2022 Argentina National Team

Version 4.0

<Victor DeLarosa>

<University of Mississippi>

<5/3/2024>

Player
PlayerID (Primary Key)
First Name
Last Name
Age
Position
ClubID (Foreign Key referencing Club)

Player Statistics
PlayerID (Primary Key)
Goals Scored
Assists
Yellow Cards
Red Cards

Player History
HistoryID (Primary Key)
PlayerID (Foreign Key referencing Player)
ClubID
Start Date
End Date

Referee
RefereeID (Primary Key)
First Name
Last Name
Age
Total Fouls Called
Yellow Cards Given
Red Cards Given

Club
ClubID (Primary Key)
Club Name
Country
League

National Team
National Team ID (Primary Key)
Coach
Year Founded

Game
GameID (Primary Key)
RefereeID (Foreign Key referencing Referee)
Date
Opponent
Competition
Result

Game Time
Game Time ID (Primary Key)
GameID (Foreign Key referencing Game)
Eastern Standard Time (EST)
Central Standard Time (CST)
Mountain Standard Time (MST)
Pacific Standard Time (PST)

Player Statistics Per Match
GameID (Primary Key)
PlayerID (Primary Key)
Goals Scored
Assists
Yellow Cards
Red Cards

Match Location
Match Location ID (Primary Key)
GameID (Foreign Key referncing Game)
Opponent
City
Country
Stadium Name

Attendance
AttendanceID (Primary Key)
GameID (Foreign Key referencing Game)
Stadium Name
Stadium Capacity
Stadium Attendance
Television Coverage

Possession Time
Possession Time ID (Primary Key
GameID (Foreign Key referencing Game)
Possession Percentage
Opponent Possession Percentage
Formation
Opponent Formation

UML DIAGRAM:

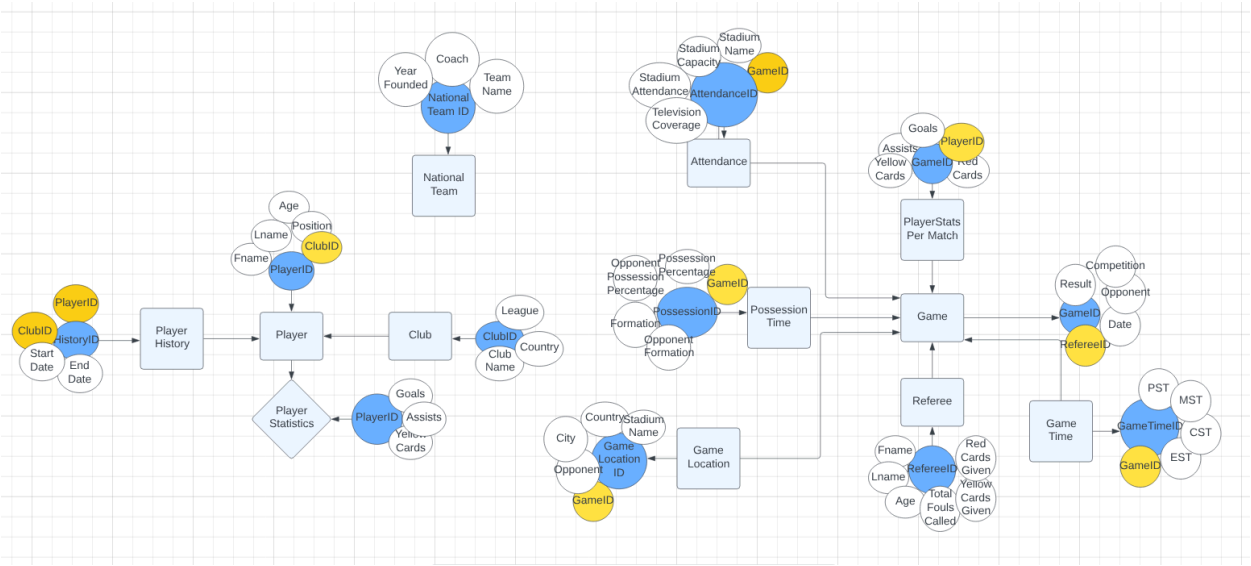

Table of Contents:

Brief Description: creates cursors for most goals, assists and yellow cards of game 6. Then prints the players information which reflects the usual 'norm' of soccer… in which the forwards are getting goals/assists, while the defenders are getting yellow cards.

Block 7 **Page# 49-52**

Brief Description: this query creates a cursor to essentially achieve the exact same goal as block 6 only that it covers game 2 rather than 6.

Block 8 **Page# 53-55**

Brief Description: shows the use of exception handling, in this case it's tracking the most goals scored. An exception is thrown based on a certain number of goals.

Block 9 **Page# 56-61**

Brief Description: Displays the possession time difference in the games, separating them from the highest possession time to the lowest.

Block 10 **Page# 62-64**

Brief Description: enacts a fine against players who received a penalty during the last game of the tournament.

Block 11 **Page# 65-68**

Brief Description: query seeks to discover the earliest start time as well as the latest start time across all seven games played in the World Cup using cursors.It prints the general information to go along with game time such as, the opponent, game result and television coverage.

Block 12 **Page# 69-72**

Brief Description: cursors are created to display and calculate the players with the longest tenure on the Argentina national team. Calculates in various ways.

Block 13 **Page# 73-75**

Brief Description: uses package in order to break down the total length of the 2022 World Cup, breaks it down into different intervals. While displaying each game, date and result.

Block 14 **Page# 76-79**

Brief Description: dispurses a fine to players that received yellow cards during the final game of the tournament. Uses the keyword UPDATE to add a new fictional player.

Project Description

i. The goal of this project is to illustrate the use of this database to track the performance of the 2022 Argentina National team, specifically during the 2022 World Cup. The dataset consists of a variety of tables that include all of the important facts to know about the Argentinian team.

Information Needs:

i. Players are essentially the basis of this. Without the players the data of all these players does not exist. This will include their names, age, position, (club) salary along with their corresponding ClubID, which indicates the regional club they play for.

ii. Furthermore, these players are not the stars that they have become without their incredible statistics. This dataset has recorded all of the goals, assists, yellow and red cards throughout the tournament.

