A Database of Euroleague –Report

Short Description

This database was built with the intent of keeping a record of the matches, teams, players referees and tracking player's performance in the context of the Euroligue Competition. With that in mind 8 tables were created as follows:

```
CREATE TABLE EUROLEAGUE( League_Id Number PRIMARY KEY, Points int ,Goals_Scored int
,Goals_Conceded int ,Difference int ,);
```

```
CREATE TABLE LOG_TEAM( Log_Id Number PRIMARY KEY, Log_Time TIMESTAMP NOT
NULL);
```

```
CREATE TABLE TEAMS(Team_Id Number Primary Key,Team_Name VARCHAR2(30),
Team_Country VARCHAR2(10) NOT NULL, Log_Id Number, CONSTRAINT chk_country
CHECK (TeamCountry IN ('Spain', 'England'))-- C) Country is either Spain or England, FOREIGN
KEY (Log_Id) REFERENCES LOG_TEAM(Log_Id),);
```

```
CREATE TABLE MATCHES( Match_ID NUMBER PRIMARY KEY,-- A) MatchID is an auto
increment key. Using a sequence to define it. Team_1_Id Number NOT NULL, Team_2_Id
Number NOT NULL,Goal__Team_1 VARCHAR2(10) NOT NULL,Goal_Team_2 VARCHAR2(10)
NOT NULL, Competition VARCHAR2(20) NOT NULL, League_Id Number NOT NUll,
CONSTRAINT chk_competition CHECK (Competition IN ('Champions League', 'Europa Leauge',
'Premier League', 'La Liga')), CONSTRAINT chk_goal_A CHECK (Goal_A >= 0 AND Goal_B >= 0),
FOREIGN KEY (Team_1_Id) REFERENCES TEAMS(Team_Name), FOREIGN KEY (Team_2_Id)
REFERENCES TEAMS(Team_Name),FOREIGN KEY (League_Id) REFERENCES
EUROLEAGUE(League_Id),);
```

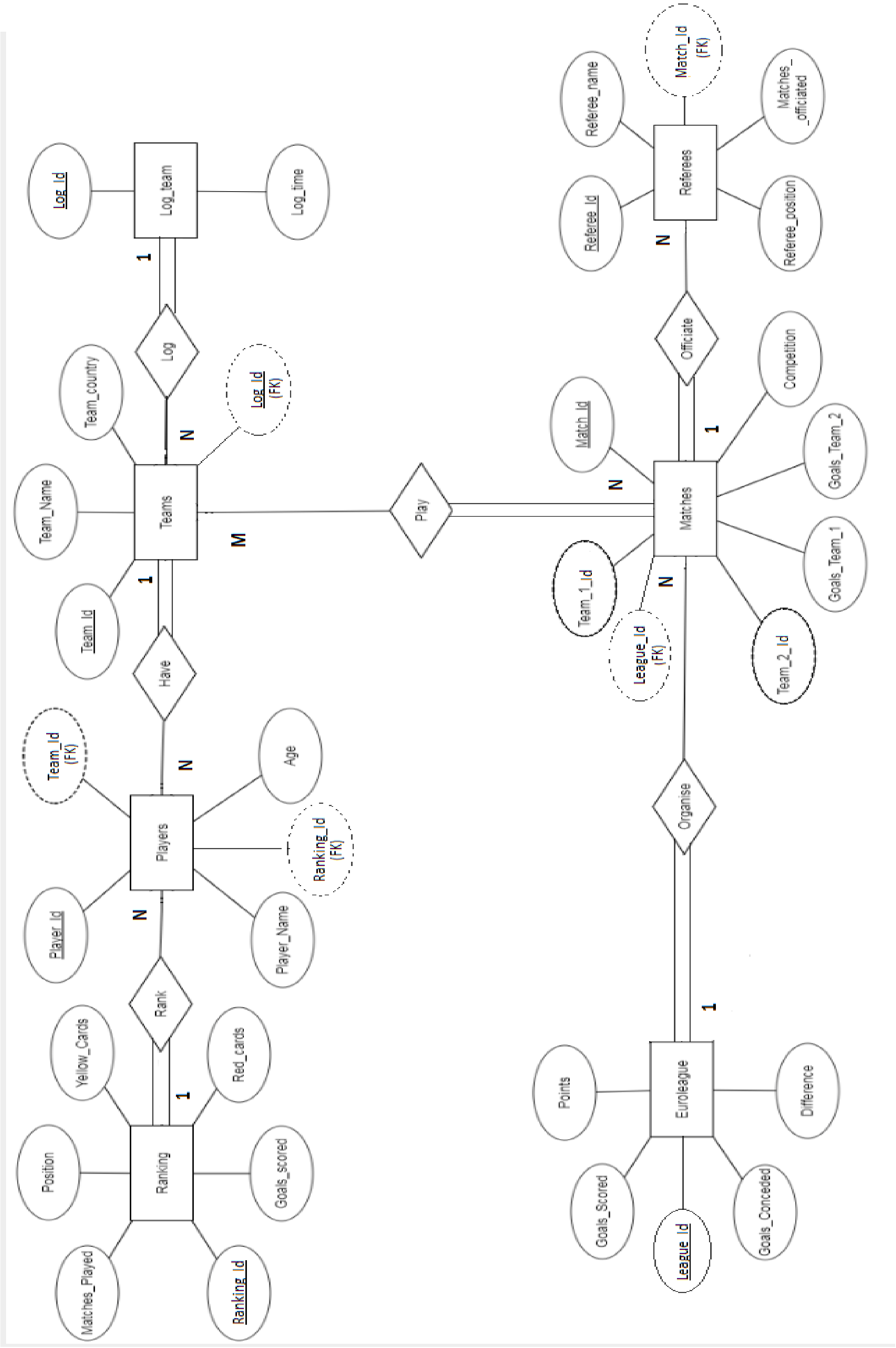
```
 CREATE TABLE Game(Match_Id Number Primary Key,Team_Id Number Primary
Key,FOREIGN KEY (Team_2_Id) REFERENCES TEAMS(Team_Name), FOREIGN KEY (Match_Id)
REFERENCES MATCHES(Match_Id),);
```

```
CREATE TABLE RANKINGS (Ranking_Id Number PRIMARY KEY,Position int NOT NULL,
Goals_Scored int, Matches_Played int, Yellow_Cards int, Red_Cards int,);
```

