```
CREATE DATABASE Final
USE Final

CREATE TABLE Sports (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50) UNIQUE,
   capacity INT
);


CREATE TABLE Team (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50) UNIQUE,
   sportstype INT,
   rating decimal(4,2),
   check(rating>=1 and rating<=5),
   countRatings int,
   FOREIGN KEY (sportstype) REFERENCES Sports(ID)
);

CREATE TABLE Player (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50),
   username VARCHAR(50) UNIQUE,
   password VARCHAR(50),
   email VARCHAR(50),
   contactnumber VARCHAR(11) UNIQUE,
   gender CHAR CHECK(gender='F' OR gender='M'),
   sportstype INT,
   AssociatedTeam INT,
   FOREIGN KEY (sportstype) REFERENCES Sports(ID),
   FOREIGN KEY (AssociatedTeam) REFERENCES Team(ID)
);

CREATE TABLE captains (
   teamid INT,
   playerid INT,
   PRIMARY KEY (teamid, playerid),
   FOREIGN KEY (teamid) REFERENCES Team(ID),
   FOREIGN KEY (playerid) REFERENCES Player(ID)
);

CREATE TABLE MatchType (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50)
);
```

```
CREATE TABLE Field (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   ownername VARCHAR(50),
   ownermail VARCHAR(50),
   name VARCHAR(50) UNIQUE,
   location VARCHAR(50),
   sportsoffering INT,
   PhoneNumber VARCHAR(11),
   FOREIGN KEY (sportsoffering) REFERENCES Sports(ID)
);


CREATE TABLE Tournament (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   name VARCHAR(50) UNIQUE,
   field INT,
   noofmatches INT,
   duration INT,
   startingDate DATE,
   sports INT,
   entry_fee INT,
   PlayerID INT,
   FOREIGN KEY (field) REFERENCES Field(ID),
   FOREIGN KEY (sports) REFERENCES Sports(ID),
   CONSTRAINT xyz FOREIGN KEY (PlayerID) REFERENCES Player(ID)
);



create table torregistration(
        torid int ,
   teamid int,
   foreign key(torid) references tournament(ID),
   foreign key(teamid) references team(ID),
   primary key(torid,teamid)
);

CREATE TABLE Matches (
   ID INT PRIMARY KEY AUTO_INCREMENT,
   matchtype INT,
   TimeOfMatch TIME,
   DateofMatch DATE,
   Team1 INT,
   Team2 INT,
   field INT,
   CHECK (Team1 != Team2),
   UNIQUE (field, TimeOfMatch, DateOfMatch),
   FOREIGN KEY (matchtype) REFERENCES MatchType(ID),
```

```sql
    FOREIGN KEY (Team1) REFERENCES Team(ID),
    FOREIGN KEY (Team2) REFERENCES Team(ID),
    FOREIGN KEY (field) REFERENCES Field(ID)
);

CREATE TABLE TournamentMatch (
    MatchId INT,
    TournamentId INT,
    PRIMARY KEY (MatchId, TournamentId),
    FOREIGN KEY (MatchId) REFERENCES Matches(ID),
    FOREIGN KEY (TournamentId) REFERENCES Tournament(ID)
);

CREATE TABLE Payment (
    ID INT PRIMARY KEY AUTO_INCREMENT,
    DateofPayment DATE,
    amount INT,
    BankTo VARCHAR(50),
    BankFrom VARCHAR(50),
    PlayerID INT,
    field INT,
    FOREIGN KEY (PlayerID) REFERENCES Player(ID),
    FOREIGN KEY (field) REFERENCES Field(ID)
);

CREATE TABLE feed (
    Links VARCHAR(255) UNIQUE,
    sportstype INT,
    Title VARCHAR(50),
    FOREIGN KEY (sportstype) REFERENCES Sports(ID)
);

CREATE TABLE rewards (
    nofofmatches INT PRIMARY KEY,
    name VARCHAR(50)
);


– END OF TABLES
– VALUES START HERE

INSERT INTO Sports (name, capacity) VALUES
('Football', 15),
('Cricket', 15),
('Tennis', 2),
('Basketball', 15);
```

```
INSERT INTO rewards (nofofmatches, name)
VALUES
(10,'Noob'),
(50,'Half-Century'),
(100,'Century');

– PROCEDURES

_ *****************************************************************************************
– USER SIGNUP
_ *****************************************************************************************

DELIMITER //
CREATE PROCEDURE UserSignup(
   IN name VARCHAR(50),
   IN user VARCHAR(50),
   IN pass VARCHAR(50),
   IN sp INT,
   IN phone VARCHAR(50),
   IN em VARCHAR(50),
   IN gr CHAR,
   OUT Flag INT
)
BEGIN
   DECLARE username_count INT;
   DECLARE email_count INT;
   DECLARE phone_count INT;

   SELECT COUNT(*) INTO username_count FROM Player WHERE username = user;

   IF (username_count > 0) THEN
      -- Username already taken
      SET Flag = 1;
   ELSE