iii. The goal of this project is to simply highlight all elements of the Argentina National Teams journey in the World Cup using SQL. Including every single important entity involved.

```
-- Victor DeLarosa mis 409 final project
-- for all blocks run each procedure/function one at a time and separately to avoid any issues
-- Drop Tables

DROP table Referees;
DROP table Club;
DROP table Game;
DROP table Game_Time;
DROP table Player;
DROP table Player_Statistics;
DROP table Player_History;
DROP table Game_Location;
DROP table Attendance;
DROP table Possession_Time;
DROP table PlayerStats_PerMatch;



-- create table Referees
CREATE TABLE Referees(
   Referee_ID VARCHAR(2) PRIMARY KEY,
   First_Name VARCHAR(15),
   Age INT,
   Total_Fouls_Given INT,
   Total_Yellow_Cards_Given INT,
   Total_Red_Cards_Given INT
);

INSERT INTO Referees VALUES('R1', 'Slavko', 44, 28, 6, 0);
INSERT INTO Referees VALUES('R2', 'Daniele', 48, 57, 9, 0);
INSERT INTO Referees VALUES('R3', 'Danny', 40, 17, 2, 0);
INSERT INTO Referees VALUES('R4', 'Szymon', 42, 68, 10, 0);
```

```
INSERT INTO Referees VALUES('R5', 'Antonio Miguel', 46, 48, 14, 1);


-- create table Club
CREATE TABLE Club (
   Club_ID VARCHAR(3) PRIMARY KEY,
   Club_Name VARCHAR(22),
   Country VARCHAR(18),
   League VARCHAR(19)
);

INSERT INTO Club VALUES('101', 'Inter Miami', 'United States', 'Major League
Soccer');
INSERT INTO Club VALUES('102', 'FC Barcelona', 'Spain', 'La Liga');
INSERT INTO Club VALUES('103', 'S.L. Benfica', 'Portugal', 'Primeira Liga');
INSERT INTO Club VALUES('104', 'Aston Villa', 'England', 'Premier League');
INSERT INTO Club VALUES('105', 'Sevilla FC', 'Spain', 'La Liga');
INSERT INTO Club VALUES('106', 'Atletico Madrid', 'Spain', 'La Liga');
INSERT INTO Club VALUES('107', 'Manchester City', 'England', 'Premier League');
INSERT INTO Club VALUES('108', 'Chelsea', 'England', 'Premier League');
INSERT INTO Club VALUES('109', 'Liverpool', 'England', 'Premier League');
INSERT INTO Club VALUES('110', 'Nottingham Forest F.C.', 'England', 'Premier
League');
INSERT INTO Club VALUES('111', 'Tottenham Hotspur F.C.', 'England', 'Premier
League');
INSERT INTO Club VALUES('112', 'A.S. Roma', 'Italy', 'Serie A');
INSERT INTO Club VALUES('113', 'Real Betis', 'Spain', 'La Liga');


-- create table Game
CREATE TABLE Game (
```

```
    Game_ID VARCHAR(2) PRIMARY KEY,
    Referee_ID VARCHAR(2),
    Game_Date DATE,
    Opponent VARCHAR(20),
    Competition VARCHAR(20),
    Results VARCHAR(20),
    FOREIGN KEY (Referee_ID) REFERENCES Referees(Referee_ID)
);

INSERT INTO Game VALUES('G1','R1',TO_DATE('2022-11-22',
'YYYY-MM-DD'),'Saudi Arabia','Group Stage','1-2(Loss)');
INSERT INTO Game VALUES('G2','R2',TO_DATE('2022-11-26',
'YYYY-MM-DD'),'Mexico','Group Stage','2-0(Win)');
INSERT INTO Game VALUES('G3','R3',TO_DATE('2022-11-30',
'YYYY-MM-DD'),'Poland','Group Stage','2-0(Win)');
INSERT INTO Game VALUES('G4','R4',TO_DATE('2022-12-03',
'YYYY-MM-DD'),'Australia','Round of 16','2-1(Win)');
INSERT INTO Game VALUES('G5','R5',TO_DATE('2022-12-09',
'YYYY-MM-DD'),'Netherlands','Quarter Finals','2(3) - 2(4)(Win)');
INSERT INTO Game VALUES('G6','R2',TO_DATE('2022-12-13',
'YYYY-MM-DD'),'Croatia','Semi-Finals','3-0(Win)');
INSERT INTO Game VALUES('G7','R4',TO_DATE('2022-12-18',
'YYYY-MM-DD'),'France','Final','3(4) - 3(2)(Win)');


-- create table Game_Time
CREATE TABLE Game_Time
(
    Game_Time_ID VARCHAR(6) PRIMARY KEY,
    Game_ID VARCHAR(2),
    Eastern_Standard_Time_EST DATE,
```

```
    Central_Standard_Time_CST DATE,
    Mountain_Standard_Time_MST DATE,
    Pacific_Standard_Time_PST DATE,
    FOREIGN KEY (Game_ID) REFERENCES Game (Game_ID)
);


INSERT INTO Game_Time VALUES('GT101','G1',TO_DATE('05:00:00',
'HH24:MI:SS'),TO_DATE('04:00:00', 'HH24:MI:SS'),TO_DATE('03:00:00',
'HH24:MI:SS'),TO_DATE('02:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT102','G2',TO_DATE('14:00:00',
'HH24:MI:SS'),TO_DATE('13:00:00', 'HH24:MI:SS'),TO_DATE('12:00:00',
'HH24:MI:SS'),TO_DATE('11:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT103','G3',TO_DATE('14:00:00',
'HH24:MI:SS'),TO_DATE('13:00:00', 'HH24:MI:SS'),TO_DATE('12:00:00',
'HH24:MI:SS'),TO_DATE('11:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT104','G4',TO_DATE('08:00:00',
'HH24:MI:SS'),TO_DATE('07:00:00', 'HH24:MI:SS'),TO_DATE('06:00:00',
'HH24:MI:SS'),TO_DATE('05:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT105','G5',TO_DATE('14:00:00',
'HH24:MI:SS'),TO_DATE('13:00:00', 'HH24:MI:SS'),TO_DATE('12:00:00',
'HH24:MI:SS'),TO_DATE('11:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT106','G6',TO_DATE('14:00:00',
'HH24:MI:SS'),TO_DATE('13:00:00', 'HH24:MI:SS'),TO_DATE('12:00:00',
'HH24:MI:SS'),TO_DATE('11:00:00', 'HH24:MI:SS'));
INSERT INTO Game_Time VALUES('GT107','G7',TO_DATE('10:00:00',
'HH24:MI:SS'),TO_DATE('09:00:00', 'HH24:MI:SS'),TO_DATE('08:00:00',
'HH24:MI:SS'),TO_DATE('07:00:00', 'HH24:MI:SS'));
```

```
-- create table Player
CREATE TABLE Player (
   Player_ID VARCHAR(3) PRIMARY KEY,
   First_Name VARCHAR(20),
   Last_Name VARCHAR(20),
   Age INT,
   Position VARCHAR(15),
   Club_ID VARCHAR(3),
   Salary_USD INT,
   FOREIGN KEY (Club_ID) REFERENCES Club(Club_ID)
);

INSERT INTO Player VALUES('A10','Lionel','Messi',34,'Forward','101',20400000);
INSERT INTO Player VALUES('A11','Sergio','Aguero',33,'Forward','102',11970000);
insert into Player values('A12','Nicolas','Otamendi',33,'Midfielder','103',11970000);

insert into Player values('A13','Angel','Di Maria',33,'Midfielder','103',7690000);

insert into Player values('A14','Nahuel','Molina',23,'Defender','107',5207760);

insert into Player values('A15','Emiliano','Martinez',29,'Goalkeeper','104',8124105);

insert into Player values('A16','Enzo','Fernandez',26,'Midfielder','108',16784176);

insert into Player values('A17','Julian','Alvarez',21,'Forward','107',5641740);

insert into Player values('A18','Marcos','Acuna',29,'Midfielder','105',3030000);
```

```
insert into Player values('A19','Alexis','Mac Allister',24,'Midfielder','109',8462610);

insert into Player values('A20','Gonzalo','Montiel',26,'Defender','110',1490617);

insert into Player values('A21','Cristian','Romero',25,'Defender','111',9308871);

insert into Player values('A22','Lisandro','Martinez',25,'Defender','107',6770088);

insert into Player values('A23','Leandro','Paredes',29,'Midfielder','112',4339800);

insert into Player values('A24','German','Pezzella',32,'Defender','113',3679815);
commit;



-- create table Player Stats
CREATE TABLE Player_Stats (
   Player_ID VARCHAR(3) PRIMARY KEY,
   Goals_Scored INT,
   Assists INT,
   Yellow_Cards INT,
   Red_Cards INT
);

INSERT INTO Player_Stats VALUES('A10',7,3,1,0);
INSERT INTO Player_Stats VALUES('A11',3,0,2,0);
insert into Player_Stats values('A12',0,1,2,0);
insert into Player_Stats values('A13',1,1,0,0);
insert into Player_Stats values('A14',1,1,0,0);
insert into Player_Stats values('A15',0,0,1,0);
insert into Player_Stats values('A16',1,1,1,0);
```

```
insert into Player_Stats values('A17',4,0,0,0);
insert into Player_Stats values('A18',0,0,3,0);
insert into Player_Stats values('A19',1,1,0,0);
insert into Player_Stats values('A20',0,0,3,0);
insert into Player_Stats values('A21',0,0,2,0);
insert into Player_Stats values('A22',0,0,1,0);
insert into Player_Stats values('A23',0,0,2,0);
insert into Player_Stats values('A24',0,0,1,0);
commit;

-- create table Player History
CREATE TABLE Player_History (
   History_ID VARCHAR(3),
   Player_ID VARCHAR(3),
   Start_Date DATE,
   End_Date DATE,
   PRIMARY KEY (History_ID),
   FOREIGN KEY (Player_ID) REFERENCES Player (Player_ID)
);

INSERT INTO Player_History
VALUES('H1','A10',TO_DATE('2005-08-17','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H2','A11',TO_DATE('2006-09-03','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H3','A12',TO_DATE('2009-05-20','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H4','A13',TO_DATE('2008-09-06','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H5','A14',TO_DATE('2021-03-06','YYYY-MM-DD'), NULL);
```

```
INSERT INTO Player_History
VALUES('H6','A15',TO_DATE('2021-03-06','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H7','A16',TO_DATE('2022-09-24','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H8','A17',TO_DATE('2021-03-06','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H9','A18',TO_DATE('2016-11-15','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H10','A19',TO_DATE('2016-02-27','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H11','A20',TO_DATE('2019-03-22','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H12','A21',TO_DATE('2021-03-06','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H13','A22',TO_DATE('2019-03-21','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H14','A23',TO_DATE('2017-06-16','YYYY-MM-DD'), NULL);
INSERT INTO Player_History
VALUES('H15','A24',TO_DATE('2011-12-07','YYYY-MM-DD'), NULL);

-- create table Game Location
CREATE TABLE Game_Location (
   Game_Location_ID VARCHAR(3),
   Game_ID VARCHAR(3),
   Opponent VARCHAR(15),
   Location VARCHAR(22),
   PRIMARY KEY (Game_Location_ID),
   FOREIGN KEY (Game_ID) REFERENCES Game(Game_ID)
);
```

```
INSERT INTO Game_Location VALUES('GL1','G1','Saudi Arabia','Lusail Iconic
Stadium');
INSERT INTO Game_Location VALUES('GL2','G2','Mexico','Lusail Iconic Stadium');
INSERT INTO Game_Location VALUES('GL3','G3','Poland','Stadium 974');
INSERT INTO Game_Location VALUES('GL4','G4','Australia','Ahmad bin Ali
Stadium');
INSERT INTO Game_Location VALUES('GL5','G5','Netherlands','Lusail Iconic
Stadium');
INSERT INTO Game_Location VALUES('GL6','G6','Croatia','Lusail Iconic Stadium');
INSERT INTO Game_Location VALUES('GL7','G7','France','Lusail Iconic Stadium');


-- create table Attendance
create table Attendance
(Attendance_ID VARCHAR(4), Game_ID VARCHAR(3), Stadium_Name
VARCHAR(21), Stadium_Capacity INT, Stadium_Attendance INT, Television_Coverage
VARCHAR(3),
PRIMARY KEY (Attendance_ID),
FOREIGN KEY (Game_ID) REFERENCES Game(Game_ID));

insert into Attendance values('T101','G1','Lusail Iconic Stadium',88966,88012,'FS1');
insert into Attendance values('T102','G2','Lusail Iconic Stadium',88966,88966,'FS1');
insert into Attendance values('T103','G3','Stadium 974','44089',44089,'FOX');
insert into Attendance values('T104','G4','Ahmad bin Ali Stadium',45032,45032,'FOX');
insert into Attendance values('T105','G5','Lusail Iconic Stadium',88235,88235,'FOX');
insert into Attendance values('T106','G6','Lusail Iconic Stadium',88966,88966,'FOX');
insert into Attendance values('T107','G7','Lusail Iconic Stadium',88966,88966,'FOX');
commit;
```

```
-- create table Possession_Time
CREATE TABLE Possession_Time (
    Possession_ID VARCHAR(5),
    Game_ID VARCHAR(5),
    Possesion_Percentage NUMBER(5, 2),
    Opponent_Possesion_Percentage NUMBER(5, 2),
    Formation VARCHAR(12),
    Opponent_Formation VARCHAR(12),
    PRIMARY KEY (Possession_ID),
    FOREIGN KEY (Game_ID) REFERENCES Game(Game_ID)
);

-- Insert statements
INSERT INTO Possession_Time VALUES('P101', 'G1', 69.10, 30.90, '4-4-2', '4-1-4-1');
INSERT INTO Possession_Time VALUES('P102', 'G2', 58.50, 41.50, '4-4-2', '5-3-2');
INSERT INTO Possession_Time VALUES('P103', 'G3', 73.30, 26.70, '4-4-2', '4-3-3');
INSERT INTO Possession_Time VALUES('P104', 'G4', 60.70, 39.30, '4-3-3', '4-4-2');
INSERT INTO Possession_Time VALUES('P105', 'G5', 51.90, 48.10, '5-3-2', '3-4-1-2');
INSERT INTO Possession_Time VALUES('P106', 'G6', 39.20, 60.80, '4-4-2', '4-3-3');
INSERT INTO Possession_Time VALUES('P107', 'G7', 54.00, 46.00, '4-3-3', '4-2-3-1');

COMMIT;

-- create table PlayerStats_PerMatch
CREATE TABLE PlayerStats_PerMatch (
    Game_ID VARCHAR(3),
    Player_ID VARCHAR(3),
    GoalsThisGame INT,
    AssistsThisMatch INT,
    Yellow_Cards INT,
```

```
    Red_Cards INT,
    PRIMARY KEY (Game_ID, Player_ID),
    FOREIGN KEY (Player_ID) REFERENCES Player(Player_ID)
);


insert into PlayerStats_PerMatch values('G1','A10',1,0,0,0);
insert into PlayerStats_PerMatch values('G1','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A12',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A13',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A14',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A16',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A17',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A18',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A19',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A20',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A21',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A23',0,0,0,0);
insert into PlayerStats_PerMatch values('G1','A24',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A10',1,1,0,0);
insert into PlayerStats_PerMatch values('G2','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A12',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A13',0,1,0,0);
insert into PlayerStats_PerMatch values('G2','A14',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A16',1,0,0,0);
insert into PlayerStats_PerMatch values('G2','A17',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A18',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A19',0,0,0,0);
```

```
insert into PlayerStats_PerMatch values('G2','A20',0,0,1,0);
insert into PlayerStats_PerMatch values('G2','A21',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A23',0,0,0,0);
insert into PlayerStats_PerMatch values('G2','A24',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A10',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A12',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A13',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A14',0,1,0,0);
insert into PlayerStats_PerMatch values('G3','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A16',0,1,0,0);
insert into PlayerStats_PerMatch values('G3','A17',1,0,0,0);
insert into PlayerStats_PerMatch values('G3','A18',0,0,1,0);
insert into PlayerStats_PerMatch values('G3','A19',1,0,0,0);
insert into PlayerStats_PerMatch values('G3','A20',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A21',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A23',0,0,0,0);
insert into PlayerStats_PerMatch values('G3','A24',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A10',1,0,0,0);
insert into PlayerStats_PerMatch values('G4','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A12',0,1,0,0);
insert into PlayerStats_PerMatch values('G4','A13',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A14',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A16',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A17',1,0,0,0);
insert into PlayerStats_PerMatch values('G4','A18',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A19',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A20',0,0,0,0);
```

```
insert into PlayerStats_PerMatch values('G4','A21',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A23',0,0,0,0);
insert into PlayerStats_PerMatch values('G4','A24',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A10',1,1,1,0);
insert into PlayerStats_PerMatch values('G5','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A12',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A13',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A14',1,0,0,0);
insert into PlayerStats_PerMatch values('G5','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A16',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A17',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A18',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A19',0,0,0,0);
insert into PlayerStats_PerMatch values('G5','A20',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A21',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A22',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A23',0,0,1,0);
insert into PlayerStats_PerMatch values('G5','A24',0,0,1,0);
insert into PlayerStats_PerMatch values('G6','A10',1,1,0,0);
insert into PlayerStats_PerMatch values('G6','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A12',0,0,1,0);
insert into PlayerStats_PerMatch values('G6','A13',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A14',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A15',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A16',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A17',2,0,0,0);
insert into PlayerStats_PerMatch values('G6','A18',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A19',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A20',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A21',0,0,1,0);
```

```
insert into PlayerStats_PerMatch values('G6','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A23',0,0,0,0);
insert into PlayerStats_PerMatch values('G6','A24',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A10',2,0,0,0);
insert into PlayerStats_PerMatch values('G7','A11',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A12',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A13',1,0,0,0);
insert into PlayerStats_PerMatch values('G7','A14',2,0,0,0);
insert into PlayerStats_PerMatch values('G7','A15',0,0,1,0);
insert into PlayerStats_PerMatch values('G7','A16',0,0,1,0);
insert into PlayerStats_PerMatch values('G7','A17',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A18',0,0,1,0);
insert into PlayerStats_PerMatch values('G7','A19',1,1,0,0);
insert into PlayerStats_PerMatch values('G7','A20',0,0,1,0);
insert into PlayerStats_PerMatch values('G7','A21',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A22',0,0,0,0);
insert into PlayerStats_PerMatch values('G7','A23',0,0,1,0);
insert into PlayerStats_PerMatch values('G7','A24',0,0,0,0);
commit;
```

BLOCK 1. 104 lines - this query creates a cursor that acts as a mechanism to display all of the information being requested. In this case after the cursor is created the output being requested is the players general information. Included is the total number of all player stats, from total goals to the total salary.

