Presentation

Advanced Database project

Creators:

MAHMOUD AHMED MOHAMED

AHMED ABELHAKEEM AHMED

MAHMOUD AHMED REFAAT

SARA MOHAMED

MARIAM AYMAN

HASNAA KASSAB

Supervisor : Dr Wael Zakaria

Importance of Premier League Database

Why did we create the database?

- For easy return of data if we need it
- Record all stats such as the winning team's top scorer
- Penalties and warnings
- Teams qualified for the Champions League
- Teams qualified for the Europian League
- Teams relegated to the second division league

Requierments

- Let a sample database application, called Premier league keeps track of a Premier league's Team, player, president, Game, coach, punish , score , stadium , Achievement_player and achievement_team.

Data requirements for the Premier league database are:–

- Each Team has a unique name , a Start date , a Home kit , an Away Kit , Captain, and a Total point .
- The Team Has president , coach , stadium , Achievement Team , Player.
- Each President Has a unique Social Security number , a name , and Nationality and is related to one Team one President.
- Each Coach Has a unique Social Security number , a name , and Nationality and related to one Team one Coach.
- Each achievement_team Has a Unique Prize name , and the Number of prizes the team has won and related to the team So that the one team won many prizes.
- Each stadium Has a Unique name , a location , and Capacity and related to one team one Stadium

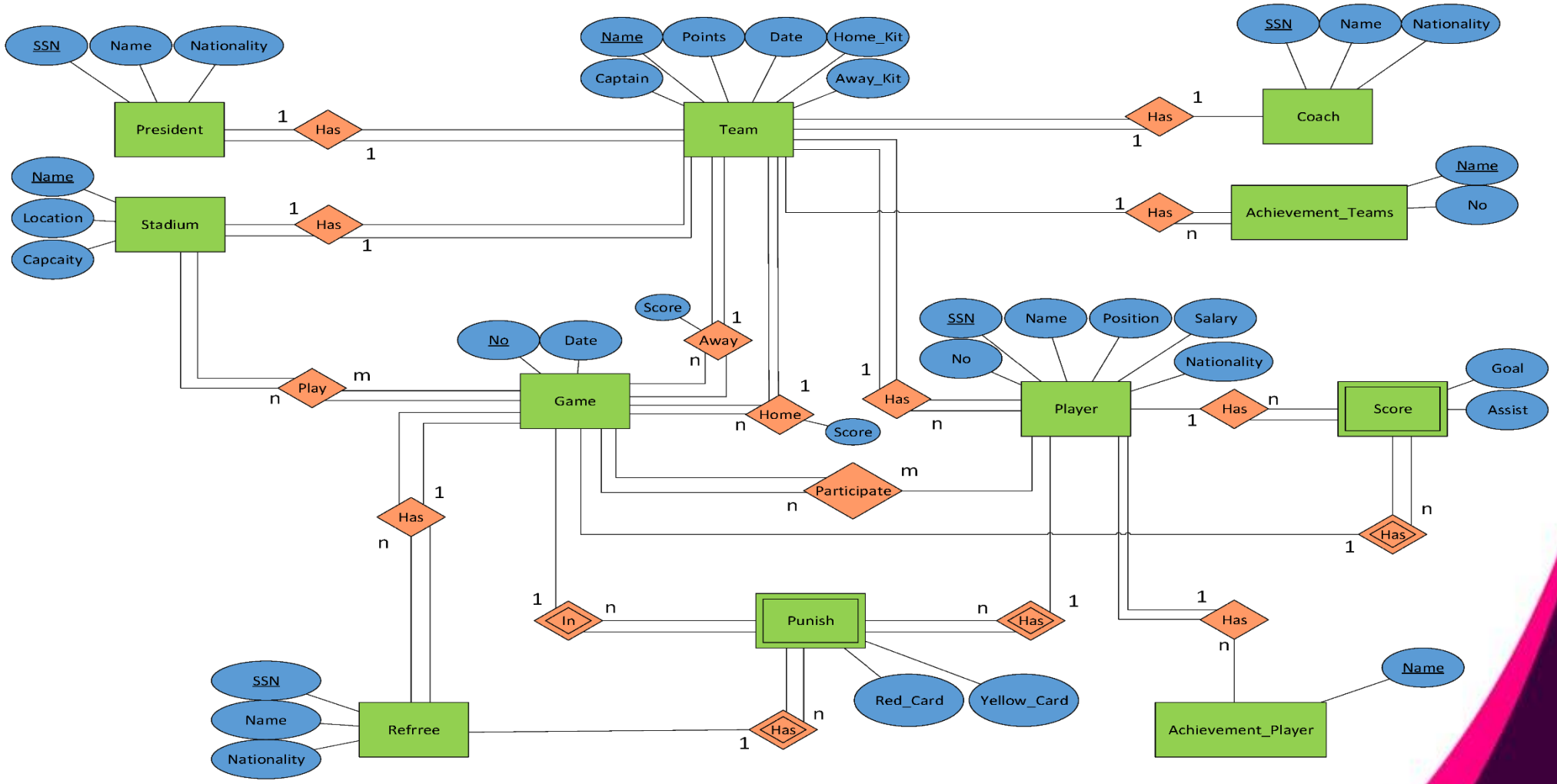
- Each Player Has a unique Social Security number , a name ,a Nationality , a Number player , a position , and Salary and related of team So that the one team has many player.
- The Player has an Achievement player.
- Each Achievement player Has a Unique Prize player name So The Player has won many achievements.
- Each Game Has A unique Number game , and Date of Game and related of A home Team and Away Team.
- A home Team and Away Team HAs Score home and Score away and The home Team and away team has many games.
- The Stadium is related Game So Stadium plays more than one game.
- Each Score Has Goal and assist and related Player so the Player Scores more than or equal one goal or assist and Related Game So in one game , more than one goal or assist can be Scored.
- The player Related game So The player participating in This Game

- Each Referee Has A unique Social Security number , a name , and Nationality and related of Game so one Game one Referee and related of Punish so one Referee give many Punish.
- Each Punish Has a Yellow Card and Red Card And is Related to Game So One Game Can have more than one Punish and Related of Player So One player can have more than one Punish.