      SELECT COUNT(*) INTO email_count FROM Player WHERE email = em;

      IF (email_count > 0) THEN
         -- Gmail already taken
         SET Flag = 2;
      ELSE
         SELECT COUNT(*) INTO phone_count FROM Player WHERE contactnumber =
phone;

         IF (phone_count > 0) THEN
            -- User with this contact already exists
            SET Flag = 3;
         ELSE
            SET Flag = 0;
```

```
          END IF;
       END IF;
    END IF;
    IF (Flag=0) THEN
                    INSERT INTO
Player(name,username,password,email,contactnumber,gender,sportstype)
       VALUES(name,user,pass,em,phone,gr,sp);
       END IF;
END//

DELIMITER ;

_ *******************************************************************************************
– END OF USER SIGNUP
_ *******************************************************************************************

_ *******************************************************************************************
– USER LOGIN
_ *******************************************************************************************

DELIMITER //

CREATE PROCEDURE UserLogin(
    IN user VARCHAR(50),
    IN pass VARCHAR(50),
    OUT Flag INT
)
BEGIN
    DECLARE P INT;
    DECLARE PA INT;

    SELECT COUNT(*) INTO P FROM Player WHERE username = user;

    IF (P > 0) THEN
                     SELECT COUNT(*) INTO PA FROM Player WHERE username = user and
password=pass;
        if(PA>0) THEN
                              SET FLAG=0;
                     ELSE
                              SET FLAG=1;
         END IF;
     ELSE
       SET FLAG=2;
       END IF;
END//

DELIMITER ;
```

```
_ ******************************************************************************************
– END OF  USER LOGIN
_ ******************************************************************************************

_ ******************************************************************************************
– INSERT TOURNAMENT
_ ******************************************************************************************

DELIMITER //

CREATE PROCEDURE inserttournament(
   IN p_name VARCHAR(50),
   IN p_fieldname VARCHAR(50),
   IN p_noofteams INT,
   IN p_duration INT,
   IN p_sd DATE,
   IN spo INT,
   IN ent INT,
   IN userid INT,
   IN p_teamid INT,
   OUT flag INT
)
BEGIN
   DECLARE v_id INT;
   DECLARE v_f INT;
         DECLARE exi INT;
   -- Initialize flag to 1
   SET flag = 1;
         SELECT count(*) into exi from captains where teamid=p_teamid and playerid=userid;
   if(p_sd<current_date()) then
   set flag=3;
   end if;
   if(exi=0 and flag=1) then
    SET flag=2;
    end if;
   -- Check if the tournament name already exists
   SELECT ID INTO v_f FROM Tournament WHERE name = p_name;

   -- If tournament name already exists, set flag to 0
   IF v_f IS NOT NULL THEN
      SET flag = 0;
   END IF;

   -- If flag is 1 (i.e., tournament name doesn't exist), proceed with insertion
   IF flag = 1 THEN
      -- Get the ID of the field
      SELECT ID INTO v_id FROM Field WHERE name = p_fieldname;
```

```
        -- Insert into Tournament table
        INSERT INTO Tournament (name, field, noofmatches, duration,
startingDate,sports,entry_fee)
        VALUES (p_name, v_id, p_noofteams, p_duration, p_sd,spo,ent,userid);
    END IF;

END //

DELIMITER ;

_ ****************************************************************************************
– END OF INSERT TOURNAMENT
_ ****************************************************************************************

_ ****************************************************************************************
– CHECK TOURNAMENT
_ ****************************************************************************************

DELIMITER //
create procedure Checktournament
(
IN p_toname varchar(50),
IN userid int,
IN p_teamid int,
out flag int
)
Begin
            declare p_torid int;
            Declare coun int;
    Declare c int;
    Declare exi int;
    set flag=1;
    select ID into p_torid from tournament where name = p_toname;
