezmann
Kevin De Bruyne
Harry Kane
Sadio Mane
Robert Trifan
Andrei Murica
----------------
Jucatorii din echipa cu id-ul 1:
----------------
Alex Pascu
Cristiano Ronaldo
Robert Trifan
Andrei Murica
```

```
----------------
Toti jucatorii:
----------------
Alex Pascu
Cristiano Ronaldo
Lionel Messi
Karim Benzema
Kylian Mbappe
Toni Kroos
Neymar Jr
Luka Modric
Vinicius Jr
Andrei Ivan
Robert Lewandowski
Paul Pogba
Sergio Ramos
Eden Hazard
Luis Suarez
Antoine Griezmann
Kevin De Bruyne
Harry Kane
Sadio Mane
Robert Trifan
Andrei Murica
Nu exista echipa cu id-ul 152
###############
Echipele din grupa B:
----------------
Warriors FC
FC ASMI
Stiinta FMI
FC FMI
```

```
----------------
Capitanii tuturor echipelor:
----------------
Alex Pascu
Lionel Messi
Karim Benzema
Neymar Jr
Luka Modric
Robert Lewandowski
Paul Pogba
Sergio Ramos
----------------
Antrenorii echipelor cu experienta mai mica sau egala cu 2 ani:
----------------
Pep Guardiola
Dan Petrescu
Zinedine Zidane
Laurentiu Reghecampf
Mihai Rotaru
Andrei Pavel
----------------
Echipele din grupa A:
----------------
FMI United
Winners Club
Real FMI
UniBuc FC
----------------
```

```
Capitanii tuturor echipelor:
---------------
Alex Pascu
Lionel Messi
Karim Benzema
Neymar Jr
Luka Modric
Robert Lewandowski
Paul Pogba
Sergio Ramos
---------------
Antrenorii echipelor cu experienta mai mare de 2 ani:
---------------
Carlo Ancelotti
Gigi Becali
---------------
Parametri invalizi!
Parametri invalizi!
################
FC ASMI
---------------
Exista mai multe echipe care au acelasi numar de bani din sponsorizari!
---------------
ID-ul sponsorului nu poate fi negativ!
---------------
Nu exista sponsor cu acest ID!
---------------
Nu exista nicio echipa care sa aiba sponsorul cu ID-ul 10
################
```

```
Terenul cu cea mai mare capacitate in care a jucat Alex este Arena Leilor cu o capacitate de 8000
Exista mai mult de un jucator cu numele Andrei
Nu exista niciun jucator cu numele Obama
Numele jucatorului trebuie sa fie un sir de caractere fara cifre!
Numele jucatorului trebuie sa fie un sir de caractere fara cifre!
Numele jucatorului nu poate fi NULL!
```

14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).

```
-- exercitiul 14
-- Definiti un pachet care sa includa tipuri de date complexe si obiecte necesare
-- pentru actiuni integrate
-- Pachetul contine functii si proceduri care permit, fiind date ca parametri
```

```
-- numele unui arbitru, comentator si jucator, afisarea tuturor meciurilor in care
-- au participat, respectiv au comentat, respectiv au jucat persoanele respective
-- si a echipelor care au jucat in acele meciuri
CREATE OR REPLACE PACKAGE proiect_ex14 AS
    TYPE tip_meci IS TABLE OF MECI.match_id%TYPE;
    TYPE tip_echipa is varray(2) of ECHIPA%ROWTYPE;

    FUNCTION get_referee(p_nume_arbitru IN VARCHAR2) RETURN
ARBITRU.referee_id%type;
    FUNCTION get_commentator(p_nume_comentator IN VARCHAR2) RETURN
COMENTATOR.commentator_id%type;
    FUNCTION get_player(p_nume_jucator IN VARCHAR2) RETURN JUCATOR.player_id%type;
    PROCEDURE get_matches(p_nume_arbitru IN VARCHAR2, p_nume_comentator IN
VARCHAR2, p_nume_jucator IN VARCHAR2);
    PROCEDURE print_match_and_teams(p_match_id IN NUMBER, v_echipe IN tip_echipa);
END proiect_ex14;
CREATE OR REPLACE PACKAGE BODY proiect_ex14 AS
    PROCEDURE print_match_and_teams(p_match_id IN NUMBER, v_echipe IN tip_echipa)
AS
        v_meci MECI%rowtype;
    BEGIN
        SELECT * INTO v_meci FROM MECI WHERE MATCH_ID = p_match_id;

        DBMS_OUTPUT.PUT_LINE('Meciul a inceput la ora ' || v_meci.ORA_INCEPUT || '
si a avut un numar de ' || v_meci.SPECTATORI || ' spectatori.');

        DBMS_OUTPUT.PUT_LINE('Meciul a avut loc intre echipele: ' ||
v_echipe(1).NUME_ECHIPA || ' si ' || v_echipe(2).NUME_ECHIPA);
    END print_match_and_teams;

    FUNCTION get_referee(p_nume_arbitru IN VARCHAR2) RETURN
ARBITRU.referee_id%type AS
        v_id ARBITRU.referee_id%type;
        cnt NUMBER;
    BEGIN
        SELECT COUNT(REFEREE_ID) INTO cnt FROM ARBITRU WHERE UPPER(NUME) LIKE '%'
|| UPPER(p_nume_arbitru) || '%';
        IF cnt = 0 THEN
            raise no_data_found;
        ELSIF cnt > 1 THEN
            raise too_many_rows;
        ELSE
            SELECT REFEREE_ID INTO v_id FROM ARBITRU WHERE UPPER(NUME) LIKE '%' ||
UPPER(p_nume_arbitru) || '%';
        END IF;