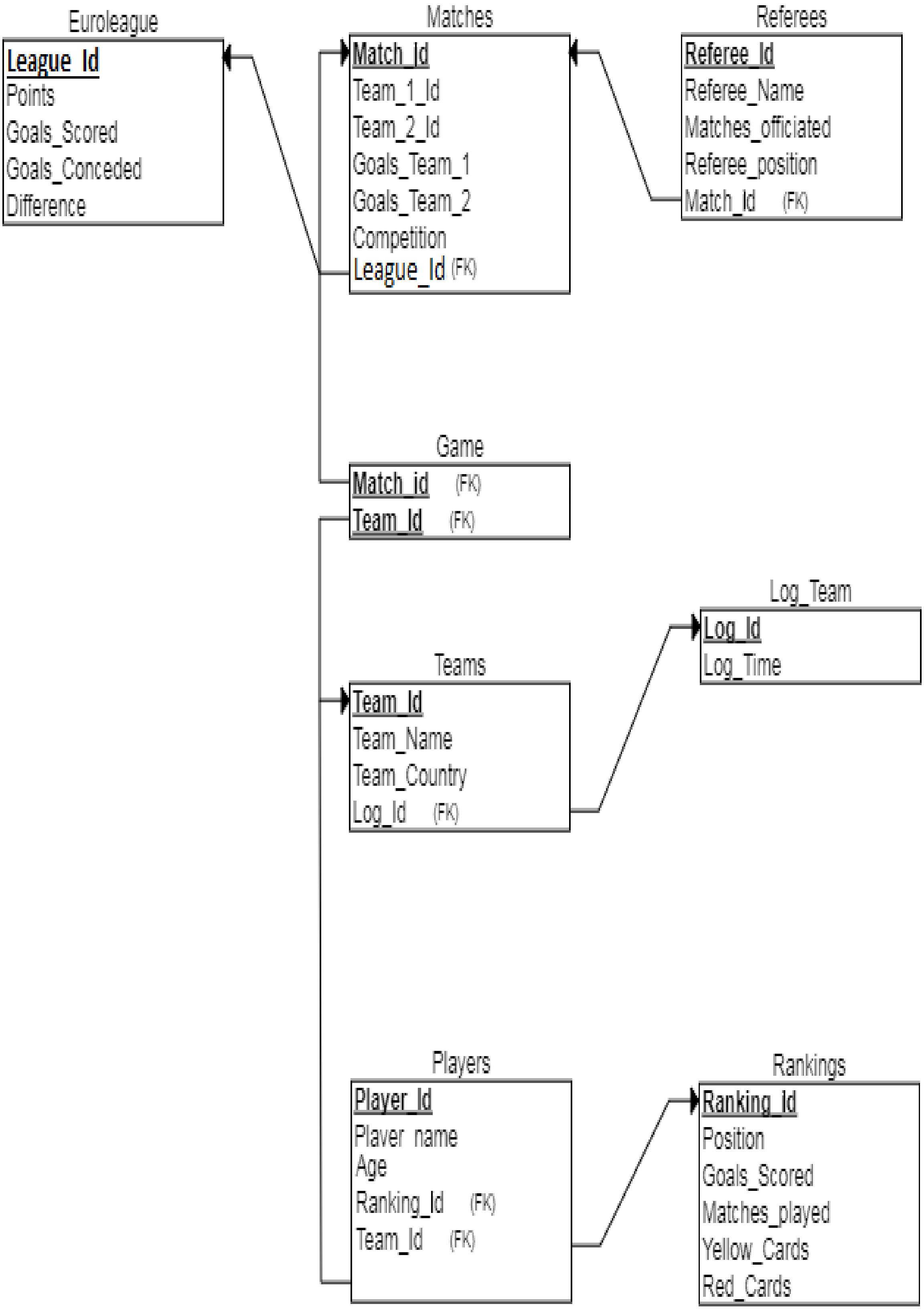
```
CREATE TABLE PLAYERS (Player_Id Number PRIMARY KEY, Player_Name VARCHAR2(20),
Age Int, Ranking_Id Number, Team_Id Number, FOREIGN KEY (Team_2_Id)
REFERENCES TEAMS(Team_Name), FOREIGN KEY (Ranging_Id) REFERENCES
RANKINGS(Ranking_Id),);
```

```
CREATE TABLE REFEREES(Referee_Id Number Primary Key, Referee_Name
VARCHAR2(20),Matches_Officiated Int, Referee_Position VARCHAR2(20), Match_Id
Number, FOREIGN KEY (Match_Id) REFERENCES MATCHES(Match_Id),);
```