    SELECT count(*) into coun from torregistration where torid=p_torid;
    SELECT noofmatches into c from tournament  where ID=p_torid;
    SELECT count(*) into exi from captains where teamid=p_teamid and playerid=userid;
    if(exi=0) then
                    set flag=2;
            end if;
    if(c=coun and flag=1) then
                    set flag=0;
            end if;
END //
DELIMITER ;

_ ****************************************************************************************
– END OF CHECK TOURNAMENT
```

```
_ ********************************************************************************************

_ ********************************************************************************************
– PAMIN
_ ********************************************************************************************

DELIMITER //

CREATE PROCEDURE pamin(
   IN p_date DATE,
   IN amount INT,
   IN to_bank VARCHAR(50),
   IN from_bank VARCHAR(50),
   IN pid INT,
   IN fid INT
)
BEGIN
   INSERT INTO Payment ( DateofPayment, amount, BankTo, BankFrom, PlayerID, field)
   VALUES ( p_date, amount, to_bank, from_bank, pid, fid);
END//

DELIMITER ;
_ ********************************************************************************************
– END OF PAMIN
_ ********************************************************************************************
_ ********************************************************************************************
– INSERT TEAM
_ ********************************************************************************************

DELIMITER //
CREATE PROCEDURE insertteam(
IN nam VARCHAR(50),
IN PID INT,
OUT flag INT
)
BEGIN
DECLARE S INT;
DECLARE t INT;
-- Check if the team already exists
IF EXISTS (SELECT 1 FROM team WHERE name = nam) THEN
SET flag = 2; -- Team already exists
ELSE
-- Get the sportstype from the Player table
SELECT sportstype INTO S FROM Player WHERE ID = PID;
-- Insert new team into the team table
INSERT INTO team (name, sportstype, rating) VALUES (nam, S, 1);
-- Get the ID of the newly inserted team
SELECT ID INTO t FROM team WHERE name = nam;
```

```
-- Insert into captains table
INSERT INTO captains (teamid, playerid) VALUES (t, PID);
-- Update the Player table with the associated team
UPDATE Player SET Associatedteam = t WHERE ID = PID;
-- Indicate success
SET flag = 1;
END IF;
END//
DELIMITER ;
_ ************************************************************************************
–  END OF INSERT TEAM
_ ************************************************************************************

_ ************************************************************************************
– INSERT FIELD
_ ************************************************************************************
DELIMITER //
CREATE PROCEDURE insertField(
   IN f_fieldname VARCHAR(50),
   IN f_ownername VARCHAR(50),
   IN f_mail VARCHAR(50),
   IN f_location varchar(50),
   IN phone VARCHAR(50),
   IN f_sports int,
   OUT Flag INT
)
BEGIN
   DECLARE username_count INT;

   SELECT COUNT(*) INTO username_count FROM Field WHERE name = f_fieldname;
         set Flag=1;
   IF (username_count > 0) THEN
      -- Username already taken
      SET Flag = 0;
         END IF;
   IF (Flag=1) THEN
                  INSERT INTO
Field(ownername,ownermail,name,location,sportsoffering,PhoneNumber)
      VALUES(f_ownername,f_mail,f_fieldname,f_location,f_sports,phone);
      END IF;
END//

DELIMITER ;

_ ************************************************************************************
– END OF INSERT FIELD
_ ************************************************************************************
```

```
_ ********************************************************************************************
– INSERT TOUrNAMENT MATCH
_ ********************************************************************************************

DELIMITER //

CREATE PROCEDURE insertTourMatch(
   IN matcht int,
   IN timeofm TIME,
   IN dateofm DATE,
   IN T1name VARCHAR(50),
   IN T2name VARCHAR(50),
   IN fid INT,
   IN p_torid INT,
   OUT flag INT
)
BEGIN

   DECLARE T1id INT;