SQL:

```
DECLARE
   CURSOR cur_players IS -- this is a cursor for all player
      SELECT
         pl.First_Name || ' ' || pl.Last_Name AS player_name,
         cl.Club_Name AS team_name,
         pl.Position AS position_name,
         ps.Goals_Scored AS goals_scored,
         ps.Assists AS assists,
         ps.Yellow_Cards AS yellow_cards,
         ps.Red_Cards AS red_cards,
         pl.Salary_USD AS salary
      FROM
         Player pl
      JOIN
         Club cl ON
         pl.Club_ID = cl.Club_ID
      JOIN
         Player_Stats ps ON
         pl.Player_ID = ps.Player_ID;

   player_name VARCHAR2(100);
   team_name VARCHAR2(100);
   position_name VARCHAR2(50);
   goals_scored NUMBER;
```

```
    assists NUMBER;
    yellow_cards NUMBER;
    red_cards NUMBER;
    salary NUMBER;

    total_players NUMBER := 0; -- set as zero to start
    total_goals NUMBER := 0; -- set total goals as zero to start counter
    total_assists NUMBER := 0; -- set total assists as zero start counter
    total_yellow_cards NUMBER := 0; -- set total yellow cards as zero start counter
    total_red_cards NUMBER := 0; -- set total red cards as zero start counter
    total_salary NUMBER := 0; -- set total salary as zero
    max_salary_player VARCHAR2(50); -- declaring max player salary
    min_salary_player VARCHAR2(50); -- declaring min player salary
    max_salary NUMBER := 0; --set max salary as zero
    min_salary NUMBER := 999999999; -- min salary needs value to count
    average_salary NUMBER := 0; --set average salary to zero
BEGIN
    dbms_output.put_line('Player Name | Team Name | Position | Goals | Assists | Yellow Cards |
Red Cards | Salary'); -- display player info

dbms_output.put_line('---------------------------------------------------------------------------------------------
-------');

    FOR cur IN cur_players LOOP -- start loop to retrieve cursor info
        player_name := cur.player_name;
        team_name := cur.team_name;
        position_name := cur.position_name;
        goals_scored := cur.goals_scored;
        assists := cur.assists;
        yellow_cards := cur.yellow_cards;
        red_cards := cur.red_cards;
```

```
    salary := cur.salary;

    dbms_output.put_line(
        RPAD(player_name, 30) || ' | ' ||
        RPAD(team_name, 25) || ' | ' ||
        RPAD(position_name, 10) || ' | ' ||
        TO_CHAR(goals_scored, '99999') || ' | ' ||
        TO_CHAR(assists, '99999') || ' | ' ||
        TO_CHAR(yellow_cards, '99999') || ' | ' ||
        TO_CHAR(red_cards, '99999') || ' | ' ||
        TO_CHAR(salary, '$999,999,999.99')
    );

    total_players := total_players + 1; -- start player counter
    total_goals := total_goals + goals_scored; -- calc for total goals
    total_assists := total_assists + assists; -- calc for total assists
    total_yellow_cards := total_yellow_cards + yellow_cards; -- calc for total yellow cards
    total_red_cards := total_red_cards + red_cards; -- calc for total red cards
    total_salary := total_salary + salary; -- calc for total salary


    IF salary > max_salary THEN -- if statement to find max salary
        max_salary := salary;
        max_salary_player := player_name;
    END IF;

    IF salary < min_salary THEN -- if statement to find min salary
        min_salary := salary;
        min_salary_player := player_name;
    END IF;
END LOOP;
```

```
    IF total_players > 0 THEN -- display output
        average_salary := total_salary / total_players; -- calc for average salary



dbms_output.put_line('-------------------------------------------------------------------------------------------
-------');
        dbms_output.put_line('Total Players: ' || total_players);
        dbms_output.put_line('Total Goals: ' || total_goals);
        dbms_output.put_line('Total Assists: ' || total_assists);
        dbms_output.put_line('Total Yellow Cards: ' || total_yellow_cards);
        dbms_output.put_line('Total Red Cards: ' || total_red_cards);
        dbms_output.put_line('Total Salary: ' || TO_CHAR(total_salary, '$999,999,999.99'));
        dbms_output.put_line('Average Salary: ' || TO_CHAR(average_salary, '$999,999,999.99'));
        dbms_output.put_line('Player with Highest Salary: ' || max_salary_player || ' ($' ||
TO_CHAR(max_salary, '999,999,999.99') || ')');
        dbms_output.put_line('Player with Lowest Salary: ' || min_salary_player || ' ($' ||
TO_CHAR(min_salary, '999,999,999.99') || ')');
    ELSE
        dbms_output.put_line('No players found.');
    END IF;
END;
/
```

OUTPUT:

```
Statement processed.
Player Name | Team Name | Position | Goals | Assists | Yellow Cards | Red Cards | Salary
-----------------------------------------------------------------------------------------------
Lionel Messi                | Inter Miami              | Forward    |       7 |       3 |       1 |       0 |   $20,400,000.00
Sergio Aguero               | FC Barcelona             | Forward    |       3 |       0 |       2 |       0 |   $11,970,000.00
Nicolas Otamendi            | S.L. Benfica             | Midfielder |       0 |       1 |       2 |       0 |   $11,970,000.00
Angel Di Maria              | S.L. Benfica             | Midfielder |       1 |       1 |       0 |       0 |    $7,690,000.00
Nahuel Molina               | Manchester City          | Defender   |       1 |       1 |       0 |       0 |    $5,207,760.00
Emiliano Martinez           | Aston Villa              | Goalkeeper |       0 |       0 |       1 |       0 |    $8,124,105.00
Enzo Fernandez              | Chelsea                  | Midfielder |       1 |       1 |       1 |       0 |   $16,784,176.00
Julian Alvarez              | Manchester City          | Forward    |       4 |       0 |       0 |       0 |    $5,641,740.00
Marcos Acuna                | Sevilla FC               | Midfielder |       0 |       0 |       3 |       0 |    $3,030,000.00
Alexis Mac Allister         | Liverpool                | Midfielder |       1 |       1 |       0 |       0 |    $8,462,610.00
Gonzalo Montiel             | Nottingham Forest F.C.   | Defender   |       0 |       0 |       3 |       0 |    $1,490,617.00
Cristian Romero             | Tottenham Hotspur F.C.   | Defender   |       0 |       0 |       2 |       0 |    $9,308,871.00
Lisandro Martinez           | Manchester City          | Defender   |       0 |       0 |       1 |       0 |    $6,770,088.00
Leandro Paredes             | A.S. Roma                | Midfielder |       0 |       0 |       2 |       0 |    $4,339,800.00
German Pezzella             | Real Betis               | Defender   |       0 |       0 |       1 |       0 |    $3,679,815.00
-----------------------------------------------------------------------------------------------
Total Players: 15
Total Goals: 18
Total Assists: 8
Total Yellow Cards: 19
Total Red Cards: 0
Total Salary:  $124,869,582.00
Average Salary:    $8,324,638.80
Player with Highest Salary: Lionel Messi ($  20,400,000.00)
Player with Lowest Salary: Gonzalo Montiel ($   1,490,617.00)
```

BLOCK 2 - 94 lines this aims to create two separate cursors to be referenced later, one in which is the lowest paid player based on salary, as the second cursor is the highest paid player. This query seeks to see the gap between the players salaries. It separates the output into the top 5 lowest and top 5 highest paid. Their respective salaries are compared to the team average to see how much more or less they are making than their teammates.

SQL:

```
DECLARE
   CURSOR cur_lowest_paid_players IS  -- 1st Cursor for the lowest paid players
      SELECT
         pl.First_Name || ' ' || pl.Last_Name AS player_name,
         pl.Salary_USD AS salary
      FROM
         Player pl
      ORDER BY
         pl.Salary_USD ASC;

   CURSOR cur_highest_paid_players IS -- 2nd Cursor for the highest paid players
      SELECT
         pl.First_Name || ' ' || pl.Last_Name AS player_name,
         pl.Salary_USD AS salary
      FROM
         Player pl
      ORDER BY
         pl.Salary_USD DESC;

   player_name VARCHAR2(100);
   salary NUMBER;
   total_salary NUMBER := 0; -- set total salary to zero
   average_salary NUMBER := 0; -- set average salary to zero
   raise_needed NUMBER := 0; -- set raise needed salary to zero
   counter NUMBER := 0;
BEGIN
   dbms_output.put_line('Lowest Paid Players and Their Salaries:');
```

```
    dbms_output.put_line('---------------------------------------');

    SELECT AVG(Salary_USD) INTO average_salary FROM Player; -- finds average salary from player table

    FOR cur IN cur_lowest_paid_players LOOP -- starts loop to find top 5 lowest paid
        counter := counter + 1; -- starts counter
        IF counter <= 5 THEN
            player_name := cur.player_name;
            salary := cur.salary; -- current salary

            raise_needed := average_salary - salary; -- calc raise needed to reach avg

            dbms_output.put_line('Current Salary for ' || player_name || ': ' || TO_CHAR(salary, '$999,999,999.99'));
            dbms_output.put_line('Potential Raise for ' || player_name || ': ' || TO_CHAR(raise_needed, '$999,999,999.99'));

            dbms_output.put_line(RPAD(player_name, 50));
            dbms_output.put_line('');

            total_salary := total_salary + salary; -- calc total salary
        ELSE
            EXIT;
        END IF;
    END LOOP;

    IF counter > 0 THEN
        raise_needed := (5 * average_salary) - total_salary;

        dbms_output.put_line('---------------------------------------');
        dbms_output.put_line('Average Overall Player Salary: ' || TO_CHAR(average_salary, '$999,999,999.99'));
```

```
        dbms_output.put_line('Total Capital to Reach Player Average: ' || TO_CHAR(raise_needed,
'$999,999,999.99'));
    ELSE
        dbms_output.put_line('No players found.');
    END IF;

    dbms_output.put_line('');
            dbms_output.put_line('---------------------------------------');
    dbms_output.put_line('Highest Paid Players and Their Salaries:');
            dbms_output.put_line('---------------------------------------');

    counter := 0;
    total_salary := 0;

    FOR cur IN cur_highest_paid_players LOOP -- start loop for top 5 highest paid
        counter := counter + 1; -- start counter
        IF counter <= 5 THEN
            player_name := cur.player_name;
            salary := cur.salary;
            total_salary := total_salary + salary;

            raise_needed := average_salary - salary;
                                dbms_output.put_line('Current Salary for ' || player_name || ': ' ||
TO_CHAR(salary, '$999,999,999.99'));
            dbms_output.put_line('Potential Pay Cut to reach Player Average Salary ' || player_name || ': ' ||
TO_CHAR(raise_needed, '$999,999,999.99'));
            dbms_output.put_line('');
        ELSE
            EXIT;
        END IF;
    END LOOP;

    IF counter > 0 THEN
        raise_needed := (5 * average_salary) - total_salary;
```

```
        dbms_output.put_line('---------------------------------------');
        dbms_output.put_line('Average Overall Player Salary: ' || TO_CHAR(average_salary,
'$999,999,999.99'));
        dbms_output.put_line('Total Capital from Average: ' || TO_CHAR(raise_needed, '$999,999,999.99'));
    ELSE
        dbms_output.put_line('No players found.');
    END IF;
END;
/
```

OUTPUT:

```
Lowest Paid Players and Their Salaries:
---------------------------------------
Current Salary for Gonzalo Montiel:    $1,490,617.00
Potential Raise for Gonzalo Montiel:    $6,834,021.80
Gonzalo Montiel
Current Salary for Marcos Acuna:    $3,030,000.00
Potential Raise for Marcos Acuna:    $5,294,638.80
Marcos Acuna
Current Salary for German Pezzella:    $3,679,815.00
Potential Raise for German Pezzella:    $4,644,823.80
German Pezzella
Current Salary for Leandro Paredes:    $4,339,800.00
Potential Raise for Leandro Paredes:    $3,984,838.80
Leandro Paredes
Current Salary for Nahuel Molina:    $5,207,760.00
Potential Raise for Nahuel Molina:    $3,116,878.80
Nahuel Molina
---------------------------------------
Average Overall Player Salary:    $8,324,638.80
Total Capital to Reach Player Average:   $23,875,202.00
---------------------------------------
Highest Paid Players and Their Salaries:
---------------------------------------
Current Salary for Lionel Messi:   $20,400,000.00
Potential Pay Cut to reach Player Average Salary Lionel Messi:  -$12,075,361.20
Current Salary for Enzo Fernandez:   $16,784,176.00
Potential Pay Cut to reach Player Average Salary Enzo Fernandez:   -$8,459,537.20
Current Salary for Sergio Aguero:   $11,970,000.00
Potential Pay Cut to reach Player Average Salary Sergio Aguero:   -$3,645,361.20
Current Salary for Nicolas Otamendi:   $11,970,000.00
Potential Pay Cut to reach Player Average Salary Nicolas Otamendi:   -$3,645,361.20
Current Salary for Cristian Romero:    $9,308,871.00
Potential Pay Cut to reach Player Average Salary Cristian Romero:     -$984,232.20
---------------------------------------
Average Overall Player Salary:    $8,324,638.80
Total Capital from Average:  -$28,809,853.00
```

BLOCK 3: - 101 lines this query aims to seek a possible relationship or correlation between the highest goal scorer and his salary versus the lowest goal scorer and their respective salary. Making sure to exclude goalkeepers since they will likely never score.