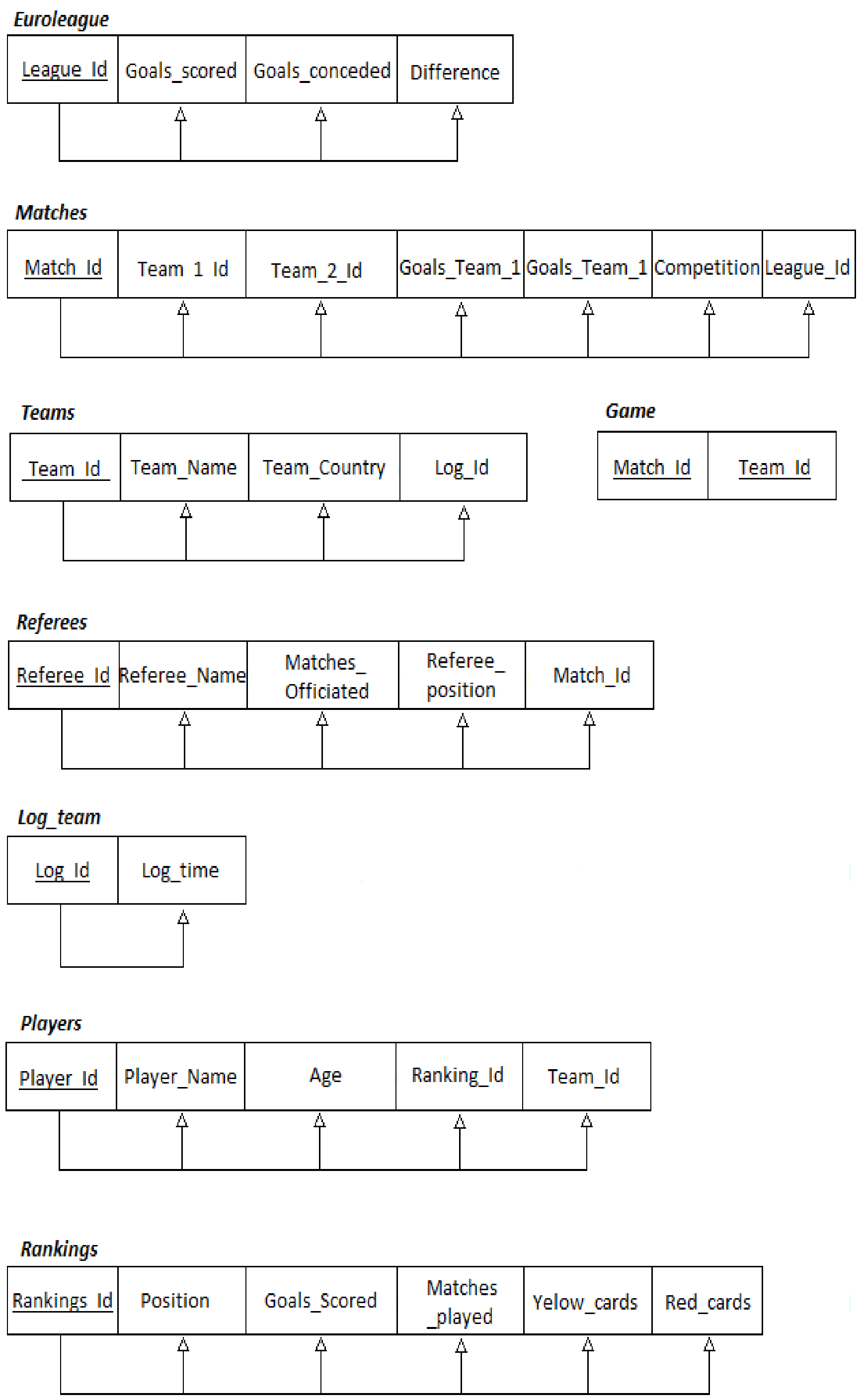
Entity Relationship Diagram



Relational Schema



Functional Dependency Diagram



Semantic Constrains

Semantic constrains were implemented in both the create statements and in the triggers to check the insert statements. Other constrains were implemented using primary and foreign keys. The first semantic constrain was used in the create statement of the table Teams to constrain the field Country to either Spain or England. The second one was used on the table Matches to constrain the Field Competition to either Champions League, Europa Leauge, Premier League or La Liga. The third one in the table Matches as well sets the field Goals to a number >=0. The fourth one, using a trigger, check that, if a match has Premier League or La Liga for competition, the country of the two teams is correct.