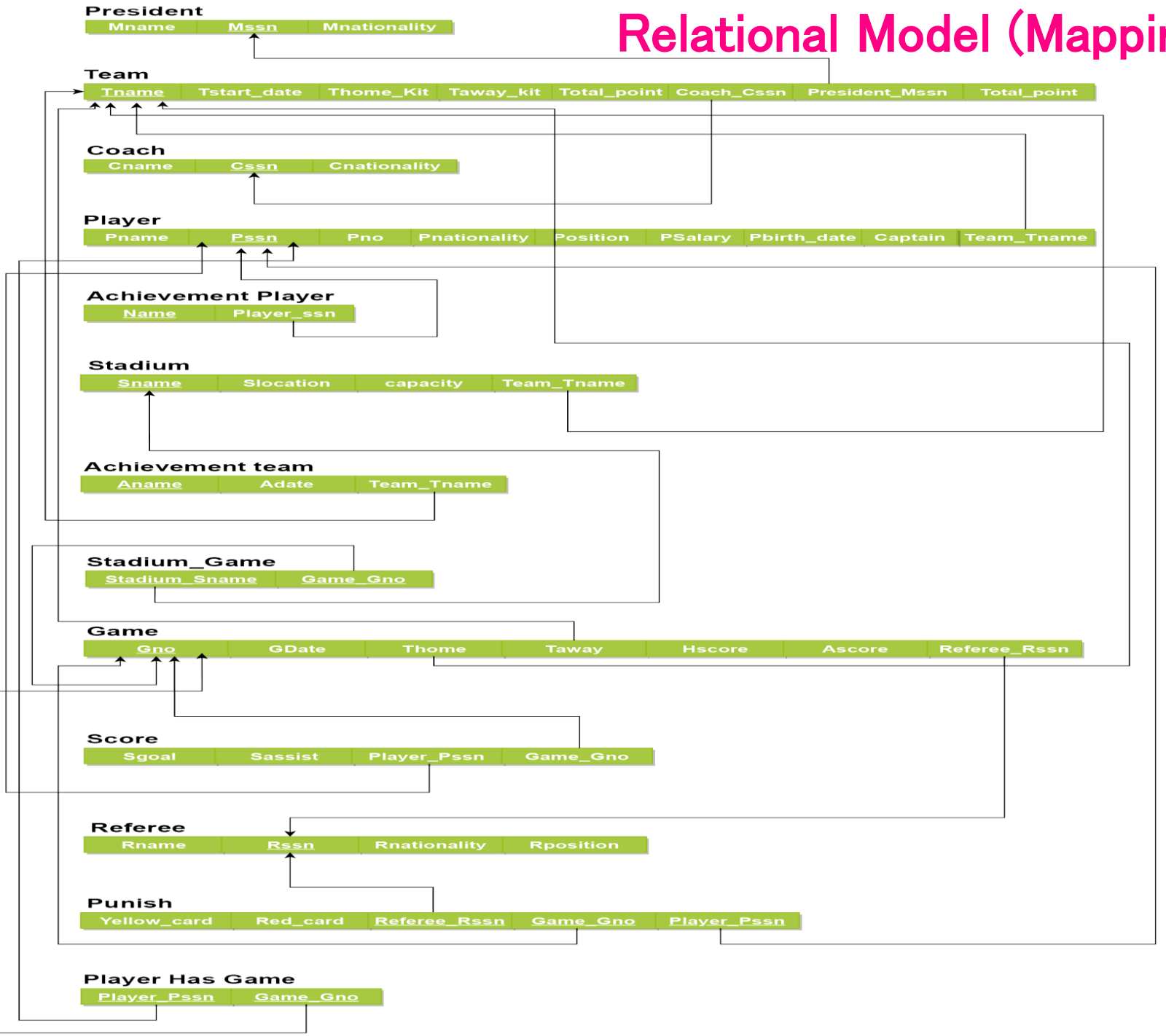
After understanding the requirements of Premier league database create its ER model.

Premier League

ER-Diagram

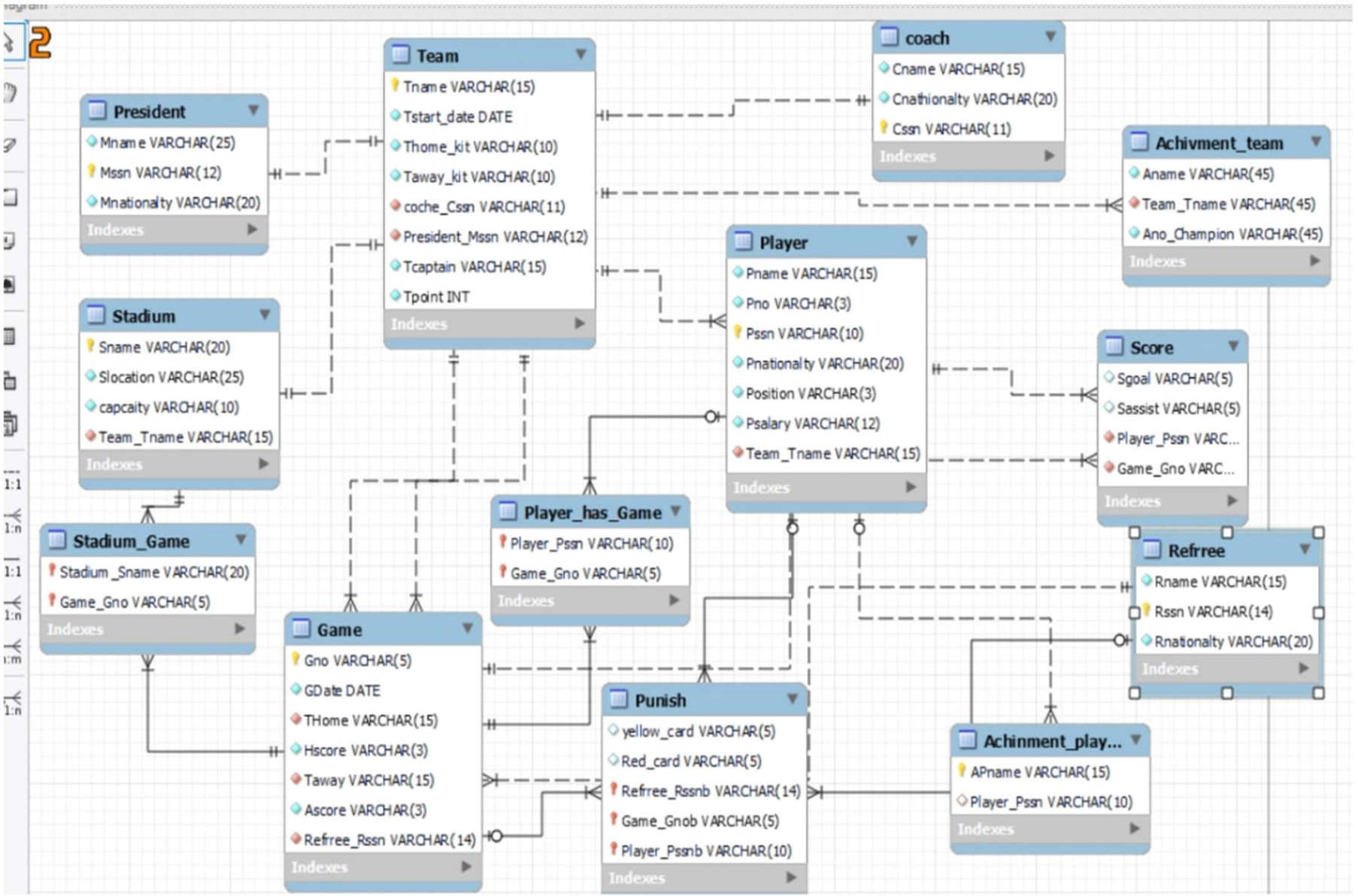

SSN
Name
Nationality
President
Has
Name
Points
Date
Home_Kit
Captain
Away_Kit
Team
SSN
Name
Nationality
Coach
Name
Location
Capcaity
Stadium
Has
Achievement_Teams
Name
No
Score
Away
No
Date
Game
Play
SSN
Name
Position
Salary
No
Nationality
Player
Goal
Assist
Score
Home
Score
Participate
Has
In
Punish
Red_Card
Yellow_Card
SSN
Name
Nationality
Refree
Achievement_Player
Name

Relational Model (Mapping)