SQL:

```
DECLARE
   TYPE player_info_rec IS RECORD ( -- creates the record for cursor data to go into
      player_name VARCHAR2(40),
      club_name VARCHAR2(22),
      country VARCHAR2(18),
      league VARCHAR2(19),
      position VARCHAR2(15),
      salary_usd NUMBER,
      total_goals NUMBER
   );

   top_player_info player_info_rec;
   lowest_player_info player_info_rec;
BEGIN

   SELECT *
   INTO top_player_info -- top player in goals and salary
   FROM (
      SELECT
         p.First_Name || ' ' || p.Last_Name AS player_name,
         c.Club_Name,
         c.Country,
         c.League,
         p.Position,
         p.Salary_USD,
         SUM(ps.GoalsThisGame) AS total_goals -- total goals of x game
      FROM
```

```
        PlayerStats_PerMatch ps,
        Player p,
        Club c
    WHERE
        ps.Player_ID = p.Player_ID
        AND p.Club_ID = c.Club_ID
        AND p.Position != 'Goalkeeper' -- excludes goalkeepers
    GROUP BY
        p.First_Name || ' ' || p.Last_Name, p.Position, c.Club_Name, c.Country, c.League,
p.Salary_USD
    ORDER BY
        total_goals DESC, p.Salary_USD DESC -- list in top goal scorer and salary in
descending order
  )
  WHERE ROWNUM = 1; -- first in that row

  SELECT *
  INTO lowest_player_info -- top lowest player in goals and salary
  FROM (
    SELECT
        p.First_Name || ' ' || p.Last_Name AS player_name,
        c.Club_Name,
        c.Country,
        c.League,
        p.Position,
        p.Salary_USD,
        SUM(ps.GoalsThisGame) AS total_goals -- finds total goals
    FROM
        PlayerStats_PerMatch ps,
        Player p,
        Club c
```

```
        WHERE
            ps.Player_ID = p.Player_ID
            AND p.Club_ID = c.Club_ID
            AND p.Position != 'Goalkeeper'
        GROUP BY
            p.First_Name || ' ' || p.Last_Name, p.Position, c.Club_Name, c.Country, c.League,
p.Salary_USD
        ORDER BY
            total_goals ASC, p.Salary_USD ASC -- in ascending order therefore lowest on top
    )
    WHERE ROWNUM = 1; -- select first in row

    IF top_player_info.player_name IS NOT NULL THEN
        dbms_output.put_line('Top Goal Scorers (Non-Goalkeepers) with Highest Salaries:');
        dbms_output.put_line('-------------------------------------------------------');

        dbms_output.put_line('Player Name: ' || top_player_info.player_name); -- output for top
player
        dbms_output.put_line('Club: ' || top_player_info.club_name);
        dbms_output.put_line('Country: ' || top_player_info.country);
        dbms_output.put_line('League: ' || top_player_info.league);
        dbms_output.put_line('Position: ' || top_player_info.position);
        dbms_output.put_line('Salary: $' || TO_CHAR(top_player_info.salary_usd, '999,999,999'));
        dbms_output.put_line('Total Goals: ' || top_player_info.total_goals);
        dbms_output.put_line('');
    ELSE
        dbms_output.put_line('No data found for top goal scorers (non-goalkeepers).');
    END IF;

    IF lowest_player_info.player_name IS NOT NULL THEN
        dbms_output.put_line('-------------------------------------------------------');
```

```
        dbms_output.put_line('Lowest Goal Scorers (Non-Goalkeepers) with Lowest Salaries:');
        dbms_output.put_line('-------------------------------------------------------');

        dbms_output.put_line('Player Name: ' || lowest_player_info.player_name); -- output for
lowest players
        dbms_output.put_line('Club: ' || lowest_player_info.club_name);
        dbms_output.put_line('Country: ' || lowest_player_info.country);
        dbms_output.put_line('League: ' || lowest_player_info.league);
        dbms_output.put_line('Position: ' || lowest_player_info.position);
        dbms_output.put_line('Salary: $' || TO_CHAR(lowest_player_info.salary_usd,
'999,999,999'));
        dbms_output.put_line('Total Goals: ' || lowest_player_info.total_goals);
        dbms_output.put_line('');
    ELSE
        dbms_output.put_line('No data found for lowest goal scorers (non-goalkeepers).');
    END IF;
END;
/
```

OUTPUT:

```
Statement processed.
Top Goal Scorers (Non-Goalkeepers) with Highest Salaries:
------------------------------------------------------------
Player Name: Lionel Messi
Club: Inter Miami
Country: United States
League: Major League Soccer
Position: Forward
Salary: $  20,400,000
Total Goals: 7
------------------------------------------------------------
Lowest Goal Scorers (Non-Goalkeepers) with Lowest Salaries:
------------------------------------------------------------
Player Name: Gonzalo Montiel
Club: Nottingham Forest F.C.
Country: England
League: Premier League
Position: Defender
Salary: $   1,490,617
Total Goals: 0
```

BLOCK 4: - 83 lines goal is to create a fictitious player that is a midfielder and calculate what his average stats would be given the stats of the other midfielders as well as calculating which club out of the clubs listed he would get paid the most at

SQL:

```
DECLARE
    v_player_id VARCHAR2(3);
    v_first_name VARCHAR2(20);
    v_last_name VARCHAR2(20);
    v_age INT;
    v_position VARCHAR2(15);
    v_club_id VARCHAR2(3);
    v_salary_usd INT;
    v_goals_scored INT;
    v_assists INT;
    v_yellow_cards INT;
    v_red_cards INT;
    v_player_exists INT;
    v_max_salary INT;
    v_max_salary_club VARCHAR2(3);
    v_max_salary_club_name VARCHAR2(22);
    v_max_salary_club_country VARCHAR2(18);
BEGIN
    v_player_id := 'A25'; -- new player id insertion
    v_first_name := 'John'; -- new player first name insertion
    v_last_name := 'Doe'; -- new player last nameinsertion
    v_age := 25; -- new player age insertion
    v_position := 'Midfielder'; -- new player position insertion
    v_club_id := '101'; -- new player club id insertion
    v_salary_usd := 1500000; -- new player salary insertion
    SELECT AVG(Goals_Scored), AVG(Assists), AVG(Yellow_Cards), AVG(Red_Cards) -- calc total stats to give new player the avg
```

```
    INTO v_goals_scored, v_assists, v_yellow_cards, v_red_cards
    FROM Player_Stats
    WHERE Player_ID IN (SELECT Player_ID FROM Player WHERE Position = 'Midfielder');

    SELECT COUNT(*)
    INTO v_player_exists
    FROM Player
    WHERE Player_ID = v_player_id;

    IF v_player_exists = 0 THEN --insert statements
        INSERT INTO Player (Player_ID, First_Name, Last_Name, Age, Position, Club_ID,
Salary_USD)
        VALUES (v_player_id, v_first_name, v_last_name, v_age, v_position, v_club_id,
v_salary_usd);

        INSERT INTO Player_Stats (Player_ID, Goals_Scored, Assists, Yellow_Cards, Red_Cards)
        VALUES (v_player_id, v_goals_scored, v_assists, v_yellow_cards, v_red_cards);

        DBMS_OUTPUT.PUT_LINE('Player Information:');
        DBMS_OUTPUT.PUT_LINE('--------------------');
        DBMS_OUTPUT.PUT_LINE('Player ID: ' || v_player_id);
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_first_name || ' ' || v_last_name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || TO_CHAR(v_age));
        DBMS_OUTPUT.PUT_LINE('Position: ' || v_position);
        DBMS_OUTPUT.PUT_LINE('Club ID: ' || v_club_id);
        DBMS_OUTPUT.PUT_LINE('Salary (USD): $' || TO_CHAR(v_salary_usd));
        DBMS_OUTPUT.PUT_LINE('Goals Scored: ' || TO_CHAR(v_goals_scored));
        DBMS_OUTPUT.PUT_LINE('Assists: ' || TO_CHAR(v_assists));
        DBMS_OUTPUT.PUT_LINE('Yellow Cards: ' || TO_CHAR(v_yellow_cards));
        DBMS_OUTPUT.PUT_LINE('Red Cards: ' || TO_CHAR(v_red_cards));
    ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Player ID ' || v_player_id || ' already exists in the database.');
  END IF;

  SELECT Salary_USD INTO v_max_salary FROM Player WHERE Player_ID = v_player_id;

  SELECT Club_ID, Club_Name, Country INTO v_max_salary_club,
v_max_salary_club_name, v_max_salary_club_country
  FROM (
    SELECT p.Club_ID, c.Club_Name, c.Country, AVG(p.Salary_USD) AS avg_salary
    FROM Player p
    JOIN Club c ON p.Club_ID = c.Club_ID
    GROUP BY p.Club_ID, c.Club_Name, c.Country
    ORDER BY avg_salary DESC
  )
  WHERE ROWNUM = 1;
  DBMS_OUTPUT.PUT_LINE('');
  DBMS_OUTPUT.PUT_LINE('Club Information:');
  DBMS_OUTPUT.PUT_LINE('-------------------');
  DBMS_OUTPUT.PUT_LINE('Maximum Salary: $' || TO_CHAR(v_max_salary));
  DBMS_OUTPUT.PUT_LINE('Likely Club with Maximum Salary:');
  DBMS_OUTPUT.PUT_LINE('  Club ID: ' || v_max_salary_club);
  DBMS_OUTPUT.PUT_LINE('  Club Name: ' || v_max_salary_club_name);
  DBMS_OUTPUT.PUT_LINE('  Country: ' || v_max_salary_club_country);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An unexpected error occurred. Please contact the system
administrator.');
END;
/
```

OUTPUT:

```
Statement processed.
Player Information:
-------------------
Player ID: A25
Name: John Doe
Age: 25
Position: Midfielder
Club ID: 101
Salary (USD): $1500000
Goals Scored: 1
Assists: 1
Yellow Cards: 1
Red Cards: 0
Club Information:
------------------
Maximum Salary: $1500000
Likely Club with Maximum Salary:
  Club ID: 108
  Club Name: Chelsea
  Country: England
```

2ND RUN OUTPUT:

```
Statement processed.
Player ID A25 already exists in the database.
Club Information:
------------------
Maximum Salary: $1500000
Likely Club with Maximum Salary:
  Club ID: 108
  Club Name: Chelsea
  Country: England
```

BLOCK 5 104 lines - this query aims to create cursors to separate the best performance and worst performance on a statistics basis for game 7. It shuffles through the dataset, displaying the result of the game along with every other player that did not score in game 7.

SQL:

```
DECLARE
   v_game_id Game.Game_ID%TYPE;
   v_player_id Player.Player_ID%TYPE;
   v_player_name VARCHAR2(50);
   v_player_position VARCHAR2(50);
   v_competition_name VARCHAR2(50);
   v_game_result VARCHAR2(20);
   v_least_goals PlayerStats_PerMatch.GoalsThisGame%TYPE;
   v_least_assists PlayerStats_PerMatch.AssistsThisMatch%TYPE;
   v_least_player_name VARCHAR2(50);
   v_most_assists_player_id Player.Player_ID%TYPE;
   v_most_assists_player_name VARCHAR2(50);
   v_most_assists NUMBER;

   CURSOR c_best_performance IS -- cursor to find player with best performance in game 7
      SELECT g.Game_ID, p.Player_ID, p.First_Name || ' ' || p.Last_Name,
           p.Position,
           g.Competition, g.Results
      FROM Game g
      JOIN PlayerStats_PerMatch ps ON
      g.Game_ID = ps.Game_ID
      JOIN Player p ON
      ps.Player_ID = p.Player_ID
      WHERE g.Game_ID = 'G7' -- only for game 7
      ORDER BY ps.GoalsThisGame DESC, ps.AssistsThisMatch DESC; -- goals and assists
listed in descending order
```

```
    CURSOR c_least_performance IS -- cursor for worst performance in game 7
        SELECT g.Game_ID, p.Player_ID, p.First_Name || ' ' || p.Last_Name,
            p.Position,
            ps.GoalsThisGame, ps.AssistsThisMatch
        FROM Game g
        JOIN PlayerStats_PerMatch ps ON g.Game_ID = ps.Game_ID
        JOIN Player p ON ps.Player_ID = p.Player_ID
        WHERE g.Game_ID = 'G7' -- only from game 7
        ORDER BY ps.GoalsThisGame ASC, ps.AssistsThisMatch ASC; -- goals and assists in
ascending order

    CURSOR c_most_assists_player IS -- cursor for most assists
        SELECT p.Player_ID, p.First_Name || ' ' || p.Last_Name, COUNT(ps.AssistsThisMatch) AS
assists_count -- assist counter
        FROM Player p
        JOIN PlayerStats_PerMatch ps ON p.Player_ID = ps.Player_ID
        WHERE ps.AssistsThisMatch IS NOT NULL
        GROUP BY p.Player_ID, p.First_Name, p.Last_Name
        ORDER BY COUNT(ps.AssistsThisMatch) DESC; -- list in descending order

    CURSOR c_players_no_goals_in_game7 IS -- cursor for players with no goals in game 7
        SELECT DISTINCT p.Player_ID, p.First_Name, p.Last_Name, p.Position
        FROM Player p
        WHERE p.Player_ID NOT IN
        (SELECT ps.Player_ID FROM PlayerStats_PerMatch ps WHERE ps.Game_ID = 'G7'
AND ps.GoalsThisGame > 0)
        ORDER BY p.Player_ID;

BEGIN
    OPEN c_best_performance; -- open fetch for best performance
```

```
    FETCH c_best_performance INTO v_game_id, v_player_id, v_player_name,
v_player_position, v_competition_name, v_game_result;

    DBMS_OUTPUT.PUT_LINE('Player with the best performance in Game 7:');
    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Player ID: ' || v_player_id);
    DBMS_OUTPUT.PUT_LINE('Player Name: ' || v_player_name);
    DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_position);
    DBMS_OUTPUT.PUT_LINE('Competition Name: ' || v_competition_name);
    DBMS_OUTPUT.PUT_LINE('Game Result: ' || v_game_result);
    DBMS_OUTPUT.PUT_LINE('');

    CLOSE c_best_performance; -- close best performance fetch

    OPEN c_least_performance; -- open fetch for worst performance
    FETCH c_least_performance INTO v_game_id, v_player_id, v_least_player_name,
v_player_position, v_least_goals, v_least_assists;

    DBMS_OUTPUT.PUT_LINE('Player with the worst performance in Game 7:');
    DBMS_OUTPUT.PUT_LINE('-------------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Player Name: ' || v_least_player_name);
    DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_position);
    DBMS_OUTPUT.PUT_LINE('Goals: ' || v_least_goals);
    DBMS_OUTPUT.PUT_LINE('Assists: ' || v_least_assists);

    CLOSE c_least_performance; -- close fetch for worst performance

    OPEN c_most_assists_player; -- open fetch for most assists
    FETCH c_most_assists_player INTO v_most_assists_player_id, v_most_assists_player_name,
v_most_assists;
    CLOSE c_most_assists_player; -- close fetch
```

```
    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE('Player with the most assists in all games:'); -- display player with most assists
    DBMS_OUTPUT.PUT_LINE('----------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Player Name: ' || v_most_assists_player_name);
    DBMS_OUTPUT.PUT_LINE('Assists Count: ' || v_most_assists);

    DBMS_OUTPUT.PUT_LINE('');
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Players who did not score in Game 7:');
    DBMS_OUTPUT.PUT_LINE('-------------------------------------');

    FOR player_rec IN c_players_no_goals_in_game7 LOOP
        DBMS_OUTPUT.PUT_LINE('Player Name: ' || player_rec.First_Name || ' ' || player_rec.Last_Name || ' '|| 'Position: ' || player_rec.Position);
    END LOOP;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No data found for Game 7.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An unexpected error occurred. Please contact the system administrator.');
END;
/
```