A) Country is either Spain or England

```
CONSTRAINT chk_country CHECK (TeamCountry IN ('Spain', 'England'))
```

B) Competition is either Champions League, Europa Leauge, Premier League or La Liga :

```
CONSTRAINT chk_competition CHECK (Competition IN ('Champions League', 'Europa Leauge',
'Premier League', 'La Liga'))
```

D) Goals in the table MATCHES is a number >=0.

```
CONSTRAINT chk_goal_A CHECK (Goal_A >= 0 AND Goal_B >= 0)
```

D) Using a trigger, check that, if a match has Premier League or La Liga for competition, the country of the two teams is correct (both Teams from England for Premier League and both Teams from Spain for La Liga) If not, the match cannot be inserted into the table MATCHES.

```
CREATE OR REPLACE TRIGGER matchCompetition_trigger

BEFORE INSERT ON MATCHES

FOR EACH ROW

DECLARE

  competitionName VARCHAR(20);

  teamA Number;

  teamB Number;

  teamCheckVar VARCHAR(30);

  team2CheckVar VARCHAR(30);

BEGIN

  IF Inserting then

    SELECT :new.Competition INTO competitionName FROM dual;

    SELECT :new.Team_1_Id INTO teamA FROM dual;
    SELECT :new.Team_1_Id INTO teamB FROM dual;


    IF competitionName IN ('Premier League') then
        SELECT Team_Country INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
        SELECT Team_Country INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
        IF teamCheckVar not in ('England') or team2CheckVar not in ('England') then
          Raise_application_error(-20000,'Only English teams are allowed in Premier League!');
        END IF;
    END IF;
    IF competitionName in ('La Liga') then
    SELECT TeamCountry INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
        SELECT TeamCountry INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
        IF teamCheckVar not in ('Spain') or team2CheckVar not in ('Spain') then
          Raise_application_error(-20000,'Only Spanish teams are allowed in La Liga!');
        END IF;
    END IF;
  END IF;
END;
/
```

Security constrains

The security constrains were implemented by creating two main roles in the database, each with specific privileges granted. The main role was declared as Manager, while the secondary role was called Player. Hagi was granted the manager role, identified by “Team”, with the following privileges create table, delete table and update table. Lately the delete table privilege was revoked. Ronaldo was granted the Player role, identified by “Football”, with select privileges on the view Rankings_View.

Use Of users and privileges to implement security constrains

```
Create ROLE Manager IDENTIFIED by Team;

Grant create table to Manager;

Grant delete table to Manager;

Grant update table to Manager;
```

```
REVOKE Delete TABLE FROM Manager;


Create ROLE Player IDENTIFIED by Football;

Grant select on Table Rankings_View;

GRANT Manager TO Hagi;

GRANT Employee TO Ronaldo;
```

View Creation and Select statements

The database has been provided with a view called Rankings_View. This view has been used later used to implement security constrains.

```
CREATE VIEW  Rankings_View (Player_name, Age, Position) AS SELECT Player_Name, Age, Position, FROM PLAYERS, RAKINGS WHERE PLAYERS.Ranking_Id = RANKINGS.Ranking_Id;
```

The select statements are present throughout the code (create view and triggers) as well as on their own right. Two self-standing view statements have been however, one to display all the entries from the view Rankings_View and the second to display all the entries corresponding to games played in the Premiere League competition from the Euroleague table.

```
SELECT * FROM Rankings_View;

SELECT * FROM EUROLEAGUE WHERE Competition='Premiere League';
```

Triggers

This database uses triggers extensively to perform various checks and to automatically populate some tables. They were also used to update values on some tables without the need of the database manager's intervention. In order to accomplish that variables declaration and PL/SQL have been used. Each of the main triggers is explained below.

A) Using a trigger, log into a table LOG_TEAM the timestamps of all the insertions in the table TEAMS:

```
CREATE OR REPLACE CREATE OR REPLACE TRIGGER log_trigger

  AFTER INSERT ON TEAMS

  FOR EACH ROW

  DECLARE

  teamLog_Ide Number;

  theTime TIMESTAMP;
  BEGIN
    teamLog_Ide := :OLD.Log_Id;
    theTime := CURRENT_TIMESTAMP;
    INSERT INTO LOG_TEAM(Log_Id, Log_Time) VALUES(teamLog_Ide,theTime);
  END;
/
```

B) Create a trigger that, every time a new match is inserted in the table MATCHES, inserts the League_Id of the match in the EUROLEAGUE table if the match is not already there and sets points and goals to zero.

```
CREATE OR REPLACE CREATE OR REPLACE TRIGGER league_trigger
  AFTER INSERT ON Matches
  FOR EACH ROW
  Declare
  league_Ide Number;
  anyRowsFoundLeague NUMBER;
           BEGIN
           league_Ide := :OLD.League_Id;
           SELECT count(*) INTO anyRowsFoundLeague from EUROLEAGUE where
EUROLEAGUE.League_Id in League_Ide;
        if anyRowsFound > 0 then
      INSERT INTO EUROLEAGUE VALUES(league_Ide,0,0,0,0);
    end if;
           END;
/
```