   DECLARE T2id INT;
   declare matid int;
   set flag=1;
   IF T1name IS NULL THEN
      SET T1id = NULL;
   ELSE
      SELECT ID INTO T1id FROM Team WHERE name = T1name;
   END IF;

   IF T2name IS NULL THEN
      SET T2id = NULL;
   ELSE
      SELECT ID INTO T2id FROM Team WHERE name = T2name;
   END IF;
   if(T1id=T2id) then
   set flag=0;
   end if;
   if(dateofm<current_date()) then
   set flag=2;
   end if;
   if(flag=1) then
   INSERT INTO Matches (matchtype, TimeOfmatch, DateofMatch, Team1, Team2, field)
   VALUES (matcht, timeofm, dateofm, T1id, T2id, fid);
   end if;
   if (matcht=2 and flag=1)
   then
   select ID into matid from matches as m where m.DateofMatch = dateofm and field = fid
and m.TimeofMatch = timeofm;
   INSERT into tournamentmatch(MatchId, TournamentId)
   Values (matid, p_torid);
         end if;


END //
DELIMITER ;

_ **********************************************************************************************
–  END OF TOURINSERT
_ **********************************************************************************************

_ **********************************************************************************************
– FIND MATCH
_ **********************************************************************************************

DELIMITER //

CREATE PROCEDURE findmatch(
   IN fieldname VARCHAR(50),
   IN timeofmat TIME,
   IN userid INT,
   IN p_teamid INT,
   IN sd DATE,
   OUT flag INT
)
BEGIN
   DECLARE test INT;
         DECLARE t INT;
   DECLARE exi INT;
   SET flag=1;
   SELECT ID INTO test FROM Field WHERE name = fieldname;
         SELECT         count(*) INTO t FROM matches WHERE TimeOfMatch=timeofmat and
field=test and DateofMatch=sd;
   SELECT count(*) into exi from captains where teamid=p_teamid and playerid=userid;
   if(sd<current_date()) then
   set flag=3;
   end if;
   if(exi=0 and flag=1) then
    SET flag=2;
    end if;
   If(t!=0 and flag=1) then
                  SET flag=0;
         END IF;
END //

DELIMITER ;

_ **********************************************************************************************
```

```
– END OF FIND MATCH
_ ****************************************************************************************

_ ****************************************************************************************
– INSERT MATCH
_ ****************************************************************************************

DELIMITER //

CREATE PROCEDURE insertMatch(
   IN timeofm TIME,
   IN dateofm DATE,
   IN T1ID VARCHAR(50),
   IN fieldname VARCHAR(50)
)
BEGIN
   DECLARE fid INT;
   SELECT ID INTO fid FROM Field WHERE name = fieldname;
   INSERT INTO Matches (matchtype, TimeOfmatch, DateofMatch, Team1, Team2, field)
   VALUES (1, timeofm, dateofm, T1ID, NULL, fid);
END //
DELIMITER ;

_ ****************************************************************************************
– END OF INSERT MATCH
_ ****************************************************************************************

_ ****************************************************************************************
– ADD TEAM
_ ****************************************************************************************

DELIMITER //
CREATE PROCEDURE addteam
(IN nam varchar(50),
IN PID INT,
OUT flag INT)
BEGIN
Declare S int;
Declare c int;
Declare t int;
Declare l int;
set flag=0;
select sportstype into S from Player where ID=PID;
select  capacity into l from sports where ID=S;
select ID into t from team where name=nam;
select count(*) into c from Player group by AssociatedTeam having AssociatedTeam=t  ;
IF c < l THEN
      UPDATE Player
```

```
        SET AssociatedTeam = t
        WHERE ID = PID;
        SET flag = 1;
    ELSE
        SET flag = 3;
    END IF;

END//
DELIMITER ;

_ ****************************************************************************************
– END OF ADD TEAM
_ ****************************************************************************************


_ ****************************************************************************************