OUTPUT:

```
Statement processed.
Player with the best performance in Game 7:
---------------------------------------
Player ID: A10
Player Name: Lionel Messi
Position: Forward
Competition Name: Final
Game Result: 3(4) - 3(2)(Win)
Player with the least performance in Game 7:
-------------------------------------------
Player Name: Sergio Aguero
Position: Forward
Goals: 0
Assists: 0
Player with the most assists in all games:
-----------------------------------------
Player Name: Lionel Messi
Assists Count: 7
Players who did not score in Game 7:
-----------------------------------
Player Name: Sergio Aguero || Position: Forward
Player Name: Nicolas Otamendi || Position: Midfielder
Player Name: Emiliano Martinez || Position: Goalkeeper
Player Name: Marcos Acuna || Position: Midfielder
Player Name: Gonzalo Montiel || Position: Defender
Player Name: Cristian Romero || Position: Defender
Player Name: Lisandro Martinez || Position: Defender
Player Name: Leandro Paredes || Position: Midfielder
Player Name: German Pezzella || Position: Defender
```

BLOCK 6 79 lines - creates cursors for most goals, assists and yellow cards of game 6. Then prints the players information which reflects the usual 'norm' of soccer… in which the forwards are getting goals/assists, while the defenders are getting yellow cards.

SQL:

```
DECLARE
   CURSOR goals_cursor IS -- creates cursor for highest goals in game 6
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G6'
      ORDER BY GoalsThisGame DESC; -- highest in row, descending order

   CURSOR assists_cursor IS -- creates cursor for highest assists in game 6
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G6'
      ORDER BY AssistsThisMatch DESC; -- highest in row, descending order

   CURSOR yellow_cards_cursor IS  -- creates cursor for highest yellow cards in game 6
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G6'
      ORDER BY Yellow_Cards DESC; -- highest in row, descending order

   v_player_most_goals_g6 PlayerStats_PerMatch%ROWTYPE;
   v_player_most_assists_g6 PlayerStats_PerMatch%ROWTYPE;
   v_player_most_yellow_cards_g6 PlayerStats_PerMatch%ROWTYPE;
   v_player_info Player%ROWTYPE;
BEGIN
   OPEN goals_cursor; --open goal cursor fetch
   FETCH goals_cursor INTO v_player_most_goals_g6;
   CLOSE goals_cursor; -- close fetch
```

```
    SELECT *
    INTO v_player_info
    FROM Player
    WHERE Player_ID = v_player_most_goals_g6.Player_ID;

    IF v_player_most_goals_g6.Player_ID IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Game 6 - Player with the most goals:');
                    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
        DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
        DBMS_OUTPUT.PUT_LINE('Goals Scored: ' || v_player_most_goals_g6.GoalsThisGame);
        DBMS_OUTPUT.PUT_LINE('');
    END IF;

    OPEN assists_cursor;
    FETCH assists_cursor INTO v_player_most_assists_g6;
    CLOSE assists_cursor;

    SELECT *
    INTO v_player_info
    FROM Player
    WHERE Player_ID = v_player_most_assists_g6.Player_ID;

    IF v_player_most_assists_g6.Player_ID IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Game 6 - Player with the most assists:');
                    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
```

```
    DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
    DBMS_OUTPUT.PUT_LINE('Assists: ' || v_player_most_assists_g6.AssistsThisMatch);
    DBMS_OUTPUT.PUT_LINE('');
  END IF;

  OPEN yellow_cards_cursor;
  FETCH yellow_cards_cursor INTO v_player_most_yellow_cards_g6;
  CLOSE yellow_cards_cursor;

  SELECT *
  INTO v_player_info
  FROM Player
  WHERE Player_ID = v_player_most_yellow_cards_g6.Player_ID;

  IF v_player_most_yellow_cards_g6.Player_ID IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Game 6 - Player with the most yellow cards:');
            DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
    DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
    DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
    DBMS_OUTPUT.PUT_LINE('Yellow Cards: ' || v_player_most_yellow_cards_g6.Yellow_Cards);
    DBMS_OUTPUT.PUT_LINE('');
  END IF;
END;
/
```

OUTPUT:

```
Statement processed.
--------------------------------------
Game 6 - Player with the most goals:
--------------------------------------
Name: Julian Alvarez
Age: 21
Position: Forward
Goals Scored: 2
--------------------------------------
Game 6 - Player with the most assists:
--------------------------------------
Name: Lionel Messi
Age: 34
Position: Forward
Assists: 1
--------------------------------------
Game 6 - Player with the most yellow cards:
--------------------------------------
Name: Cristian Romero
Age: 25
Position: Defender
Yellow Cards: 1
```

BLOCK 7 - 79 lines discover the top stat receivers within game 2. Does this by separating into three different queries all aimed to find the highest number for that specific cursor. Therefore the goals cursor finds the highest amount of goals, as the assist cursor finds the highest amount of assists, etc.

SQL:

```
DECLARE
   CURSOR goals_cursor_g2 IS -- cursor for highest goals in game 2
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G2'
      ORDER BY GoalsThisGame DESC;

   CURSOR assists_cursor_g2 IS --cursor for highest assists in game 2
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G2'
      ORDER BY AssistsThisMatch DESC;

   CURSOR yellow_cards_cursor_g2 IS -- cursor for highest yellow cards in game 2
      SELECT *
      FROM PlayerStats_PerMatch
      WHERE Game_ID = 'G2'
      ORDER BY Yellow_Cards DESC;

   v_player_most_goals_g2 PlayerStats_PerMatch%ROWTYPE;
   v_player_most_assists_g2 PlayerStats_PerMatch%ROWTYPE;
   v_player_most_yellow_cards_g2 PlayerStats_PerMatch%ROWTYPE;
   v_player_info Player%ROWTYPE;
BEGIN
   OPEN goals_cursor_g2;
   FETCH goals_cursor_g2 INTO v_player_most_goals_g2;
   CLOSE goals_cursor_g2;

   SELECT *
```

```
    INTO v_player_info
    FROM Player
    WHERE Player_ID = v_player_most_goals_g2.Player_ID;

    IF v_player_most_goals_g2.Player_ID IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Game 2 - Player with the most goals:');
                DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
        DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
        DBMS_OUTPUT.PUT_LINE('Goals Scored: ' || v_player_most_goals_g2.GoalsThisGame);
        DBMS_OUTPUT.PUT_LINE('');
    END IF;

    OPEN assists_cursor_g2;
    FETCH assists_cursor_g2 INTO v_player_most_assists_g2;
    CLOSE assists_cursor_g2;

    SELECT *
    INTO v_player_info
    FROM Player
    WHERE Player_ID = v_player_most_assists_g2.Player_ID;

    IF v_player_most_assists_g2.Player_ID IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Game 2 - Player with the most assists:');
                DBMS_OUTPUT.PUT_LINE('-------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
        DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
        DBMS_OUTPUT.PUT_LINE('Assists: ' || v_player_most_assists_g2.AssistsThisMatch);
```

```
    DBMS_OUTPUT.PUT_LINE('');
  END IF;

  OPEN yellow_cards_cursor_g2;
  FETCH yellow_cards_cursor_g2 INTO v_player_most_yellow_cards_g2;
  CLOSE yellow_cards_cursor_g2;

  SELECT *
  INTO v_player_info
  FROM Player
  WHERE Player_ID = v_player_most_yellow_cards_g2.Player_ID;

  IF v_player_most_yellow_cards_g2.Player_ID IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Game 2 - Player with the most yellow cards:');
            DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Name: ' || v_player_info.First_Name || ' ' ||
v_player_info.Last_Name);
    DBMS_OUTPUT.PUT_LINE('Age: ' || v_player_info.Age);
    DBMS_OUTPUT.PUT_LINE('Position: ' || v_player_info.Position);
    DBMS_OUTPUT.PUT_LINE('Yellow Cards: ' || v_player_most_yellow_cards_g2.Yellow_Cards);
    DBMS_OUTPUT.PUT_LINE('');
  END IF;
END;
/
```

OUTPUT:

```
Statement processed.
--------------------------------------
Game 2 - Player with the most goals:
--------------------------------------
Name: Lionel Messi
Age: 34
Position: Forward
Goals Scored: 1
--------------------------------------
Game 2 - Player with the most assists:
--------------------------------------
Name: Angel Di Maria
Age: 33
Position: Midfielder
Assists: 1
--------------------------------------
Game 2 - Player with the most yellow cards:
--------------------------------------
Name: Gonzalo Montiel
Age: 26
Position: Defender
Yellow Cards: 1
```

BLOCK 8: 83 lines purpose of this query is primarily to exemplify the exception case and use. In this particular block, the goal is to find the highest scorer of every game. However, since there are multiple players that scored the same amount an exception is the output.