C) Using a trigger, check that, if a match has Premier League or La Liga for competition, the country of the two teams is correct (both England for Premier League and Spain for La Liga). If not, the match cannot be inserted into the table MATCHES.

```
CREATE OR REPLACE TRIGGER matchCompetition_trigger
BEFORE INSERT ON MATCHES
FOR EACH ROW
```

```
DECLARE
  competitionName VARCHAR(20);
  teamA Number;
  teamB Number;
  teamCheckVar VARCHAR(30);
  team2CheckVar VARCHAR(30);

BEGIN
  IF Inserting then
    SELECT :new.Competition INTO competitionName FROM dual;
    SELECT :new.Team_1_Id INTO teamA FROM dual;
    SELECT :new.Team_1_Id INTO teamB FROM dual;

    IF competitionName IN ('Premier League') then
      SELECT Team_Country INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
      SELECT Team_Country INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
      IF teamCheckVar not in ('England') or team2CheckVar not in ('England') then
        Raise_application_error(-20000,'Only English teams are allowed in Premier League!');
      END IF;
    END IF;

    IF competitionName in ('La Liga') then
    SELECT TeamCountry INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
      SELECT TeamCountry INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
      IF teamCheckVar not in ('Spain') or team2CheckVar not in ('Spain') then
        Raise_application_error(-20000,'Only Spanish teams are allowed in La Liga!');
      END IF;
    END IF;
  END IF;
END;
/
```

D) Using a trigger, check that a team has no more than 4 matches in the table MATCHES.

E) Using a trigger , update the EUROLEAGUE table by adding three points to the winning team, 1 point to each team in case of a draw, and by updating goals scored, conceded and the goal difference.

Note that the league table is for all the teams and competitions:

```
CREATE OR REPLACE TRIGGER team_no_matches_trigger
BEFORE INSERT ON MATCHES
FOR EACH ROW


DECLARE
no_of_matches number;
no_of_matches2 number;
theTeamAId Number;
theTeamBId Number;
noOfGoalsA int;
noOfGoalsB int;
difference int;
team_a_exists_in_a int;
team_a_exist_in_b int;
team_b_exists_in_a int;
team_b_exists_in_b int;
```

```
BEGIN
  -- Select Values Into Variables
  SELECT :new.Team_1_Id INTO theTeamAId FROM dual;
  SELECT :new.Team_2_Id INTO theTeamBId FROM dual;
  SELECT :new.Goal_Team_1 INTO noOfGoalsA FROM dual;
  SELECT :new.Goal_Team_2 INTO noOfGoalsB FROM dual;
  SELECT count(*) INTO team_a_exists_in_a FROM MATCHES where Team_1_Id = theTeamAId;
  SELECT count(*) INTO team_a_exists_in_b FROM MATCHES where Team_2_Id = theTeamAId;
  SELECT count(*) INTO team_b_exists_in_a FROM MATCHES where Team_1_Id = theTeamBId;
  SELECT count(*) INTO team_b_exists_in_b FROM MATCHES where Team_2_Id = theTeamBId;

  no_of_matches := team_a_exists_in_a + team_a_exists_in_b;
  no_of_matches2 := team_b_exists_in_a + team_b_exists_in_b;
```

Check if the team has more than 4 matches, if so raise an exception:

```
  IF no_of_matches > 4 or no_of_matches2 > 4 then
    RAISE_APPLICATION_ERROR(-19999, 'Team cannot have more than 4 matches scheduled!');
  END IF;
```

There was a draw so each team gets one point:

```
  IF noOfGoalsA = noOfGoalsB then
     INSERT INTO EUROLEAGUE VALUES(theTeamAName, 1, noOfGoalsA, noOfGoalsB, 0);
     INSERT INTO EUROLEAGUE VALUES(theTeamBname, 1, noOfGoalsB, noOfGoalsA, 0);
  END IF;
```

Check if Team A has won, if so give it 3 points.

```
  IF noOfGoalsA > noOfGoalsB then
```

```
        difference := noOfGoalsA - noOfGoalsB;

       INSERT INTO EUROLEAGUE VALUES(theTeamAName, 3, noOfGoalsA, noOfGoalsB,
difference);

    END IF;
```

Check if Team B has won, if so give it 3 points.

```
    IF noOfGoalsB > noOfGoalsA then

        difference := noOfGoalsB - noOfGoalsA;

       INSERT INTO EUROLEAGUE VALUES(theTeamBName, 3, noOfGoalsB, noOfGoalsA,
difference);

    END IF;

END;

/
```