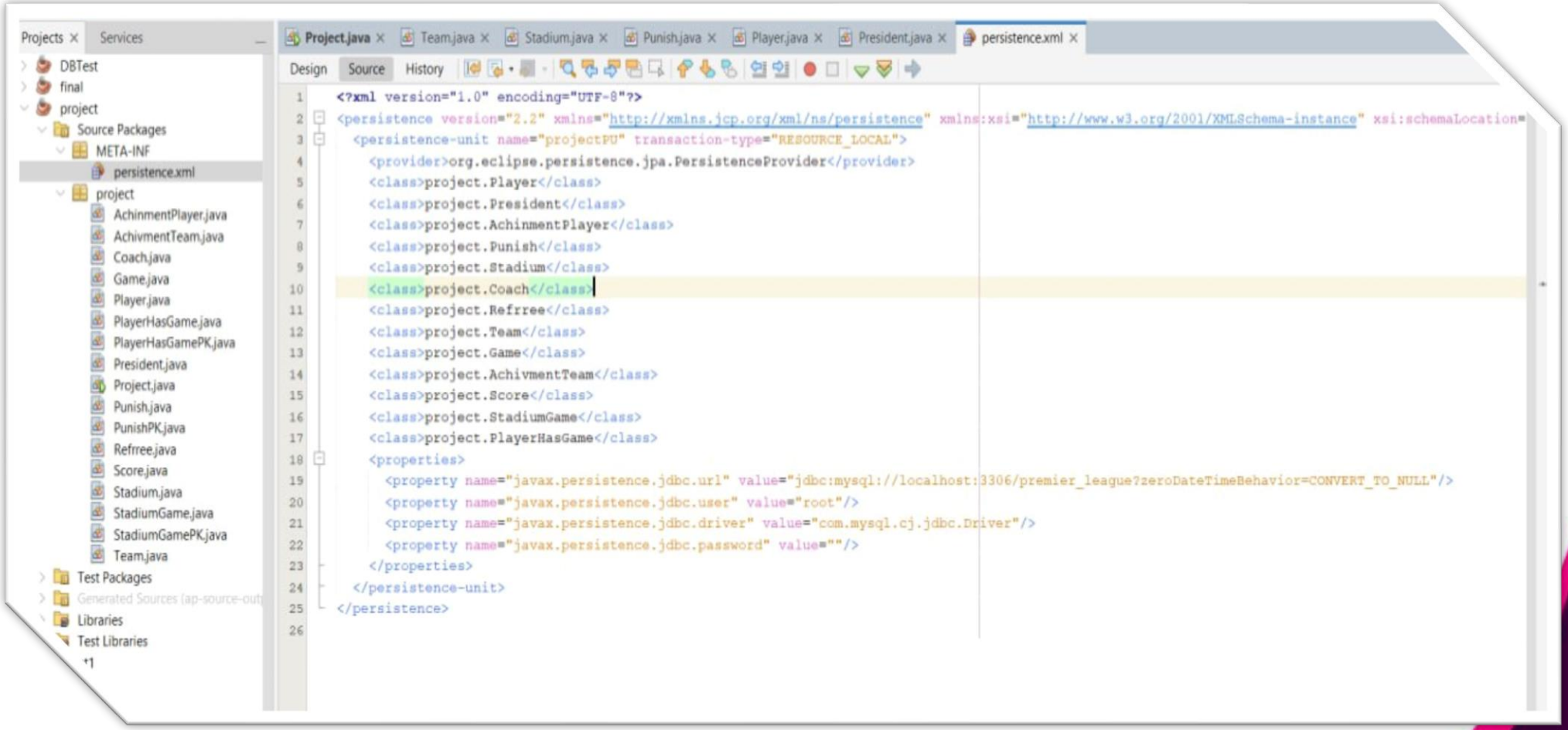
Premier League

ER-Diagram on Work-Bench



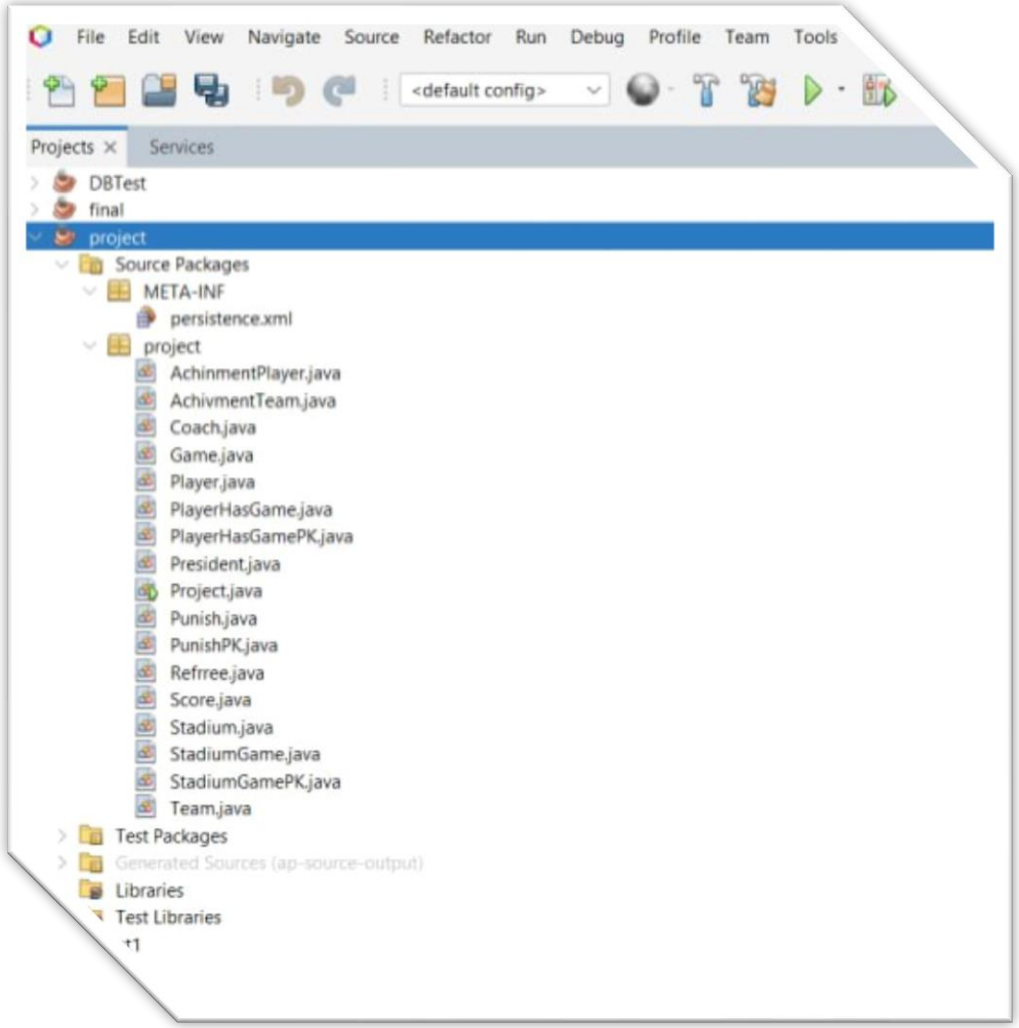
Premier League

XML FILE IN JAVA



Premier League

TABLE CLASSES IN JAVA



Premier League

Team

Tname	Tstart_date	Thome_kit	Taway_kit	coche_Cssn	President_Mssn	Tcaptain
Arsenal	1886-01-01	Red	Black	100	101	Xhaka
Man_city	1880-02-02	Babyblue	White	200	201	Depruyne
Man_united	1878-03-03	Red	Green	300	301	Ronaldo
Scient_soccer	2020-04-04	White	Purble	400	401	Zenhom