SQL:

```
DECLARE
   v_player_id Player.Player_ID%TYPE;
   v_player_name VARCHAR2(100);
   v_club_name VARCHAR2(50);
   v_position VARCHAR2(20);
   v_total_goals NUMBER;
   v_game_id PlayerStats_PerMatch.Game_ID%TYPE;
   v_stadium_attendance VARCHAR2(8);
   v_goal_counter NUMBER := 0;
   v_referee_id Referees.Referee_ID%TYPE;
   v_referee_name Referees.First_Name%TYPE;
BEGIN
   SELECT
      ps.Player_ID,
      p.First_Name || ' ' || p.Last_Name,
      c.Club_Name,
      p.Position,
      SUM(ps.GoalsThisGame) AS total_goals,
      ps.Game_ID
   INTO
      v_player_id,
      v_player_name,
      v_club_name,
      v_position,
      v_total_goals,
      v_game_id
   FROM
      PlayerStats_PerMatch ps
   JOIN
```

```
    Player p ON ps.Player_ID = p.Player_ID
  JOIN
    Club c ON p.Club_ID = c.Club_ID
  WHERE
    p.Position != 'Goalkeeper'
  GROUP BY
    ps.Player_ID, p.First_Name, p.Last_Name, c.Club_Name, p.Position, ps.Game_ID
  HAVING
    SUM(ps.GoalsThisGame) = (SELECT MAX(GoalsThisGame) FROM PlayerStats_PerMatch
WHERE Game_ID = ps.Game_ID)
  ORDER BY
    total_goals DESC;

  SELECT
    g.Referee_ID,
    r.First_Name
  INTO
    v_referee_id,
    v_referee_name
  FROM
    Game g
  JOIN
    Referees r ON g.Referee_ID = r.Referee_ID
  WHERE
    g.Game_ID = v_game_id;

  SELECT
    STADIUM_ATTENDANCE
  INTO
    v_stadium_attendance
  FROM
    Attendance
  WHERE
    GAME_ID = v_game_id;
```

```
    IF v_player_id IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE('Player Name: ' || v_player_name);
        DBMS_OUTPUT.PUT_LINE('Club: ' || v_club_name);
        DBMS_OUTPUT.PUT_LINE('Position: ' || v_position);
        DBMS_OUTPUT.PUT_LINE('Total Goals: ' || v_total_goals);
        DBMS_OUTPUT.PUT_LINE('Game ID: ' || v_game_id);
        DBMS_OUTPUT.PUT_LINE('Attendance: ' || v_stadium_attendance);
        DBMS_OUTPUT.PUT_LINE('Referee ID: ' || v_referee_id);
        DBMS_OUTPUT.PUT_LINE('Referee Name: ' || v_referee_name);
    ELSE
        DBMS_OUTPUT.PUT_LINE('No data found.');
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No data found.');
            WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('More than one player found with the highest number of goals.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred: ');
END;
/
```

OUTPUT:

```
Statement processed.
More than one player found with the highest number of goals.
```

BLOCK 9: 116 lines the aim of this query is to find and compare the possession times across different games. It provides the game in which Argentina had the highest possession time vs their opponent as well as the lowest amount of possession time. The output includes the final score of the game as well as the opponent, and team formation to draw possible correlation towards formation type and possession time.

SQL:

```
DECLARE
    v_max_possession_percentage NUMBER;
    v_min_possession_percentage NUMBER;
    v_max_opponent_possession_percentage NUMBER;
    v_min_opponent_possession_percentage NUMBER;
    v_game_number_1 VARCHAR2(100);
    v_game_number_2 VARCHAR2(100);
    v_game_number_3 VARCHAR2(100);
    v_game_number_4 VARCHAR2(100);
    v_opponent_name_1 VARCHAR2(100);
    v_opponent_name_2 VARCHAR2(100);
    v_opponent_name_3 VARCHAR2(100);
    v_opponent_name_4 VARCHAR2(100);
    v_results_1 VARCHAR(100);
    v_results_2 VARCHAR(100);
    v_results_3 VARCHAR(100);
    v_results_4 VARCHAR(100);
    v_stadium_name_1 VARCHAR2(100);
    v_stadium_name_2 VARCHAR2(100);
    v_stadium_name_3 VARCHAR2(100);
    v_stadium_name_4 VARCHAR2(100);
    v_attendance_1 NUMBER;
    v_attendance_2 NUMBER;
    v_attendance_3 NUMBER;
    v_attendance_4 NUMBER;
    v_capacity_1 NUMBER;
```

```
    v_capacity_2 NUMBER;
    v_capacity_3 NUMBER;
    v_capacity_4 NUMBER;
BEGIN
    SELECT MAX(possesion_percentage) -- finds the highest possession percentage
    INTO v_max_possession_percentage
    FROM Possession_Time;

    FOR game_rec IN (SELECT pt.game_id, g.opponent, g.results, a.stadium_name,
a.stadium_attendance, a.stadium_capacity, pt.possesion_percentage, pt.formation
                FROM Possession_Time pt
                JOIN Attendance a ON pt.game_id = a.game_id
                JOIN Game g ON pt.game_id = g.game_id
                WHERE possesion_percentage = v_max_possession_percentage) LOOP
        v_game_number_1 := game_rec.game_id;
        v_opponent_name_1 := game_rec.opponent;
        v_results_1 := game_rec.results;
        v_stadium_name_1 := game_rec.stadium_name;
        v_attendance_1 := game_rec.stadium_attendance;
        v_capacity_1 := game_rec.stadium_capacity;

        SELECT MIN(opponent_possesion_percentage)
        INTO v_min_opponent_possession_percentage
        FROM Possession_Time;

        FOR game_rec_4 IN (SELECT pt.game_id, g.opponent, g.results, a.stadium_name,
a.stadium_attendance, a.stadium_capacity, pt.opponent_possesion_percentage,
pt.opponent_formation
                        FROM Possession_Time pt
                        JOIN Attendance a ON pt.game_id = a.game_id
                        JOIN Game g ON pt.game_id = g.game_id
```

```
                    WHERE opponent_possesion_percentage =
v_min_opponent_possession_percentage) LOOP
        v_game_number_4 := game_rec_4.game_id;
        v_opponent_name_4 := game_rec_4.opponent;
        v_results_4 := game_rec_4.results;
        v_stadium_name_4 := game_rec_4.stadium_name;
        v_attendance_4 := game_rec_4.stadium_attendance;
        v_capacity_4 := game_rec_4.stadium_capacity;

        DBMS_OUTPUT.PUT_LINE('-------------------------------------------------');
        dbms_output.put_line('Game Number with Highest Possession Percentage:          |
Game Number with Lowest Opponent Possession Percentage:');
        dbms_output.put_line('Game Number: ' || v_game_number_1 || '
| Game Number: ' || v_game_number_4);
        dbms_output.put_line('Opponent: ' || v_opponent_name_1 || ' - Final Score: ' || v_results_1
|| '              | Opponent: ' || v_opponent_name_4 || ' - Final Score: ' || v_results_4);
        dbms_output.put_line('Stadium Name: ' || v_stadium_name_1 || '                          |
Stadium Name: ' || v_stadium_name_4);
        dbms_output.put_line('Attendance: ' || v_attendance_1 || '/' || v_capacity_1 || '
| Attendance: ' || v_attendance_4 || '/' || v_capacity_4);
        dbms_output.put_line('Possession Percentage: ' || game_rec.possesion_percentage || '%' || '
| Opponent Possession %: ' || game_rec_4.opponent_possesion_percentage || '%');
        dbms_output.put_line('Formation: ' || game_rec.formation || '                           |
Opponent Formation: ' || game_rec_4.opponent_formation);
    END LOOP;

    SELECT MAX(opponent_possesion_percentage)
    INTO v_max_opponent_possession_percentage
    FROM Possession_Time;
```

```
    FOR game_rec_2 IN (SELECT pt.game_id, g.opponent, g.results, a.stadium_name,
a.stadium_attendance, a.stadium_capacity, pt.opponent_possesion_percentage,
pt.opponent_formation
                    FROM Possession_Time pt
                    JOIN Attendance a ON pt.game_id = a.game_id
                    JOIN Game g ON pt.game_id = g.game_id
                    WHERE opponent_possesion_percentage =
v_max_opponent_possession_percentage) LOOP
        v_game_number_2 := game_rec_2.game_id;
        v_opponent_name_2 := game_rec_2.opponent;
        v_results_2 := game_rec_2.results;
        v_stadium_name_2 := game_rec_2.stadium_name;
        v_attendance_2 := game_rec_2.stadium_attendance;
        v_capacity_2 := game_rec_2.stadium_capacity;

        SELECT MIN(possesion_percentage)
        INTO v_min_possession_percentage
        FROM Possession_Time;

        FOR game_rec_3 IN (SELECT pt.game_id, g.opponent, g.results, a.stadium_name,
a.stadium_attendance, a.stadium_capacity, pt.possesion_percentage, pt.formation
                    FROM Possession_Time pt
                    JOIN Attendance a ON pt.game_id = a.game_id
                    JOIN Game g ON pt.game_id = g.game_id
                    WHERE possesion_percentage = v_min_possession_percentage) LOOP
            v_game_number_3 := game_rec_3.game_id;
            v_opponent_name_3 := game_rec_3.opponent;
            v_results_3 := game_rec_3.results;
            v_stadium_name_3 := game_rec_3.stadium_name;
            v_attendance_3 := game_rec_3.stadium_attendance;
            v_capacity_3 := game_rec_3.stadium_capacity;
```

```
            DBMS_OUTPUT.PUT_LINE('-------------------------------------------------');
            dbms_output.put_line('Game Number with Highest Opponent Possession Percentage:
| Game Number with Lowest Possession Percentage:');
            dbms_output.put_line('Game Number: ' || v_game_number_2 || '
| Game Number: ' || v_game_number_3);
            dbms_output.put_line('Opponent: ' || v_opponent_name_2 || ' - Final Score: ' ||
v_results_2 || '                | Opponent: ' || v_opponent_name_3 || ' - Final Score: ' || v_results_3);
            dbms_output.put_line('Stadium Name: ' || v_stadium_name_2 || '                    |
Stadium Name: ' || v_stadium_name_3);
            dbms_output.put_line('Attendance: ' || v_attendance_2 || '/' || v_capacity_2 || '
| Attendance: ' || v_attendance_3 || '/' || v_capacity_3);
            dbms_output.put_line('Opponent Possession Percentage: ' ||
game_rec_2.opponent_possesion_percentage || '%' || '                    | Possession Percentage: ' ||
game_rec_3.possesion_percentage || '%');
            dbms_output.put_line('Opponent Formation: ' || game_rec_2.opponent_formation || '
| Formation: ' || game_rec_3.formation);
        END LOOP;
    END LOOP;
  END LOOP;
END;
```

OUTPUT:

```
Statement processed.
-------------------------------------------------
Game Number with Highest Possession Percentage:          | Game Number with Lowest Opponent Possession Percentage:
Game Number: G3                                          | Game Number: G3
Opponent: Poland - Final Score: 2-0(Win)                 | Opponent: Poland - Final Score: 2-0(Win)
Stadium Name: Stadium 974                                | Stadium Name: Stadium 974
Attendance: 44089/44089                                  | Attendance: 44089/44089
Possession Percentage: 73.3%                             | Opponent Possession %: 26.7%
Formation: 4-4-2                                         | Opponent Formation: 4-3-3
-------------------------------------------------
Game Number with Highest Opponent Possession Percentage: | Game Number with Lowest Possession Percentage:
Game Number: G6                                          | Game Number: G6
Opponent: Croatia - Final Score: 3-0(Win)                | Opponent: Croatia - Final Score: 3-0(Win)
Stadium Name: Lusail Iconic Stadium                      | Stadium Name: Lusail Iconic Stadium
Attendance: 88966/88966                                  | Attendance: 88966/88966
Opponent Possession Percentage: 60.8%                    | Possession Percentage: 39.2%
Opponent Formation: 4-3-3                                | Formation: 4-4-2
```

BLOCK 10: 67 lines create a cursor that lists the location from the game_location table of game 5. create a cursor to list the player(s) that got yellow cards during this game. create a cursor that applies a -$100,000 loss to their total salary for receiving a yellow card. print the players names, salaries and the new salary after the 100,000 decrease

SQL:

```
DECLARE
   CURSOR game_info_cursor IS -- cursor for game 5 general information
   SELECT g.Results, gl.Location, gl.Opponent
   FROM Game g
   JOIN Game_Location gl ON g.Game_ID = gl.Game_ID
   WHERE g.Game_ID = 'G5';

   v_results VARCHAR2(20);
   v_game_location VARCHAR2(22);
   v_opponent VARCHAR2(15);

   CURSOR yellow_card_players_cursor IS -- cursor for players who got yellow cards
   SELECT p.Player_ID, p.First_Name, p.Last_Name, p.Position, p.Age, p.Salary_USD,
pspm.Yellow_Cards
   FROM Player p
   JOIN PlayerStats_PerMatch pspm ON p.Player_ID = pspm.Player_ID
   WHERE pspm.Game_ID = 'G5' AND pspm.Yellow_Cards > 0;

   v_player_id VARCHAR2(3);
   v_first_name VARCHAR2(20);
   v_last_name VARCHAR2(20);
   v_position VARCHAR2(15);
   v_age INT;
```

```
    v_salary NUMBER;
    v_fine NUMBER := 100000; -- set fine amount
    v_total_yellow_cards INT := 0; -- set total yellow card counter to 0
    v_yellow_cards INT;

    CURSOR game_attendance_cursor IS -- cursor to find attendance # of game 5
    SELECT a.Stadium_Attendance
    FROM Attendance a
    WHERE a.Game_ID = 'G5';

    v_stadium_attendance NUMBER;
BEGIN
    OPEN game_info_cursor;
    FETCH game_info_cursor INTO v_results, v_game_location, v_opponent;
    CLOSE game_info_cursor;

    OPEN game_attendance_cursor;
    FETCH game_attendance_cursor INTO v_stadium_attendance;
    CLOSE game_attendance_cursor;

    DBMS_OUTPUT.PUT_LINE('Game Results: ' || v_results || ' | Attendance: ' ||
v_stadium_attendance);
    DBMS_OUTPUT.PUT_LINE('Game 5 Location: ' || v_game_location || ' | Opponent: ' ||
v_opponent);
    DBMS_OUTPUT.PUT_LINE('-----------------------------------');

    OPEN yellow_card_players_cursor;
    LOOP
        FETCH yellow_card_players_cursor INTO v_player_id, v_first_name, v_last_name,
v_position, v_age, v_salary, v_yellow_cards;
        EXIT WHEN yellow_card_players_cursor%NOTFOUND;
```

```
        v_total_yellow_cards := v_total_yellow_cards + v_yellow_cards; -- calc for total yellow cards

        v_salary := v_salary - v_fine; -- salary after fine

        DBMS_OUTPUT.PUT_LINE(
            RPAD('Player: ' || v_first_name || ' ' || v_last_name, 30) ||
            RPAD('Position: ' || v_position, 15) ||
            RPAD('Age: ' || v_age, 10) ||
            'Fine Amount: $' || TO_CHAR(v_fine) ||
            ' | Salary after Fine: $' || TO_CHAR(v_salary)
        );
    END LOOP;
    CLOSE yellow_card_players_cursor;

    DBMS_OUTPUT.PUT_LINE('Total Yellow Cards: ' || v_total_yellow_cards);
END;
```