Appendix

```
DROP SEQUENCE matchID_sequence;

DROP TABLE LOG_TEAM;

DROP TABLE EUROLEAGUE;

DROP TABLE MATCHES;

DROP TABLE TEAMS;

DROP TABLE Referees;

DROP TABLE Game;

DROP TABLE Players;

DROP TABLE Rankings;




CREATE TABLE EUROLEAGUE(

  League_Id      Number  PRIMARY KEY,
  Points         int ,
  Goals_Scored    int ,
  Goals_Conceded  int ,
  Difference      int ,
);

CREATE TABLE LOG_TEAM(
  Log_Id       Number PRIMARY KEY,
  Log_Time      TIMESTAMP NOT NULL
);

CREATE TABLE TEAMS(
        Team_Id       Number Primary Key,
   Team_Name      VARCHAR2(30),
   Team_Country    VARCHAR2(10) NOT NULL,
        Log_Id        Number,

   CONSTRAINT chk_country CHECK (TeamCountry IN ('Spain', 'England')) -- C) Country is either Spain or England
        FOREIGN KEY (Log_Id) REFERENCES LOG_TEAM(Log_Id),
);

CREATE TABLE MATCHES(
  Match_ID       NUMBER PRIMARY KEY,-- A) MatchID is an auto increment key. Using a sequence to define it.
  Team_1_Id      Number NOT NULL,
  Team_2_Id      Number NOT NULL,
  Goal__Team_1   VARCHAR2(10) NOT NULL,
  Goal_Team_2    VARCHAR2(10) NOT NULL,
  Competition    VARCHAR2(20) NOT NULL,
  League_Id      Number  NOT NUll,


  CONSTRAINT chk_competition  CHECK (Competition IN ('Champions League', 'Europa
Leauge', 'Premier League', 'La Liga')),
  -- B) Competition is either Champions League, Europa Leauge, Premier League or La Liga .
  CONSTRAINT chk_goal_A CHECK (Goal_A >= 0 AND Goal_B >= 0),-- D) Goals in the table
MATCHES is a number >=0.
  FOREIGN KEY (Team_1_Id) REFERENCES TEAMS(Team_Name),
  FOREIGN KEY (Team_2_Id) REFERENCES TEAMS(Team_Name),
  FOREIGN KEY (League_Id) REFERENCES EUROLEAGUE(League_Id),
);


 CREATE TABLE Game(
  Match_Id Number  Primary  Key,
  Team_Id  Number  Primary  Key,


  FOREIGN KEY (Team_2_Id) REFERENCES TEAMS(Team_Name),
  FOREIGN KEY (Match_Id) REFERENCES MATCHES(Match_Id),
);


CREATE TABLE RANKINGS (
            Ranking_Id      Number  PRIMARY KEY,
            Position        int NOT NULL,
            Goals_Scored    int,
            Matches_Played  int,
            Yellow_Cards    int,
            Red_Cards       int,


);
```

```
CREATE TABLE PLAYERS (
          Player_Id         Number  PRIMARY KEY,
          Player_Name        VARCHAR2(20),
          Age             Int,
   Ranking_Id       Number,
          Team_Id          Number,

  FOREIGN KEY (Team_2_Id) REFERENCES TEAMS(Team_Name),
  FOREIGN KEY (Ranging_Id) REFERENCES RANKINGS(Ranking_Id),



);

CREATE TABLE REFEREES(
          Referee_Id         Number  Primary Key,
          Referee_Name        VARCHAR2(20),
          Matches_Officiated  Int,
          Referee_Position    VARCHAR2(20),
          Match_Id           Number,

          FOREIGN KEY (Match_Id)  REFERENCES MATCHES(Match_Id),
);

--Create View
CREATE VIEW  Rankings_View (Player_name, Age, Position) AS SELECT Player_Name, Age,
Position, FROM PLAYERS, RAKINGS WHERE PLAYERS.Ranking_Id = RANKINGS.Ranking_Id;


--  Match_ID is an auto increment  key. Using a sequence to define it.
CREATE SEQUENCE matchID_sequence
  START WITH 1
```

```
 INCREMENT BY 1;

CREATE OR REPLACE TRIGGER matchID_trigger
 BEFORE INSERT ON MATCHES
 FOR EACH ROW
 BEGIN
  SELECT matchID_sequence.nextval INTO :new.Match_ID FROM dual;
 END;
 /


-- Triggers:
-- a)Using a trigger, log into a table LOG_TEAM the timestamps of all the insertions in the
table TEAMS.

CREATE OR REPLACE CREATE OR REPLACE TRIGGER log_trigger
 AFTER INSERT ON TEAMS
 FOR EACH ROW

 DECLARE

 teamLog_Ide Number;
 theTime TIMESTAMP;


 BEGIN
  teamLog_Ide := :OLD.Log_Id;
  theTime := CURRENT_TIMESTAMP;
  INSERT INTO LOG_TEAM(Log_Id, Log_Time) VALUES(teamLog_Ide,theTime);
  END;
/


-- b)Create a trigger that, every time a new match is inserted in the table MATCHES,
    -- inserts the League_Id of the match in the EUROLEAGUE
    -- table if the match is not already there and sets points and goals to zero.


CREATE OR REPLACE CREATE OR REPLACE TRIGGER league_trigger
  AFTER INSERT ON Matches
  FOR EACH ROW


  Declare


  league_Ide Number;
  anyRowsFoundLeague NUMBER;


            BEGIN


            league_Ide := :OLD.League_Id;


            SELECT count(*) INTO anyRowsFoundLeague from EUROLEAGUE where
EUROLEAGUE.League_Id in League_Ide;


    if anyRowsFound > 0 then
      INSERT INTO EUROLEAGUE VALUES(league_Ide,0,0,0,0);
    end if;



            END;
/
```

```
-- INSERT DATA TO CHECK IF THE TRIGGERS WORK FINE

DELETE FROM TEAMS;


--200 Log_Id  starting value

--300 Team_Id starting value



INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES
(300,'Arsenal','England',200);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES
(301,'Manchester United', 'England',201);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (302,'Chelsea',
'England',202);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (303,'Everton',
'England',203);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (304,'Barcelona'
,'Spain',204);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (305,'Real
Madrid', 'Spain',205);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (306,'Atletico
Madrid' ,'Spain',206);

INSERT INTO TEAMS(Team_Id,Team_Name, Team_Country,Log_Id) VALUES (307,'Sevilla',
'Spain',207);


SELECT * FROM TEAMS;
```