Player

Pname	Pno	Pssn	Pnationalty	Position	Psalary	Team_Tname
Ramsdale	1	102	England	GK	63000	Arsenal
Ben_white	2	103	England	CB	6000	Arsenal
Odegard	8	104	Norway	CAM	115000	Arsenal
Xhaka	31	105	Swiss	CDM	15000	Arsenal
Jesus	9	106	Braziliane	ST	13000	Arsenal
Ederson	1	202	Braziliane	GK	180000	Man_city
Cancelo	26	203	Portuguese	LB	180000	Man_city
Depruyne	17	204	Belgium	CAM	104000	Man_city
Silva	24	205	Portuguese	CAM	150000	Man_city
Halaand	9	206	Norway	ST	1000000	Man_city
De_gea	1	302	Spansih	GK	375000	Man_united
Maguire	3	303	England	CB	200000	Man_united
Bruno	17	304	Portuguese	CAM	240000	Man_united
Rashford	16	305	England	LM	225000	Man_united
Ronaldo	7	306	Portuguese	ST	480000	Man_united
Ahmed_emad	1	402	Egyption	GK	1000	Scient_soccer
Mo3tesm	2	403	Egyption	CB	5000	Scient_soccer
Belal	17	404	Egyption	CAM	50000	Scient_soccer
Omar_elqady	26	405	Egyption	CM	30000	Scient_soccer
Zenhom	7	406	Egyption	ST	100000	Scient_soccer

Premier League

coach

Cname	Cnathionalty	Cssn
Arteta	Spanish	100
Guardiola	Spanish	200
Erik_ten_hag	Dutch	300
Zahran	Egyption	400
mansour	Egyption	800
offa	Egyption	900

President

Mname	Mssn	Mnationalty
Korenke	101	American
Khaldoon_almub...	201	Emirati
Joel_glazer	301	American
Wael_Zakaria	401	Egyption

stadium

Sname	Slocation	capcaity	Team_Tname
Emirates	London	60260	Arsenal
Etihad	Manchester	53400	Man_city
Old_trafford	Manchester	74310	Man_united
Cairo	Cairo	75000	Scient_soccer

Player_Award

APname	Player_Pssn
Goal_scorrer	406
Top_assists	404

Team Achievment

Aname	Team_Tname	Ano_Champio
FA_cup	Arsenal	14
FA_cup	Man_city	6
FA_cup	Man_united	12
FA_cup	Scient_soccer	2
Premier_League	Arsenal	13
Premier_League	Man_city	8
Premier_League	Man_united	20
Premier_League	Scient_soccer	1

Referee

Rname	Rssn	Rnationalty
Anthony_taylor	500	England
Michael_oliver	501	England
Craig_bawson	502	England
Jarred_jilletg	503	England

Starting_players

Player_Pssn	Game_Gno
102	1
103	1
104	1
106	1
301	1
302	1
303	1
305	1
306	1
102	2
103	2
105	2
106	2
202	2
203	2
204	2
206	2

Stadium's game week

	Stadium_Sname	Game_Gno
▶	Emirates	1
	Etihad	7
	Old_trafford	4
	Cairo	10
	Emirates	2
	Emirates	3
	Old_trafford	5
	Old_trafford	6
	Etihad	8
	Etihad	9
	Cairo	11
	Cairo	12

Score

	Sgoal	Sassist	Player_Pssn	Game_Gno
▶	1	0	106	1
	1	1	104	1
	1	0	306	1
	0	1	305	1
	1	0	106	2
	1	0	206	2
	0	1	204	2
	1	0	406	3
	0	1	404	3
	1	0	103	4
	1	0	105	4
	0	1	106	4
	1	0	306	5
	1	0	305	5
	0	1	305	5
	1	1	404	6
	1	0	405	6
	0	1	406	6

Premier League

Game

Gno	GDate	THome	Hscore	Taway	Ascore	Refrree_Rssn
1	2022-06-01	Arsenal	2	Man_united	1	500
2	2022-06-04	Arsenal	1	Man_city	1	501
3	2022-06-07	Arsenal	0	Scient_soccer	1	503
4	2022-06-10	Man_united	0	Arsenal	2	502
5	2022-06-13	Man_united	2	Man_city	0	502
6	2022-06-16	Man_united	0	Scient_soccer	2	500
7	2022-06-19	Man_city	1	Arsenal	3	500
8	2022-06-22	Man_city	1	Man_united	0	503
9	2022-06-25	Man_city	1	Scient_soccer	1	501
10	2022-06-28	Scient_soccer	0	Arsenal	0	502
11	2022-07-01	Scient_soccer	2	Man_united	1	50
12	2022-07-03	Scient_soccer	3	Man_city	0	

Punish

	yellow_card	Red_card	Refree_Rssnb	Game_Gnob	Player_Pssnb
▶	1	NULL	500	1	106
	1	NULL	502	3	404
	NULL	1	502	3	406
	1	NULL	503	4	303
	1	NULL	500	5	206
	1	NULL	500	5	306
	1	NULL	501	6	402
	1	NULL	501	6	303
	1	NULL	502	7	203
	1	NULL	503	8	206
	1	NULL	500	9	402
	1	NULL	502	11	303
	1	NULL	503	12	405

Premier League

QUERIES

1. Some Query from one table:

- **Retrieve the Team name , Start date of team , Home Kit and away kit of the team.**

```
1  select  Tname As Team_Name,Tstart_date AS Start_Date,Thome_kit AS Home_Kit,Taway_kit as Away_Kit
2  from Team;
3
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Team_Name	Start_Date	Home_Kit	Away_Kit
Arsenal	1886-01-01	Red	Black
Man_city	1880-02-02	Babyblue	White
Man_united	1878-03-03	Red	Green
Scient_soccer	2020-04-04	White	Purble

Premier League

QUERIES IN JAVA

1. ON JAVA:

```
Source  History
1  package project;
2
3  import java.util.List;
4  import javax.persistence.EntityManager;
5  import javax.persistence.Persistence;
6
7  public class Project
8  {
9      public static void main(String[] args)
10     {
11         EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12         em.getTransaction().begin();
13         List<Team> Teams = em.createNamedQuery(string:"Team.findAll").getResultList();
14         System.out.println(x: "First Query: Teasms");
15         System.out.format(format:"%10s %40s %25s %25s\n",args: "Team Name",args: "Start Date", args: "Home Kit",args: "Away Kit");
16         for(Team team : Teams)
17         {
18             System.out.format(format:"%10s %40s %25s %25s\n",args: team.getTname(),args: team.getTstartdate(), args: team.getThomekit(),args: team.getTawaykit());
19         }
20
21
22     }
23
24 }
25
```

Output - project (run)

```
First Query: Teasms
 Team Name                                 Start Date                 Home Kit                  Away Kit
   Arsenal              Fri Jan 01 00:00:00 EET 1886                      Red                     Black
  Man_city              Mon Feb 02 00:00:00 EET 1880                Babyblue                     White
Man_united              Sun Mar 03 00:00:00 EET 1878                      Red                     Green
Scient_soccer              Sat Apr 04 00:00:00 EET 2020                    White                     Purble
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve the player name , player number , player nationality , Player position and salary of the player whose Team_Tname is 'Man_city'.

achinment_player | person

Limit to 1000 rows

```
1  select  Pname As Player_Name, pno AS Player_Number,Pnationalty AS Nationalty,Position ,Psalary As Salary
2  from player
3  where Team_Tname="Man_city";
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Player_Name	Player_Number	Nationalty	Position	Salary
Ederson	1	Braziliane	GK	180000
Cancelo	26	Portuguese	LB	180000
Depruyne	17	Belgium	CAM	104000
Silva	24	Portuguese	CAM	150000
Halaand	9	Norway	ST	1000000

Premier League

QUERIES IN JAVA

Source History