        RETURN v_id;

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Nu exista niciun arbitru cu numele ' ||
p_nume_arbitru);
                RETURN -1;
            WHEN TOO_MANY_ROWS THEN
                DBMS_OUTPUT.PUT_LINE('Exista mai mult de un arbitru cu numele ' ||
p_nume_arbitru);
                RETURN -1;
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Alta eroare!');
```

```
                RETURN -1;

    end get_referee;

    FUNCTION get_commentator(p_nume_comentator IN VARCHAR2) RETURN
COMENTATOR.commentator_id%type AS
        v_id COMENTATOR.commentator_id%type;
        cnt NUMBER;
    BEGIN
        SELECT COUNT(COMMENTATOR_ID) INTO cnt FROM COMENTATOR WHERE UPPER(NUME)
LIKE '%' || UPPER(p_nume_comentator) || '%';
        IF cnt = 0 THEN
            raise no_data_found;
        ELSIF cnt > 1 THEN
            raise too_many_rows;
        ELSE
            SELECT COMMENTATOR_ID INTO v_id FROM COMENTATOR WHERE UPPER(NUME) LIKE
'%' || UPPER(p_nume_comentator) || '%';
        END IF;

        RETURN v_id;

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Nu exista niciun comentator cu numele ' ||
p_nume_comentator);
                RETURN -1;
            WHEN TOO_MANY_ROWS THEN
                DBMS_OUTPUT.PUT_LINE('Exista mai mult de un comentator cu numele '
|| p_nume_comentator);
                RETURN -1;
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Alta eroare!');
                RETURN -1;
    end get_commentator;

    FUNCTION get_player(p_nume_jucator IN VARCHAR2) RETURN JUCATOR.player_id%type
AS
        v_id JUCATOR.player_id%type;
        cnt NUMBER;
    BEGIN
        SELECT COUNT(PLAYER_ID) INTO cnt FROM JUCATOR WHERE UPPER(NUME_PRENUME)
LIKE '%' || UPPER(p_nume_jucator) || '%';
        IF cnt = 0 THEN
            raise no_data_found;
        ELSIF cnt > 1 THEN
            raise too_many_rows;
        ELSE
            SELECT PLAYER_ID INTO v_id FROM JUCATOR WHERE UPPER(NUME_PRENUME) LIKE
'%' || UPPER(p_nume_jucator) || '%';
        END IF;

        RETURN v_id;

        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Nu exista niciun jucator cu numele ' ||
p_nume_jucator);
                RETURN -1;
```

```
            WHEN TOO_MANY_ROWS THEN
                DBMS_OUTPUT.PUT_LINE('Exista mai mult de un jucator cu numele ' ||
p_nume_jucator);
                RETURN -1;
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('Alta eroare!');
                RETURN -1;
    end get_player;

    PROCEDURE get_matches(p_nume_arbitru IN VARCHAR2, p_nume_comentator IN
VARCHAR2, p_nume_jucator IN VARCHAR2) AS
        v_id_arbitru ARBITRU.referee_id%type;
        v_id_comentator COMENTATOR.commentator_id%type;
        v_id_jucator JUCATOR.player_id%type;
        v_id_echipa ECHIPA.team_id%type;
        v_meciuri tip_meci;
        v_echipe tip_echipa;
        i NUMBER;
    BEGIN
        v_id_arbitru := get_referee(p_nume_arbitru);
        v_id_comentator := get_commentator(p_nume_comentator);
        v_id_jucator := get_player(p_nume_jucator);

        IF v_id_arbitru = -1 OR v_id_comentator = -1 OR v_id_jucator = -1 THEN
            RETURN;
        END IF;

        SELECT ECHIPA.TEAM_ID INTO v_id_echipa
        FROM ECHIPA
        JOIN JUCATOR ON ECHIPA.TEAM_ID = JUCATOR.TEAM_ID
        WHERE JUCATOR.PLAYER_ID = v_id_jucator;

        IF v_id_echipa IS NULL THEN
            DBMS_OUTPUT.PUT_LINE('Jucatorul ' || p_nume_jucator || ' nu face parte
din nici o echipa!');
            RETURN;
        END IF;

        SELECT J.MATCH_ID
        BULK COLLECT INTO v_meciuri
        FROM JOACA J
        JOIN MECI on MECI.MATCH_ID = J.MATCH_ID
        WHERE REFEREE_ID = v_id_arbitru AND COMMENTATOR_ID = v_id_comentator
        AND J.TEAM_ID = v_id_echipa;

        FOR i IN 1..v_meciuri.COUNT LOOP
            SELECT E.*
            BULK COLLECT INTO v_echipe
            FROM JOACA J
            JOIN ECHIPA E ON E.TEAM_ID = J.TEAM_ID
            WHERE J.MATCH_ID = v_meciuri(i);

            print_match_and_teams(v_meciuri(i), v_echipe);

        END LOOP;
    END get_matches;

END;
/
```

```
-- testare pachet
BEGIN
    proiect_ex14.get_matches('Cristian','Alexandru','Alex');
    DBMS_OUTPUT.PUT_LINE('-----------------');
    proiect_ex14.get_matches('Mihai', 'Mihai', 'Mihai');
    DBMS_OUTPUT.PUT_LINE('-----------------');
END;
```

```
Meciul a inceput la ora 16:00 si a avut un numar de 7200 spectatori.
Meciul a avut loc intre echipele: FMI United si Winners Club
-----------------
Exista mai mult de un comentator cu numele Mihai
Nu exista niciun jucator cu numele Mihai
-----------------
```