C) Using a trigger, check that, if a match has Premier League or La Liga for competition, the country of the two teams is correct (both Teams from England for Premier League and both Teams from Spain for La Liga) If not, the match cannot be inserted into the table MATCHES.

```
CREATE OR REPLACE TRIGGER matchCompetition_trigger
BEFORE INSERT ON MATCHES
FOR EACH ROW

DECLARE
 competitionName  VARCHAR(20);
 teamA Number;
 teamB Number;
 teamCheckVar VARCHAR(30);
 team2CheckVar VARCHAR(30);

BEGIN
 IF Inserting then
  SELECT :new.Competition  INTO competitionName  FROM dual;
  SELECT :new.Team_1_Id  INTO teamA FROM dual;
  SELECT :new.Team_1_Id  INTO teamB FROM dual;

  IF competitionName  IN ('Premier League') then
    SELECT Team_Country INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
    SELECT Team_Country INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
    IF teamCheckVar not in ('England') or team2CheckVar not in ('England') then
     Raise_application_error(-20000,'Only English teams are allowed in Premier League!');
    END IF;
  END IF;

  IF competitionName  in ('La Liga') then
  SELECT TeamCountry  INTO teamCheckVar FROM TEAMS WHERE Team_Id = teamA;
    SELECT TeamCountry INTO team2CheckVar FROM TEAMS WHERE Team_Id = teamB;
    IF teamCheckVar not in ('Spain') or team2CheckVar not in ('Spain') then
```

```
        Raise_application_error(-20000,'Only Spanish teams are allowed in La Liga!');
       END IF;
    END IF;



  END IF;
END;
/



-- D) Using a trigger, check that a team has no more than 4 matches in the table MATCHES.
-- E) -- Using a trigger , update the EUROLEAGUE table by
-- adding three points to the winning team,
-- 1 point to each team in case of a draw,
-- and by updating goals scored, conceded and the goal difference.
-- Note that the league table is for all the teams and competitions

CREATE OR REPLACE TRIGGER team_no_matches_trigger
BEFORE INSERT ON MATCHES
FOR EACH ROW

DECLARE
no_of_matches number;
no_of_matches2 number;
theTeamAId Number;
theTeamBId Number;
noOfGoalsA int;
noOfGoalsB int;
difference int;
```

```
team_a_exists_in_a int;
team_a_exist_in_b int;
team_b_exists_in_a int;
team_b_exists_in_b int;

BEGIN
   -- Select Values Into Variables
   SELECT :new.Team_1_Id  INTO theTeamAId  FROM dual;
   SELECT :new.Team_2_Id  INTO theTeamBId  FROM dual;
   SELECT :new.Goal_Team_1  INTO noOfGoalsA FROM dual;
   SELECT :new.Goal_Team_2  INTO noOfGoalsB FROM dual;
   SELECT count(*)  INTO team_a_exists_in_a FROM MATCHES where  Team_1_Id =
theTeamAId;
   SELECT count(*)  INTO team_a_exists_in_b FROM MATCHES where  Team_2_Id =
theTeamAId;
   SELECT count(*)  INTO team_b_exists_in_a FROM MATCHES where  Team_1_Id =
theTeamBId;
   SELECT count(*)  INTO team_b_exists_in_b FROM MATCHES where  Team_2_Id =
theTeamBId;

   no_of_matches  := team_a_exists_in_a + team_a_exists_in_b;
   no_of_matches2  := team_b_exists_in_a + team_b_exists_in_b;

   -- Check if the team has more  than 4 matches, if so raise an exception
   IF no_of_matches  > 4 or no_of_matches2 > 4 then
      RAISE_APPLICATION_ERROR(-19999, 'Team cannot  have more  than 4 matches
scheduled!');
   END IF;

   -- There was a  draw so each team gets one  point.
   IF noOfGoalsA = noOfGoalsB then
```

```
      INSERT INTO EUROLEAGUE VALUES(theTeamAName, 1, noOfGoalsA, noOfGoalsB, 0);
      INSERT INTO EUROLEAGUE VALUES(theTeamBname, 1, noOfGoalsB, noOfGoalsA, 0);
   END IF;
   -- Check if Team A has won, if so give it 3 points.
   IF noOfGoalsA > noOfGoalsB then
      difference := noOfGoalsA - noOfGoalsB;
      INSERT INTO EUROLEAGUE VALUES(theTeamAName, 3, noOfGoalsA, noOfGoalsB,
difference);
   END IF;
   -- Check if Team B has won, if so give it 3 points.
   IF noOfGoalsB > noOfGoalsA then
      difference := noOfGoalsB - noOfGoalsA;
      INSERT INTO EUROLEAGUE VALUES(theTeamBName, 3, noOfGoalsB, noOfGoalsA,
difference);
   END IF;
END;
/


--100 League_Id starting value


-- This insert should raise an exception, because user is trying to insert a Spanish team with
an English team.
INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES ( 300, 304,1,1,'La Liga',100);


-- The following inserts should work fine.
 -- This insert shuould insert a draw in EuroLeague
INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (300, 301,1,1,'Premiere League',100);
 -- This insert should insert winning points for Arsenal in EuroLeague
```

```
INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (300, 302,2,1,'Premiere League',101);

 -- This insert should insert winning points for Arsenal in EuroLeague

INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (300, 303,2,1,'Premiere League',102);

 -- This insert should insert winning points for Everton in EuroLeague

INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (300, 303,1,2,'Premiere League',103);

 -- This insert should insert winning points for Barcelona in EuroLeague

INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (305, 304,1,2,'La Liga',103);

 -- This insert should raise an exception, because a team cannot play more than 4 matches.

INSERT INTO MATCHES(Team_1_Id, Team_2_Id, Goal_Team_1, Goal_Team_2, Competition,
League_Id) VALUES (300, 303,2,2,'Premiere League',104);


--Populate Game Table


INSERT INTO GAME (Match_Id, Team_Id) VALUES (1,300);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (1,301);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (2,300);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (2,302);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (3,300);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (3,303);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (4,300);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (4,303);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (5,305);

INSERT INTO GAME (Match_Id, Team_Id) VALUES (5,304);


--Populate Referees Table

--600 Referee_Id starting value
```

```
INSERT INTO REFEREES(Referee_Id, Referee_Name, Matches_officiated, Referee_position,
Match_id ) VALUES (600,'Juan Gonsalves', 100, 'Field Judge',1);

INSERT INTO REFEREES(Referee_Id, Referee_Name, Matches_officiated, Referee_position,
Match_id ) VALUES (601,'Ahmad Orin', 75, 'Center Judge',1);

INSERT INTO REFEREES(Referee_Id, Referee_Name, Matches_officiated, Referee_position,
Match_id ) VALUES (602,'Ion Popescu',80, 'Line Judge', 1);

INSERT INTO REFEREES(Referee_Id, Referee_Name, Matches_officiated, Referee_position,
Match_id ) VALUES (603,'Cris Oreily',105,'Side Judge', 1);

INSERT INTO REFEREES(Referee_Id, Referee_Name, Matches_officiated, Referee_position,
Match_id ) VALUES (604,'Laurent Hugh',115, 'Head Linesman', 1);


-- Populate Players Table

--500 Player_Id starting value


INSERT INTO PLAYERS(Player_Id, Player_name, Age, Ranking_Id, Team_Id ) VALUES (500,
'Cris Ronaldo', 23, 400, 305);

INSERT INTO PLAYERS(Player_Id, Player_name, Age, Ranking_Id, Team_Id ) VALUES (501,
'John Cena', 30, 401, 300);

INSERT INTO PLAYERS(Player_Id, Player_name, Age, Ranking_Id, Team_Id ) VALUES (502,
'Cris Ferdinand', 402, 301);

INSERT INTO PLAYERS(Player_Id, Player_name, Age, Ranking_Id, Team_Id ) VALUES (503,
'Ion Hagi', 31, 403, 304);

INSERT INTO PLAYERS(Player_Id, Player_name, Age, Ranking_Id, Team_Id ) VALUES (504,
'Jean Pascale', 27, 404, 302);


-- Populate Players Table

--400 Ranking_Id estarting value


INSERT INTO RANKINGS(Ranking_Id, Position, Goals_Scored, Matches_Played, Yellow_Cards,
Red_Cards) VALUES (403, 1, 30, 35, 5, 1);
```

```
INSERT INTO RANKINGS(Ranking_Id, Position, Goals_Scored, Matches_Played, Yellow_Cards,
Red_Cards) VALUES (401, 2, 25, 37, 6, 2);

INSERT INTO RANKINGS(Ranking_Id, Position, Goals_Scored, Matches_Played, Yellow_Cards,
Red_Cards) VALUES (404, 3, 20, 40, 8, 5);

INSERT INTO RANKINGS(Ranking_Id, Position, Goals_Scored, Matches_Played, Yellow_Cards,
Red_Cards) VALUES (400, 4, 15, 35, 6, 2);

INSERT INTO RANKINGS(Ranking_Id, Position, Goals_Scored, Matches_Played, Yellow_Cards,
Red_Cards) VALUES (402, 5, 10, 30, 3, 1);


-- select statments


SELECT * FROM Rankings_View;

SELECT * FROM EUROLEAGUE WHERE Competition='Premiere  League';


-- Use Of users and priviledges to implemet security constrains


Create ROLE Manager IDENTIFIED by Team;

Grant create table to Manager;

Grant delete table to Manager;

Grant drop table to Manager


REVOKE DROP TABLE FROM Manager;


Create ROLE Player IDENTIFIED by Football;

Grant select on Table Rankings_View;

GRANT Manager TO Hagi;

GRANT Employee TO Ronaldo;
```