```
4  import javax.persistence.EntityManager;
5  import javax.persistence.Persistence;
6
7  public class Project
8  {
9      public static void main(String[] args)
10     {
11         EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12         em.getTransaction().begin();
13         List<Player> Players = em.createNamedQuery(string:"Player.findAll").getResultList();
14         System.out.println(x: "Second Query: PLayers");
15         System.out.format(format:"%10s %10s %10s %10s %10s\n",args: "Player Name",args: "Player Number",args: "Nationalty", args: "Position",args: "Salary");
16         for(Player Player : Players)
17         {
18             if("Man_city".equals(anObject: Player.getTeamTname().getTname()))
19             {
20                 System.out.format(format:"%10s %10s %10s %10s %10s\n",
21                         args: Player.getPname(),args: Player.getPno(),args: Player.getPnationalty(),args: Player.getPosition(),args: Player.getPsalary());
22             }
23         }
24     }
25
```

Output

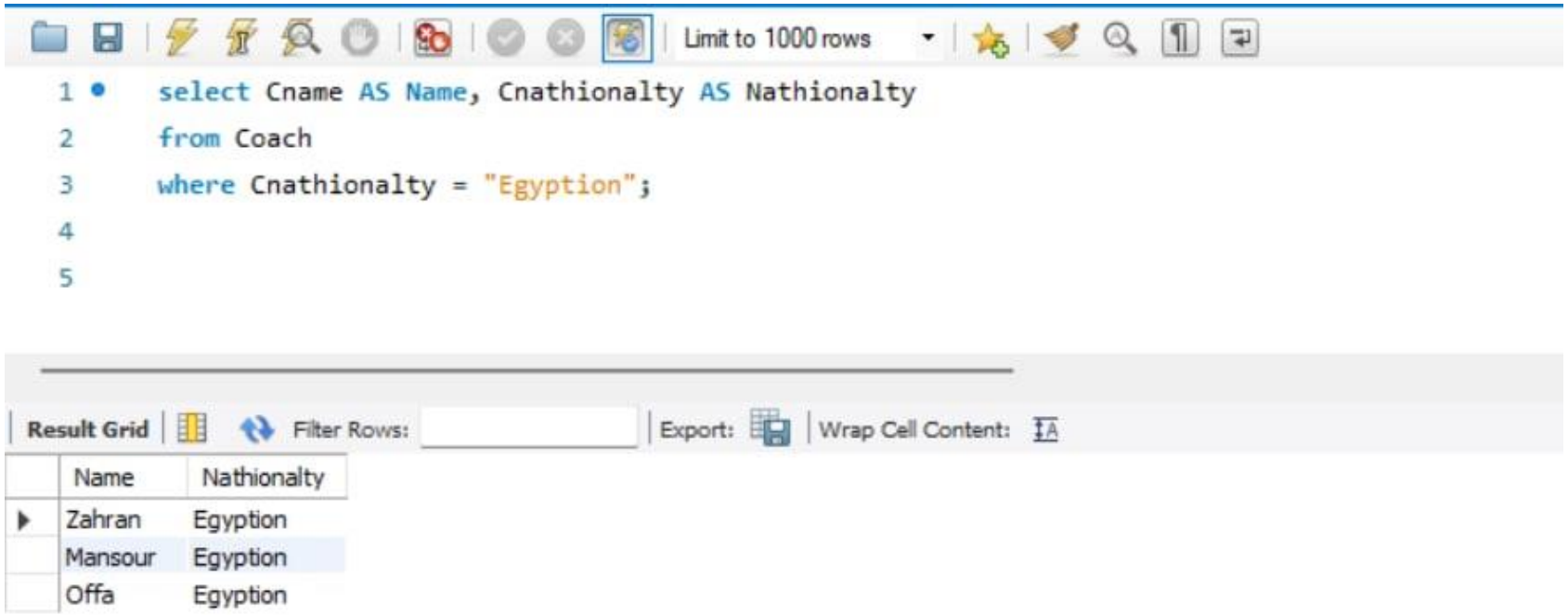
SQL 3 execution | project (run)

```
[EL Info]: 2024-05-01 02:07:42.149--ServerSession(1346343363)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
Second Query: PLayers
Player Name Player Number Nationalty   Position     Salary
   Ederson             1 Braziliane         GK     180000
   Cancelo            26 Portuguese         LB     180000
  Depruyne            17    Belgium        CAM     104000
     Silva            24 Portuguese        CAM     150000
   Halaand             9     Norway         ST    1000000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve coach name and coach nationality of the coach Whose Cnathionalty is 'Egyption'.



Premier League

QUERIES IN JAVA

Source History

```java
package project;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
        em.getTransaction().begin();
        List<Coach> Coachs = em.createNamedQuery(string:"Coach.findAll").getResultList();
        System.out.println(x: "Third Query: Coachs");
        System.out.format(format:"%20s %20s\n",args: "Coach Name", args: "Nationalty");
        for(Coach Coach : Coachs)
        {
            if(Coach.getCnathionalty().equals(anObject: "Egyption"))
                System.out.format(format:"%20s %20s\n",args: Coach.getCname(),args: Coach.getCnathionalty());
        }
    }

```

Output

SQL 3 execution × project (run) ×

```
run:
[EL Info]: 2024-05-01 02:13:12.484--ServerSession(1003292107)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5
Third Query: Coachs
          Coach Name           Nationalty
              Zahran             Egyption
             Mansour             Egyption
                Offa             Egyption
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve the Team Name , Prize Name of the Achivment Team who has Fa_cup and greater than 6 awards

```
1 •  Select Team_Tname as team_name , Aname as prize_name
2    from achivment_team
3    where  Aname ="FA_cup" and Ano_Champion > 6;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

team_name	prize_name
Arsenal	FA_cup
Man_united	FA_cup

Premier League

QUERIES IN JAVA

Source History

```
4  import javax.persistence.EntityManager;
5  import javax.persistence.Persistence;
6
7  public class Project
8  {
9      public static void main(String[] args)
10     {
11         EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12         em.getTransaction().begin();
13         List<AchivmentTeam> AchivmentTeams = em.createNamedQuery(string:"AchivmentTeam.findAll").getResultList();
14         System.out.println(x: "Fourth Query");
15         System.out.format(format:"%20s %20s %20s\n",args: "Team Name",args: "Prize Name", args: "Prize Number");
           for(AchivmentTeam achivmentTeam : AchivmentTeams)
17         {
18             if(achivmentTeam.getAname().equals(anObject: "FA_cup") && achivmentTeam.getAnoChampion() > 6)
19             {
20                 System.out.format(format:"%20s %20s %20s\n",args: achivmentTeam.getTeamTname().getTname(),args: achivmentTeam.getAname(),args: achivmentTeam.getAnoChampion());
21             }
22         }
23
24
25     }
```

Output

SQL 3 execution × | project (run) ×

```
run:
[EL Info]: 2024-05-01 03:27:22.408--ServerSession(1096485705)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
Fourth Query
           Team Name           Prize Name         Prize Number
             Arsenal              FA_cup                   14
          Man_united              FA_cup                   12
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve all details the Game who The result of all game ended 2 - 1

```
1  select *
2  from Game
3  where Hscore=2 and Ascore=1;
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Gno	GDate	THome	Hscore	Taway	Ascore	Refrree_Rssn
1	2022-06-01	Arsenal	2	Man_united	1	500
11	2022-07-01	Scient_soccer	2	Man_united	1	500
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Premier League