OUTPUT:

```
Statement processed.
Game Results: 2(3) - 2(4)(Win) | Attendance: 88235
Game 5 Location: Lusail Iconic Stadium | Opponent: Netherlands
-----------------------------------
Player: Lionel Messi          Position: ForwaAge: 34   Fine Amount: $100000 | Salary after Fine: $20300000
Player: Nicolas Otamendi      Position: MidfiAge: 33   Fine Amount: $100000 | Salary after Fine: $11870000
Player: Marcos Acuna          Position: MidfiAge: 29   Fine Amount: $100000 | Salary after Fine: $2930000
Player: Gonzalo Montiel       Position: DefenAge: 26   Fine Amount: $100000 | Salary after Fine: $1390617
Player: Cristian Romero       Position: DefenAge: 25   Fine Amount: $100000 | Salary after Fine: $9208871
Player: Lisandro Martinez     Position: DefenAge: 25   Fine Amount: $100000 | Salary after Fine: $6670088
Player: Leandro Paredes       Position: MidfiAge: 29   Fine Amount: $100000 | Salary after Fine: $4239800
Player: German Pezzella       Position: DefenAge: 32   Fine Amount: $100000 | Salary after Fine: $3579815
Total Yellow Cards: 8
```

BLOCK 11: 88 lines: creates different cursors to find the earliest game played throughout all seven games for the Argentinian national team. As well it creates cursors to find the latest game to start during their run. The results print out the opponent as well as different stadium, attendance and coverage stats.

SQL:

```
DECLARE

   CURSOR attendance_egame_cursor IS -- creates cursor to find  the earliest game, its attendance information and game time information
   SELECT at.Stadium_Attendance, at.Television_Coverage, at.Stadium_Name, at.Stadium_Capacity
   FROM Attendance at
   JOIN Game_Time gt ON at.Game_ID = gt.Game_ID
   JOIN Game g ON gt.Game_ID = g.Game_ID
   WHERE gt.Eastern_Standard_Time_EST = (SELECT MIN(Eastern_Standard_Time_EST) FROM Game_Time); -- minimum time in est

   CURSOR attendance_lgame_cursor IS  -- creates cursor to find  the latest game, its attendance information and game time information
   SELECT at.Stadium_Attendance, at.Television_Coverage, at.Stadium_Name, at.Stadium_Capacity
   FROM Attendance at
   JOIN Game_Time gt ON at.Game_ID = gt.Game_ID
   JOIN Game g ON gt.Game_ID = g.Game_ID
   WHERE gt.Eastern_Standard_Time_EST = (SELECT MAX(Eastern_Standard_Time_EST) FROM Game_Time);-- maximum time in est

   CURSOR earliest_game_cursor IS
   SELECT gt.Game_ID, gt.Eastern_Standard_Time_EST, g.Opponent, g.Results
   FROM Game_Time gt
   JOIN Game g ON gt.Game_ID = g.Game_ID
   WHERE gt.Eastern_Standard_Time_EST = (SELECT MIN(Eastern_Standard_Time_EST) FROM Game_Time);

   CURSOR latest_game_cursor IS
   SELECT gt.Game_ID, gt.Eastern_Standard_Time_EST, g.Opponent, g.Results
```

```
    FROM Game_Time gt
    JOIN Game g ON gt.Game_ID = g.Game_ID
    WHERE gt.Eastern_Standard_Time_EST = (SELECT MAX(Eastern_Standard_Time_EST) FROM
Game_Time);

    earliest_game_id VARCHAR2(2);
    earliest_game_time DATE;
    earliest_opponent_name VARCHAR2(100);
    earliest_game_results VARCHAR(50);
    latest_game_id VARCHAR2(2);
    latest_game_time DATE;
    latest_opponent_name VARCHAR2(100);
    latest_game_results VARCHAR(50);
    max_attendance NUMBER;
    television_coverage VARCHAR2(100);
    stadium_name VARCHAR2(100);
    stadium_capacity NUMBER;
BEGIN

    OPEN earliest_game_cursor;

    FETCH earliest_game_cursor INTO earliest_game_id, earliest_game_time, earliest_opponent_name,
earliest_game_results;

    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Earliest Game: ' || earliest_game_id || '/G7, Time: ' ||
TO_CHAR(earliest_game_time, 'HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE('Opponent: ' || earliest_opponent_name);
    DBMS_OUTPUT.PUT_LINE('Results: ' || earliest_game_results);

    CLOSE earliest_game_cursor;

    OPEN attendance_egame_cursor;
```

```
    FETCH attendance_egame_cursor INTO max_attendance, television_coverage, stadium_name,
stadium_capacity;

    DBMS_OUTPUT.PUT_LINE('Stadium Name: ' || stadium_name);
    DBMS_OUTPUT.PUT_LINE('Stadium Capacity: ' || stadium_capacity);
    DBMS_OUTPUT.PUT_LINE('Attendance of Earliest Game: ' || max_attendance);
    DBMS_OUTPUT.PUT_LINE('Television Coverage of Earliest Game: ' || television_coverage);

    CLOSE attendance_egame_cursor;

    OPEN latest_game_cursor;

    FETCH latest_game_cursor INTO latest_game_id, latest_game_time, latest_opponent_name,
latest_game_results;

    DBMS_OUTPUT.PUT_LINE('-------------------------------------');
    DBMS_OUTPUT.PUT_LINE('Latest Game: ' || latest_game_id || '/G7, Time: ' ||
TO_CHAR(latest_game_time, 'HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE('Opponent: ' || latest_opponent_name);
    DBMS_OUTPUT.PUT_LINE('Results: ' || latest_game_results);

    CLOSE latest_game_cursor;

    OPEN attendance_lgame_cursor;

    FETCH attendance_lgame_cursor INTO max_attendance, television_coverage, stadium_name,
stadium_capacity;

    DBMS_OUTPUT.PUT_LINE('Stadium Name: ' || stadium_name );
    DBMS_OUTPUT.PUT_LINE('Stadium Capacity: ' || stadium_capacity);
    DBMS_OUTPUT.PUT_LINE('Attendance of Latest Game: ' || max_attendance);
    DBMS_OUTPUT.PUT_LINE('Television Coverage of Latest Game: ' || television_coverage);

    CLOSE attendance_lgame_cursor;
```

```
END;
/
```

OUTPUT:

```
Statement processed.
------------------------------------
Earliest Game: G1/G7, Time: 05:00:00
Opponent: Saudi Arabia
Results: 1-2(Loss)
Stadium Name: Lusail Iconic Stadium
Stadium Capacity: 88966
Attendance of Earliest Game: 88012
Television Coverage of Earliest Game: FS1
------------------------------------
Latest Game: G2/G7, Time: 14:00:00
Opponent: Mexico
Results: 2-0(Win)
Stadium Name: Lusail Iconic Stadium
Stadium Capacity: 88966
Attendance of Latest Game: 88966
Television Coverage of Latest Game: FS1
```

BLOCK 12: 99 LINES query aims to differentiate the top 5 players with the longest and shortest tenure on the Argentina national team. After printing out the general player information it calculates and outputs the total number of days, weeks, months and years these players have been on the team.

SQL:

```
DECLARE
   CURSOR top_players_cursor IS -- cursor to find top tenure
   SELECT ph.Player_ID, p.First_Name, p.Last_Name, p.Position, ph.Start_Date
   FROM Player_History ph
   JOIN Player p ON ph.Player_ID = p.Player_ID
   WHERE ph.End_Date IS NULL
   ORDER BY ph.Start_Date ASC;

   v_counter NUMBER := 1;
   v_player_id VARCHAR2(3);
   v_first_name VARCHAR2(20);
   v_last_name VARCHAR2(20);
   v_position VARCHAR2(15);
   v_start_date DATE;
   v_total_days_on_team NUMBER := 0;
   v_total_weeks_on_team NUMBER := 0;
   v_total_months_on_team NUMBER := 0;
   v_total_years_on_team NUMBER := 0;
BEGIN
   OPEN top_players_cursor;

   DBMS_OUTPUT.PUT_LINE('Top 5 Players Who Have Been on the Team the Longest:');
   DBMS_OUTPUT.PUT_LINE('-------------------------------------------------');
   LOOP
      FETCH top_players_cursor INTO v_player_id, v_first_name, v_last_name, v_position, v_start_date;
      EXIT WHEN top_players_cursor%NOTFOUND OR v_counter > 5;
      DBMS_OUTPUT.PUT_LINE('Player ID: ' || v_player_id || ' | Name: ' || v_first_name || ' ' ||
v_last_name || ' | Position: ' || v_position || ' | Start Date: ' || TO_CHAR(v_start_date, 'YYYY-MM-DD'));
```

```
        v_total_days_on_team := v_total_days_on_team + (SYSDATE - v_start_date); -- calcs diff between system current date and player start date

        v_counter := v_counter + 1;
    END LOOP;

    CLOSE top_players_cursor;

    v_total_days_on_team := ROUND(v_total_days_on_team); -- total number of days rounded

    v_total_weeks_on_team := ROUND(v_total_days_on_team / 7); -- divided by days in a week

    v_total_months_on_team := ROUND(v_total_weeks_on_team / 4.34524); -- divided by average weeks in a month

    v_total_years_on_team := ROUND(v_total_months_on_team / 12); -- divided by total months in yr

    DBMS_OUTPUT.PUT_LINE('Total # of Days: ' || v_total_days_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Weeks: ' || v_total_weeks_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Months: ' || v_total_months_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Years: ' || v_total_years_on_team);
END;
/

DECLARE
    CURSOR bottom_players_cursor IS -- cursor for players with lowest tenure
    SELECT ph.Player_ID, p.First_Name, p.Last_Name, p.Position, ph.Start_Date
    FROM Player_History ph
    JOIN Player p ON ph.Player_ID = p.Player_ID
    WHERE ph.End_Date IS NULL
    ORDER BY ph.Start_Date DESC;

    v_counter NUMBER := 1;
    v_player_id VARCHAR2(3);
```

```
    v_first_name VARCHAR2(20);
    v_last_name VARCHAR2(20);
    v_position VARCHAR2(15);
    v_start_date DATE;
    v_total_days_on_team NUMBER := 0;
    v_total_weeks_on_team NUMBER := 0;
    v_total_months_on_team NUMBER := 0;
    v_total_years_on_team NUMBER := 0;
BEGIN
    OPEN bottom_players_cursor;

    DBMS_OUTPUT.PUT_LINE('Top 5 Players Who Have Been on the Team the Shortest:');
    DBMS_OUTPUT.PUT_LINE('--------------------------------------------------');
    LOOP
        FETCH bottom_players_cursor INTO v_player_id, v_first_name, v_last_name, v_position,
v_start_date;
        EXIT WHEN bottom_players_cursor%NOTFOUND OR v_counter > 5;
        DBMS_OUTPUT.PUT_LINE('Player ID: ' || v_player_id || ' | Name: ' || v_first_name || ' ' ||
v_last_name || ' | Position: ' || v_position || ' | Start Date: ' || TO_CHAR(v_start_date, 'YYYY-MM-DD'));

        v_total_days_on_team := v_total_days_on_team + (SYSDATE - v_start_date);

        v_counter := v_counter + 1;
    END LOOP;

    CLOSE bottom_players_cursor;

    v_total_days_on_team := ROUND(v_total_days_on_team);

    v_total_weeks_on_team := ROUND(v_total_days_on_team / 7);

    v_total_months_on_team := ROUND(v_total_weeks_on_team / 4.34524);

    v_total_years_on_team := ROUND(v_total_months_on_team / 12);-- divided by total months in yr
```

```
    DBMS_OUTPUT.PUT_LINE('Total # of Days: ' || v_total_days_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Weeks: ' || v_total_weeks_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Months: ' || v_total_months_on_team);
    DBMS_OUTPUT.PUT_LINE('Total # of Years: ' || v_total_years_on_team);
END;
/
```

OUTPUT:

```
Statement processed.
Top 5 Players Who Have Been on the Team the Longest:
-------------------------------------------------
Player ID: A10 | Name: Lionel Messi | Position: Forward | Start Date: 2005-08-17
Player ID: A11 | Name: Sergio Aguero | Position: Forward | Start Date: 2006-09-03
Player ID: A13 | Name: Angel Di Maria | Position: Midfielder | Start Date: 2008-09-06
Player ID: A12 | Name: Nicolas Otamendi | Position: Midfielder | Start Date: 2009-05-20
Player ID: A24 | Name: German Pezzella | Position: Defender | Start Date: 2011-12-07
Total # of Days: 28998
Total # of Weeks: 4143
Total # of Months: 953
Total # of Years: 79

Statement processed.
Top 5 Players Who Have Been on the Team the Shortest:
-------------------------------------------------
Player ID: A16 | Name: Enzo Fernandez | Position: Midfielder | Start Date: 2022-09-24
Player ID: A21 | Name: Cristian Romero | Position: Defender | Start Date: 2021-03-06
Player ID: A17 | Name: Julian Alvarez | Position: Forward | Start Date: 2021-03-06
Player ID: A14 | Name: Nahuel Molina | Position: Defender | Start Date: 2021-03-06
Player ID: A15 | Name: Emiliano Martinez | Position: Goalkeeper | Start Date: 2021-03-06
Total # of Days: 5204
Total # of Weeks: 743
Total # of Months: 171
Total # of Years: 14
```

BLOCK 13: 75 Lines creates a package that finds the total length of the 2022 World Cup. This package is full of functions that break down the length into days, weeks, months and amount of the year. It then prints the opponents of all those games and their start times.