– REWARDS
_ ****************************************************************************************

DELIMITER //

CREATE PROCEDURE rewards(
    IN id INT,
    OUT num INT
)
BEGIN
    SELECT IFNULL((SELECT COUNT(Player.ID)
                FROM Player
                JOIN Matches ON (Player.AssociatedTeam = Matches.Team1 OR
Player.AssociatedTeam = Matches.Team2)
                WHERE Player.ID = id AND Matches.Team2 IS NOT NULL AND
CURRENT_DATE() > DateofMatch
                GROUP BY Player.ID), 0) INTO num;
END//

DELIMITER ;

_ ****************************************************************************************
– END OF REWARDS
_ ****************************************************************************************

DELIMITER //

CREATE PROCEDURE SearchPlayer(IN playerID INT)
BEGIN
    SELECT Player.ID,
          Player.name AS pname,
          Sports.name AS sport_name,
        Team.name AS team_name,
        Team.rating AS Rating,
        Player.contactnumber,
        Player.email
    FROM Player
    LEFT JOIN Sports ON Player.sportstype = Sports.ID
    LEFT JOIN Team ON Player.AssociatedTeam = Team.ID
    WHERE Player.id = playerID;
END //

DELIMITER ;
-- ****************************************************************************************
-- END OF SearchPlayer
-- ****************************************************************************************
DELIMITER //

CREATE PROCEDURE GetPastMatchesHistory(IN playerID INT)
BEGIN
    SELECT m.ID AS MatchID,
        mt.name AS MatchTypeName,
        m.TimeOfMatch,
        m.DateofMatch,
        f.name AS FieldName
    FROM Matches m
    LEFT JOIN MatchType mt ON m.matchtype = mt.ID
    LEFT JOIN Field f ON m.field = f.ID
    WHERE (m.Team1 IN (SELECT AssociatedTeam FROM Player WHERE ID = playerID)
       OR m.Team2 IN (SELECT AssociatedTeam FROM Player WHERE ID = playerID))
      AND TIMESTAMP(m.DateofMatch, m.TimeOfMatch) < NOW();
END //

DELIMITER ;
-- ****************************************************************************************
-- END OF GetPastMatchesHistory
-- ****************************************************************************************


DELIMITER //

CREATE PROCEDURE SearchTeam(IN teamID INT)
BEGIN
    SELECT Team.name AS TeamName,
        Player.name AS CaptainName,
        Sports.name AS SportsName,
        Team.rating AS TeamRating
    FROM Team
    LEFT JOIN captains ON Team.ID = captains.teamid
    LEFT JOIN Player ON captains.playerid = Player.ID
```

```
   JOIN Sports ON Team.sportstype = Sports.ID
   WHERE Team.ID = teamID;
END //

DELIMITER ;

-- ****************************************************************************************
-- END OF SearchTeam
-- ****************************************************************************************

DELIMITER //

CREATE PROCEDURE GetTeamMatches(IN teamID INT)
BEGIN
   SELECT Matches.ID AS MatchID,
        MatchType.name AS MatchTypeName,
        Matches.TimeOfMatch,
        Matches.DateofMatch,
        Field.name AS FieldName
   FROM Team
   LEFT JOIN Matches ON (Matches.Team1 = Team.ID OR Matches.Team2 = Team.ID)
   LEFT JOIN MatchType ON Matches.matchtype = MatchType.ID
   LEFT JOIN Field ON Matches.field = Field.ID
   WHERE Team.ID = teamID
    AND TIMESTAMP(Matches.DateofMatch, Matches.TimeOfMatch) < NOW();
END //

DELIMITER ;

-- ****************************************************************************************
-- END OF GetTeamMatches
-- ****************************************************************************************

DELIMITER //

CREATE PROCEDURE UpdateTeamRating(IN newRating DECIMAL(4, 2), IN teamID INT)
BEGIN
   UPDATE Team
   SET countRatings = countRatings + 1,
      rating = (rating * countRatings + newRating) / (countRatings + 1)
   WHERE Team.ID = teamID;
END //

DELIMITER ;

-- ****************************************************************************************
-- END OF UpdateTeamRating
-- ****************************************************************************************
```