QUERIES IN JAVA

Source History

```
7   public class Project
8   {
9       public static void main(String[] args)
10      {
11          EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12          em.getTransaction().begin();
13          List<Game> Games = em.createNamedQuery(string:"Game.findAll").getResultList();
14          System.out.println(x: "Fourth Query");
15          System.out.format(format:"%10s %35s %15s %10s %15s %10s\n",args: "Game Number",args: "Game Date", args: "Thome" , args: "Hscore", args: "Taway",args: "Ascore" , args: "Reffrree SSN");
16          for(Game Game : Games)
17          {
18              if(Game.getHscore().equals(anObject: "2") && Game.getAscore().equals(anObject: "1"))
19              {
20                  System.out.format(format:"%10s %35s %15s %10s %15s %10s\n",
21                          args: Game.getGno(),args: Game.getGDate(),
22                          args: Game.getTHome().getTname(),args: Game.getHscore(),
23                          args: Game.getTaway().getTname(),args: Game.getAscore(),
24                          args: Game.getRefrreeRssn());
25              }
26          }
27
28
```

Output

SQL 3 execution | project (run)

```
run:
[EL Info]: 2024-05-01 03:39:21.044--ServerSession(1346343363)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
Fourth Query
Game Number                         Game Date           Thome     Hscore          Taway     Ascore
          1        Wed Jun 01 00:00:00 EET 2022         Arsenal          2      Man_united          1
         11        Fri Jul 01 00:00:00 EET 2022   Scient_soccer          2      Man_united          1
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

2. Some Query from Many Tables:

- **Retrieve the player name And Achievement_Player who has an achievement**

```
1 •  select Apname , pname
2    from player , achinment_player
3    where Player_Pssn=Pssn;
4
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Apname	pname
Top_assists	Belal
Goal_scorrer	Zenhom

Premier League

QUERIES IN JAVA

2. Some Query from Many Tables IN JAVA:

Source | History

```
1  package project;
2
3  import java.util.List;
4  import javax.persistence.EntityManager;
5  import javax.persistence.Persistence;
6
7  public class Project
8  {
9      public static void main(String[] args)
10     {
11         EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12         em.getTransaction().begin();
13         List<AchinmentPlayer> AchinmentPlay = em.createNamedQuery(string:"AchinmentPlayer.findAll").getResultList();
14         System.out.format(format:"%20s %20s\n",args: "Prize Name",args: "Player Name");
15         for(AchinmentPlayer AchinmentPlayer : AchinmentPlay)
16         {
17             System.out.format(format:"%20s %20s\n",args: AchinmentPlayer.getAPname(),args: AchinmentPlayer.getPlayerPssn().getPname());
18         }
19
20
21     }
22
```

Output

SQL 3 execution × | project (run) ×

```
run:
[EL Info]: 2024-05-01 03:44:43.563--ServerSession(1003292107)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
          Prize Name          Player Name
         Top_assists                Belal
        Goal_scorrer               Zenhom
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve The Game number , Refrree name , player name who The player who received a one yellow card in the game

```
1 •  select Gno As game_number , Rname as Refree_name , pname as Player_name
2    from Game, refrree, player ,punish,player_has_game
3    where yellow_card=1 and Rssn=Refrree_Rssn and Gno=Game_Gno and Gno=Game_Gnob and Pssn=Player_Pssn and pssn=Player_Pssnb;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

game_number	Refree_name	Player_name
1	Anthony_taylor	Jesus
11	Anthony_taylor	Maguire
6	Anthony_taylor	Maguire
6	Anthony_taylor	Ahmed_emad
7	Anthony_taylor	Cancelo
12	Michael_oliver	Omar_elqady
9	Michael_oliver	Ahmed_emad
4	Craig_bawson	Maguire
5	Craig_bawson	Halaand
5	Craig_bawson	Ronaldo
3	Jarred_jilletg	Belal
8	Jarred_jilletg	Halaand

Premier League

QUERIES IN JAVA

```java
public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
        em.getTransaction().begin();
        List<Punish> punish = em.createNamedQuery(string:"Punish.findAll").getResultList();
        System.out.format(format:"%20s %20s %20s\n",args: "Game Number", args: "Refrree Name",args: "Player Name");
        for(Punish pun : punish)
        {
            if(pun.getYellowCard().equals(anObject: "1"))
            {
                System.out.format(format:"%20s %20s %20s\n", args: pun.getGame().getGno() , args: pun.getRefrree().getRname(),args: pun.getPlayer().getPname());
            }
        }
    }
```

```
run:
[EL Info]: 2024-05-01 04:00:00.729--ServerSession(391135083)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
         Game Number         Refrree Name          Player Name
                   1       Anthony_taylor                Jesus
                   5       Anthony_taylor              Halaand
                   5       Anthony_taylor              Ronaldo
                   9       Anthony_taylor           Ahmed_emad
                   6       Michael_oliver              Maguire
                   6       Michael_oliver           Ahmed_emad
                  11         Craig_bawson              Maguire
                   3         Craig_bawson                Belal
                   7         Craig_bawson              Cancelo
                  12       Jarred_jilletg           Omar_elqady
                   4       Jarred_jilletg              Maguire
                   8       Jarred_jilletg              Halaand
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve The Player name , player position , Sgoal , assists and Number game So That Zenhom Or Omar elqady made a goal or assist in a match

```
1  select pname as 'player_name' , position , sgoal as goal , Sassist as assists , gno as game_week
2   from player , score , game
3   where Pssn=Player_Pssn and Gno=Game_Gno
4   and Pname IN('omar_elqady' , 'zenhom');
5
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

player_name	position	goal	assists	game_week
Zenhom	ST	1	0	11
Omar_elqady	CM	2	0	12
Zenhom	ST	1	0	3
Omar_elqady	CM	1	0	6
Zenhom	ST	0	1	6
Zenhom	ST	1	0	9

Premier League

QUERIES IN JAVA

Source History

```java
package project;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
        em.getTransaction().begin();
        List<Score> Players = em.createNamedQuery(string:"Score.findAll").getResultList();
        System.out.format(format:"%20s %7s %7s %7s %7s\n",args: "Player Name" , args: "Position", args: "Goal", args: "Assists",args: "Game Week");
        for(Score play : Players)
        {
            if(play.getPlayerPssn().getPname().equals(anObject: "Zenhom") || play.getPlayerPssn().getPname().equals(anObject: "Omar_elqady"))
            {
                System.out.format(format:"%20s %7s %7s %7s %7s\n",
                        args: play.getPlayerPssn().getPname(), args: play.getPlayerPssn().getPosition(),
                        args: play.getSgoal(), args: play.getSassist(),
                        args: play.getGameGno().getGno()
                        );
```

Output

SQL 3 execution × project (run) ×

```
run:
[EL Info]: 2024-05-01 04:15:54.684--ServerSession(1330247343)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
         Player Name Position    Goal Assists Game Week
              Zenhom       ST       1       0        3
         Omar_elqady       CM       1       0        6
              Zenhom       ST       0       1        6
              Zenhom       ST       1       0        9
              Zenhom       ST       1       0       11
         Omar_elqady       CM       2       0       12
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- Retrieve the stadium name, the Team Away, A Game number Whose stadium name is "old Trafford" and Awayscore is greater than Homescore.