SQL:

```
CREATE OR REPLACE PACKAGE WorldCup_Length AS -- creates a package that includes functions that return total number of days/weeks, etc.
   FUNCTION get_total_days RETURN NUMBER;
   FUNCTION get_total_weeks RETURN NUMBER;
   FUNCTION get_total_months RETURN NUMBER;
   FUNCTION get_total_years RETURN NUMBER;
END WorldCup_Length;
/

CREATE OR REPLACE PACKAGE BODY WorldCup_Length AS
   FUNCTION get_total_days RETURN NUMBER IS -- function to find total days
      total_days NUMBER;
   BEGIN
      SELECT (MAX(Game_Date) - MIN(Game_Date)) INTO total_days FROM Game; -- first minus last game date
      RETURN total_days; -- return total number of days in world cup
   END get_total_days; -- end function

   FUNCTION get_total_weeks RETURN NUMBER IS -- function to find total days
      total_days NUMBER;
      total_weeks NUMBER;
   BEGIN
      total_days := get_total_days;
      total_weeks := ROUND(total_days / 7, 2); -- weeks broken up into 7 days
      RETURN total_weeks; -- return total number of weeks in world cup
   END get_total_weeks; -- ends this part of the function

   FUNCTION get_total_months RETURN NUMBER IS
      total_days NUMBER;
      total_months NUMBER;
```

```
    BEGIN
        total_days := get_total_days;
        total_months := ROUND(total_days / 30.4375, 2);  -- divided by the average number of days in a month
        RETURN total_months;-- return total number of months in world cup
    END get_total_months;-- ends this part of the function

    FUNCTION get_total_years RETURN NUMBER IS
        total_days NUMBER;
        total_years NUMBER;
    BEGIN
        total_days := get_total_days;
        total_years := ROUND(total_days / 365.25, 2); -- divided by the number of days in a year, 2 decimal
        RETURN total_years; -- return total number of months in world cup
    END get_total_years;-- ends the total year part of the function
END WorldCup_Length; -- concludes info included in this package
/

DECLARE
    total_days NUMBER; -- declare variables to print
    total_weeks NUMBER;
    total_months NUMBER;
    total_years NUMBER;
    game_time TIMESTAMP;
    opponent_game VARCHAR(20);
    game_result VARCHAR(20);
BEGIN
    total_days := WorldCup_Length.get_total_days; -- call package to get total days
    total_weeks := WorldCup_Length.get_total_weeks; -- call package to get total weeks
    total_months := WorldCup_Length.get_total_months; -- call package to get total months
    total_years := WorldCup_Length.get_total_years; -- call package to get total years

    DBMS_OUTPUT.PUT_LINE('Length of 2022 World Cup in Days: ' || total_days); -- print total days
    DBMS_OUTPUT.PUT_LINE('Length of 2022 World Cup in Weeks: ' || total_weeks);-- print total weeks
    DBMS_OUTPUT.PUT_LINE('Length of 2022 World Cup in Months: ' || total_months);-- print total months
    DBMS_OUTPUT.PUT_LINE('Length of 2022 World Cup in Years: ' || total_years);-- print total years
```

```
    FOR game_rec IN (SELECT G.Game_ID, G.Game_Date, GT.Eastern_Standard_Time_EST, G.Opponent,
G.Results
                     FROM Game G
                     JOIN Game_Time GT ON G.Game_ID = GT.Game_ID)
    LOOP
        game_time := game_rec.Game_Date + (game_rec.Eastern_Standard_Time_EST -
TRUNC(game_rec.Eastern_Standard_Time_EST));
        opponent_game := game_rec.Opponent;
        game_result := game_rec.Results;
        DBMS_OUTPUT.PUT_LINE('Game ID: ' || game_rec.Game_ID || ' | Game Date: ' ||
TO_CHAR(game_rec.Game_Date, 'YYYY-MM-DD') || ' | Start Time (EST): ' || TO_CHAR(game_time, '
HH24:MI:SS') || ' | Opponent: ' || opponent_game || ' | Result: ' || game_result);
    END LOOP;
END;
/
```

OUTPUT:

```
Package created.

Package Body created.

Statement processed.
Length of 2022 World Cup in Days: 26
Length of 2022 World Cup in Weeks: 3.71
Length of 2022 World Cup in Months: .85
Length of 2022 World Cup in Years: .07
Game ID: G1 | Game Date: 2022-11-22 | Start Time (EST):  05:00:00 | Opponent: Saudi Arabia | Result: 1-2(Loss)
Game ID: G2 | Game Date: 2022-11-26 | Start Time (EST):  14:00:00 | Opponent: Mexico | Result: 2-0(Win)
Game ID: G3 | Game Date: 2022-11-30 | Start Time (EST):  14:00:00 | Opponent: Poland | Result: 2-0(Win)
Game ID: G4 | Game Date: 2022-12-03 | Start Time (EST):  08:00:00 | Opponent: Australia | Result: 2-1(Win)
Game ID: G5 | Game Date: 2022-12-09 | Start Time (EST):  14:00:00 | Opponent: Netherlands | Result: 2(3) - 2(4)(Win)
Game ID: G6 | Game Date: 2022-12-13 | Start Time (EST):  14:00:00 | Opponent: Croatia | Result: 3-0(Win)
Game ID: G7 | Game Date: 2022-12-18 | Start Time (EST):  10:00:00 | Opponent: France | Result: 3(4) - 3(2)(Win)
```

BLOCK 14: USING this query is similar to one previously listed, except that it creates a fictitious player while also using the UPDATE keyword. Other than that the query aims to distribute fines to every player that received a yellow card in game 7. The output shows the game information as wella s the player salaries after the fine.

SQL:

```
DECLARE
   CURSOR game_info_cursor IS
   SELECT g.Results, gl.Location, gl.Opponent
   FROM Game g
   JOIN Game_Location gl ON g.Game_ID = gl.Game_ID
   WHERE g.Game_ID = 'G7';

   v_results VARCHAR2(20);
   v_game_location VARCHAR2(22);
   v_opponent VARCHAR2(15);

   CURSOR yellow_card_players_cursor IS
   SELECT p.Player_ID, p.First_Name, p.Last_Name, p.Position, p.Age, p.Salary_USD,
pspm.Yellow_Cards
   FROM Player p
   JOIN PlayerStats_PerMatch pspm ON p.Player_ID = pspm.Player_ID
   WHERE pspm.Game_ID = 'G7' AND pspm.Yellow_Cards > 0;

   v_player_id VARCHAR2(3);
   v_first_name VARCHAR2(20);
   v_last_name VARCHAR2(20);
   v_position VARCHAR2(15);
   v_age INT;
   v_salary NUMBER;
   v_fine NUMBER := 275000;
   v_total_yellow_cards INT := 0;
   v_yellow_cards INT;
```

```
    CURSOR game_attendance_cursor IS
    SELECT a.Stadium_Attendance
    FROM Attendance a
    WHERE a.Game_ID = 'G7';

    v_stadium_attendance NUMBER;
BEGIN
    OPEN game_info_cursor;
    FETCH game_info_cursor INTO v_results, v_game_location, v_opponent;
    CLOSE game_info_cursor;

    OPEN game_attendance_cursor;
    FETCH game_attendance_cursor INTO v_stadium_attendance;
    CLOSE game_attendance_cursor;

    DBMS_OUTPUT.PUT_LINE('Game Results: ' || v_results || ' | Attendance: ' || v_stadium_attendance);
    DBMS_OUTPUT.PUT_LINE('Game 7 Location: ' || v_game_location || ' | Opponent: ' || v_opponent);
    DBMS_OUTPUT.PUT_LINE('-------------------------------------');

    OPEN yellow_card_players_cursor;
    LOOP
        FETCH yellow_card_players_cursor INTO v_player_id, v_first_name, v_last_name, v_position,
v_age, v_salary, v_yellow_cards;
        EXIT WHEN yellow_card_players_cursor%NOTFOUND;

        v_total_yellow_cards := v_total_yellow_cards + v_yellow_cards;

        v_salary := v_salary - v_fine;

        DBMS_OUTPUT.PUT_LINE(
            RPAD('Player: ' || v_first_name || ' ' || v_last_name, 30) ||
            RPAD('Position: ' || v_position, 15) ||
            RPAD('Age: ' || v_age, 10) ||
            'Fine Amount: $' || TO_CHAR(v_fine) ||
```

```
            ' | Salary after Fine: $' || TO_CHAR(v_salary)
        );

        UPDATE Player -- update statement to add player
        SET Salary_USD = v_salary
        WHERE Player_ID = v_player_id;
    END LOOP;
    CLOSE yellow_card_players_cursor;

    DBMS_OUTPUT.PUT_LINE('Total Yellow Cards: ' || v_total_yellow_cards);

    FOR player_record IN (SELECT * FROM Player) LOOP

        DBMS_OUTPUT.PUT_LINE('-------------------------------------------------');
        DBMS_OUTPUT.PUT_LINE('Player: ' || player_record.First_Name || ' ' || player_record.Last_Name
|| ' | Updated Salary: $' || TO_CHAR(player_record.Salary_USD));
    END LOOP;
END;
/
```

OUTPUT:

```
Statement processed.
Game Results: 3(4) - 3(2)(Win) | Attendance: 88966
Game 7 Location: Lusail Iconic Stadium | Opponent: France
-----------------------------------
Player: Emiliano Martinez     Position: GoalkAge: 29   Fine Amount: $275000 | Salary after Fine: $7299105
Player: Enzo Fernandez        Position: MidfiAge: 26   Fine Amount: $275000 | Salary after Fine: $15959176
Player: Marcos Acuna          Position: MidfiAge: 29   Fine Amount: $275000 | Salary after Fine: $1905000
Player: Gonzalo Montiel       Position: DefenAge: 26   Fine Amount: $275000 | Salary after Fine: $365617
Player: Leandro Paredes       Position: MidfiAge: 29   Fine Amount: $275000 | Salary after Fine: $3214800
Total Yellow Cards: 5
-------------------------------------------------
Player: Lionel Messi | Updated Salary: $20100000
-------------------------------------------------
Player: Sergio Aguero | Updated Salary: $11970000
-------------------------------------------------
Player: Nicolas Otamendi | Updated Salary: $11670000
-------------------------------------------------
Player: Angel Di Maria | Updated Salary: $7690000
-------------------------------------------------
Player: Nahuel Molina | Updated Salary: $5207760
-------------------------------------------------
Player: Emiliano Martinez | Updated Salary: $7299105
-------------------------------------------------
Player: Enzo Fernandez | Updated Salary: $15959176
-------------------------------------------------
Player: Julian Alvarez | Updated Salary: $5641740
-------------------------------------------------
Player: Marcos Acuna | Updated Salary: $1905000
-------------------------------------------------
Player: Alexis Mac Allister | Updated Salary: $8462610
-------------------------------------------------
Player: Gonzalo Montiel | Updated Salary: $365617
-------------------------------------------------
Player: Cristian Romero | Updated Salary: $9008871
-------------------------------------------------
-------------------------------------------------
Player: Lisandro Martinez | Updated Salary: $6470088
-------------------------------------------------
Player: Leandro Paredes | Updated Salary: $3214800
-------------------------------------------------
Player: German Pezzella | Updated Salary: $3379815
-------------------------------------------------
Player: New Defender | Updated Salary: $8000000
```