```
Limit to 1000 rows

1  Select Sname AS Stadium , Taway As Team , Gno as game_week
2   from stadium,Game,stadium_game
3   where Ascore > Hscore and Sname = "Old_trafford" and  Sname=Stadium_Sname and Game_Gno=Gno;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Stadium	Team	game_week
Old_trafford	Arsenal	4
Old_trafford	Scient_soccer	6

Premier League

QUERIES IN JAVA

Source History

```java
package project;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
        em.getTransaction().begin();
        List<StadiumGame> stad = em.createNamedQuery(string: "StadiumGame.findAll").getResultList();
        System.out.format(format: "%20s %20s %7s\n",args: "Stadium Name" , args: "Team Name",args: "Game Week");
        for(StadiumGame st : stad)
        {
            if(st.getStadium().getSname().equals(anObject: "Old_trafford") && st.getGame().getAscore() > st.getGame().getHscore())
            {
                System.out.format(format: "%20s %20s %7s\n",
                        args: st.getStadium().getSname(),args: st.getGame().getTaway().getTname() , args: st.getGame().getGno()
                        );
            }
        }
```

Output

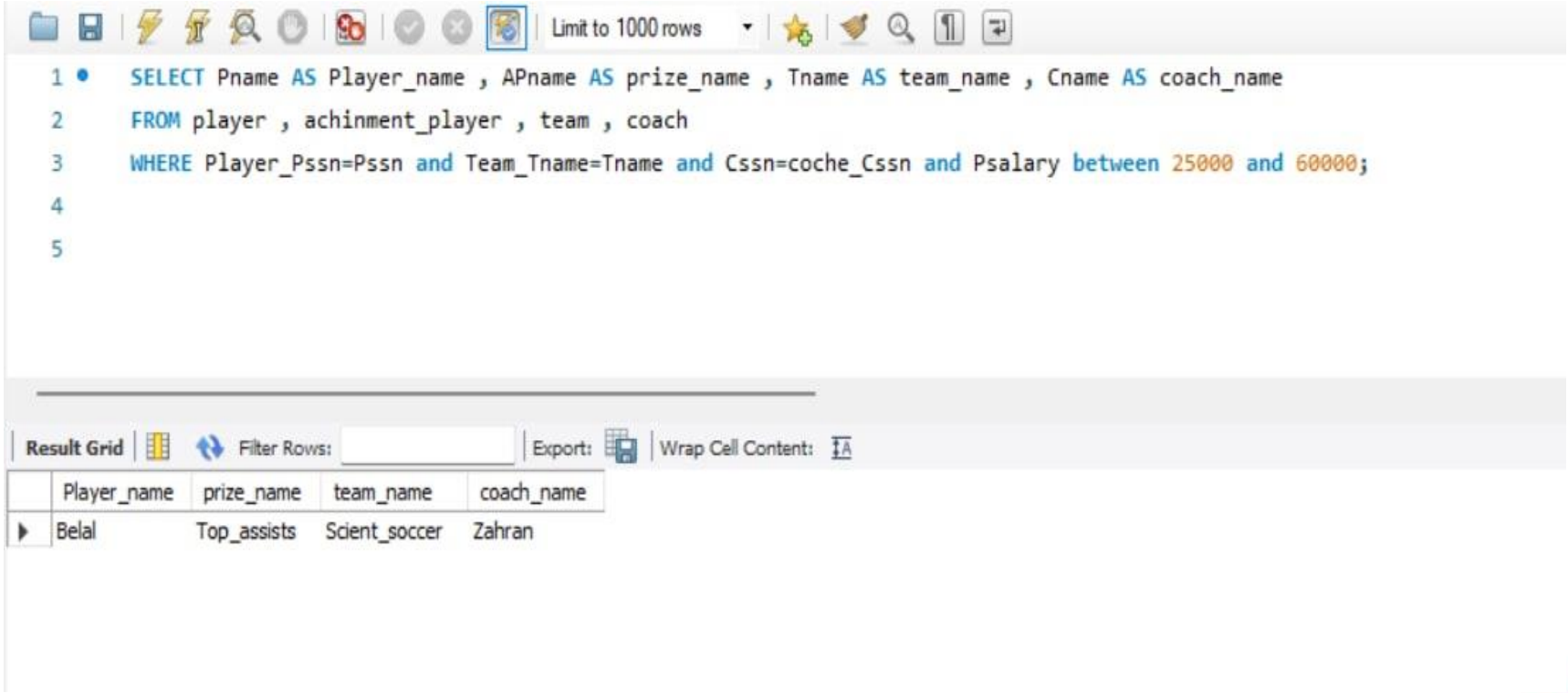
SQL 3 execution × project (run) ×

```
run:
[EL Info]: 2024-05-01 04:37:08.868--ServerSession(418958713)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
        Stadium Name            Team Name Game Week
        Old_trafford              Arsenal       4
        Old_trafford        Scient_soccer       6
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- Retrieve the player name , Prize Name of achinment_player , team name and coach_name So that Psalary of player between 25000 and 60000.



```
1  SELECT Pname AS Player_name , APname AS prize_name , Tname AS team_name , Cname AS coach_name
2  FROM player , achinment_player , team , coach
3  WHERE Player_Pssn=Pssn and Team_Tname=Tname and Cssn=coche_Cssn and Psalary between 25000 and 60000;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Player_name	prize_name	team_name	coach_name
Belal	Top_assists	Scient_soccer	Zahran

QUERIES IN JAVA

Source History

```
6
7   public class Project
8   {
9       public static void main(String[] args)
10      {
11          EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
12          em.getTransaction().begin();
13          List<AchinmentPlayer> Aplayers = em.createNamedQuery(string: "AchinmentPlayer.findAll").getResultList();
14          System.out.format(format: "%20s %20s %20s %20s\n",args:"Player Name" ,args:"Prize Name", args:"Team Name",args:"Coach Name");
15          for(AchinmentPlayer p : Aplayers)
16          {
17              if(p.getPlayerPssn().getPsalary() >= 25000 && p.getPlayerPssn().getPsalary() <= 60000)
18              {
19                  System.out.format(format: "%20s %20s %20s %20s\n",
20                          args:p.getPlayerPssn().getPname(),args:p.getAPname(),
21                          args:p.getPlayerPssn().getTeamTname().getTname(),
22                          args:p.getPlayerPssn().getTeamTname().getCocheCssn().getCname()
23                          );
24              }
25          }
26      }
27  }
```

Output

SQL 3 execution X | project (run) X

```
run:
[EL Info]: 2024-05-01 04:47:19.953--ServerSession(1885922916)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
         Player Name           Prize Name           Team Name          Coach Name
              Belal         Top_assists       Scient_soccer              Zahran
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

3. Some Query Using Aggregate Functions:

- **Retrieve the total number of Team name in the team whose The capacity of stadium greater than 65000.**

```
1 •  SELECT COUNT(Tname) AS number_of_Teams_in_league
2    FROM team , stadium
3    where capcaity > 65000 and Team_Tname=Tname;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

number_of_Teams_in_league
2

Premier League

QUERIES IN JAVA

3. Some Query Using Aggregate Functions IN JAVA:

Source | History

```java
package project;

import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
        em.getTransaction().begin();
        List capacity = em.createNamedQuery(string: "Stadium.findCapacityTeam").setParameter(string: "scapacity", o: 65000).getResultList();
        System.out.format(format: "%20s\n",args: "Number of team" );
        for(Object object : capacity)
        {
            System.out.format(format: "%20s\n",args: object);
        }
    }
}
```

Output

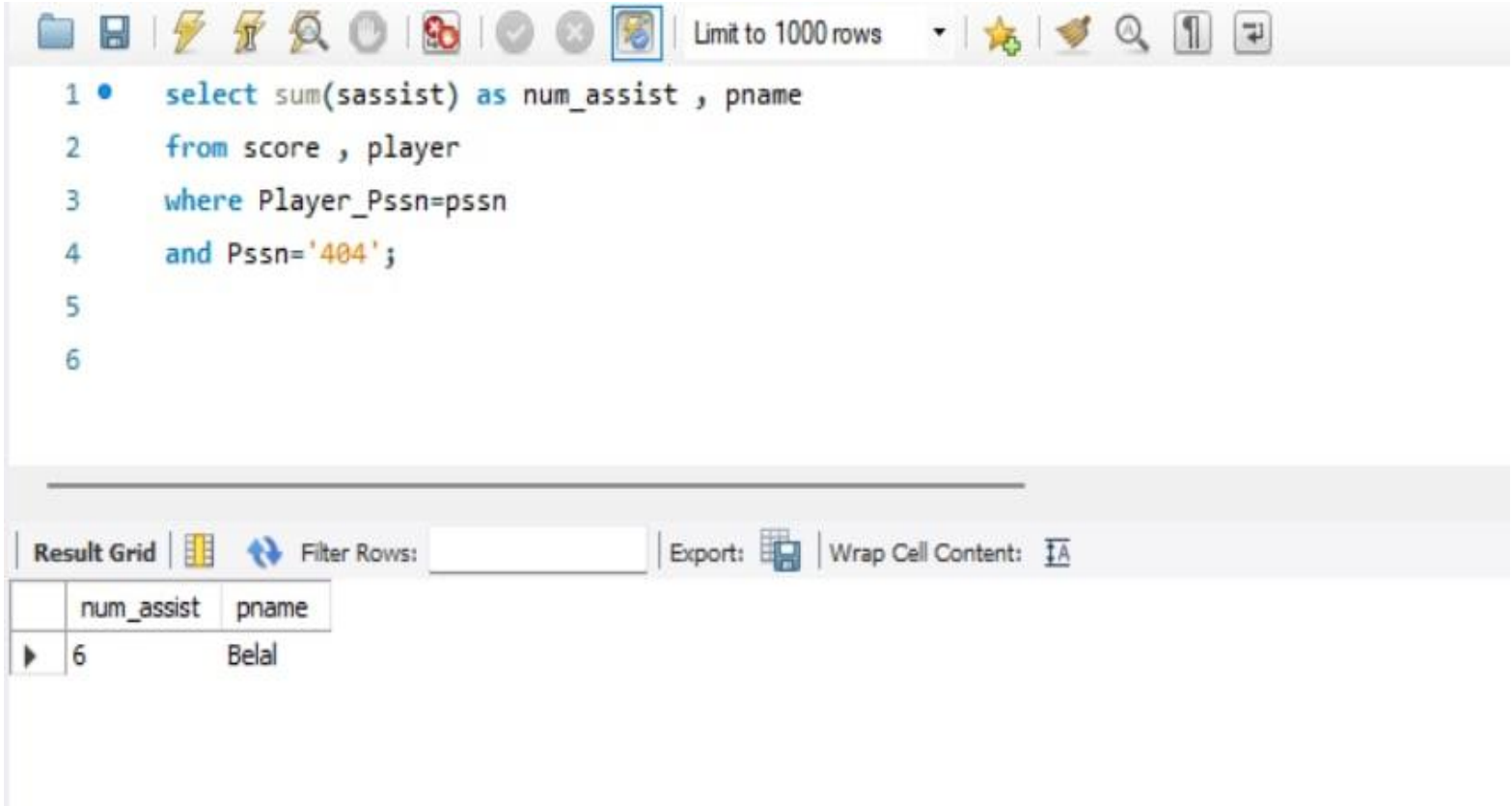
SQL 3 execution × | project (run) ×

```
run:
[EL Info]: 2024-05-01 05:35:31.857--ServerSession(605052357)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
      Number of team
                   2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- The sum of assist of player whose pssn is '404' and retrieve name of player



Premier League

QUERIES IN JAVA

```java
public class Project
{
    public static void main(String[] args)
    {
        int total = 0;
        String name = "";
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
        em.getTransaction().begin();
        List<Score> sum = em.createNamedQuery(string: "Score.findAll").getResultList();
        System.out.format(format: "%20s %20s\n",args: "All Number Assist" , args: "Player Name" );
        for(Score s : sum)
        {
                if(s.getPlayerPssn().getPssn().equals(anObject: "404"))
                {
                    total += Integer.parseInt(s: s.getSassist());
                    name = s.getPlayerPssn().getPname();
                }
        }
         System.out.format(format: "%20s %20s\n",args: total , args: name );
    }
```

Output

SQL 3 execution | project (run)

```
run:
[EL Info]: 2024-05-01 06:10:47.401--ServerSession(997033037)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
   All Number Assist          Player Name
                   6                Belal
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

4. Some Query Using Aggregate Functions and having or grouping or order by :

- **Find The Sum of The Score goal for every Team.**

```
1  Select sum(Sgoal) As Number_of_Goal,Tname As Team
2  From Score , team , Player
3  where Tname=Team_Tname and Player_Pssn=Pssn
4  group by Tname;
5
6
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Number_of_Goal	Team
8	Arsenal
4	Man_city
4	Man_united
9	Scient_soccer

Premier League

QUERIES IN JAVA

```
Source  History
10      {
11          int tArsenal = 0, tCity = 0, tscince = 0, tunited = 0;
12          EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
13          em.getTransaction().begin();
14          List<Score> sum = em.createNamedQuery(string: "Score.findAll").getResultList();
15          System.out.format(format: "%20s %20s\n",args:"All Number Goal" , args:"Team Name" );
16          for(Score s : sum)
17          {
18              String str = s.getPlayerPssn().getPssn().substring(beginIndex: 0,endIndex:1);
19              try {
20                  int goals = Integer.parseInt(s: s.getSgoal());
21                  if(str.equals(anObject:"1"))
22                  {
23                      tArsenal += goals;
24                  }
25                  else if(str.equals(anObject:"2"))
26                  {
27                      tCity += goals;
28                  }
29                  else if(str.equals(anObject:"3"))
30                  {
31                      tunited += goals;
32                  }
33                  else if(str.equals(anObject:"4"))
34                  {
35                      tscince += goals;
36                  }
37              } catch (NumberFormatException e)
38              {
39                  // Handle the error, log it, or set a default value
40                   //System.out.println("Error parsing goals for player: " + s.getPlayerPssn().getPssn());
41              }
42          }
43           System.out.format(format: "%20s %20s\n",args:tArsenal , args:"Arsenal" );
44           System.out.format(format: "%20s %20s\n",args:tCity , args:"Man_city" );
45           System.out.format(format: "%20s %20s\n",args:tunited , args:"Man_united" );
46           System.out.format(format: "%20s %20s\n",args:tscince , args:"Scient_soccer" );
47      }
```

SQL 3 execution X project (run) X

```
run:
[EL Info]: 2024-05-01 06:44:21.206--ServerSessi
     All Number Goal             Team Name
                   8               Arsenal
                   4              Man_city
                   4            Man_united
                   9         Scient_soccer
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- Retrieve The Team name and The Average of player Salary for each player Whose greater than 40000 and Descending Order.

```
1  select avg(Psalary) as avg_salary , Tname
2  from player , team
3  WHERE Tname=Team_Tname
4  group by Tname
5  having avg_salary > 40000
6  order by avg_salary desc;
7
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg_salary	Tname
322800	Man_city
304000	Man_united
42400	Arsenal

Premier League

QUERIES IN JAVA

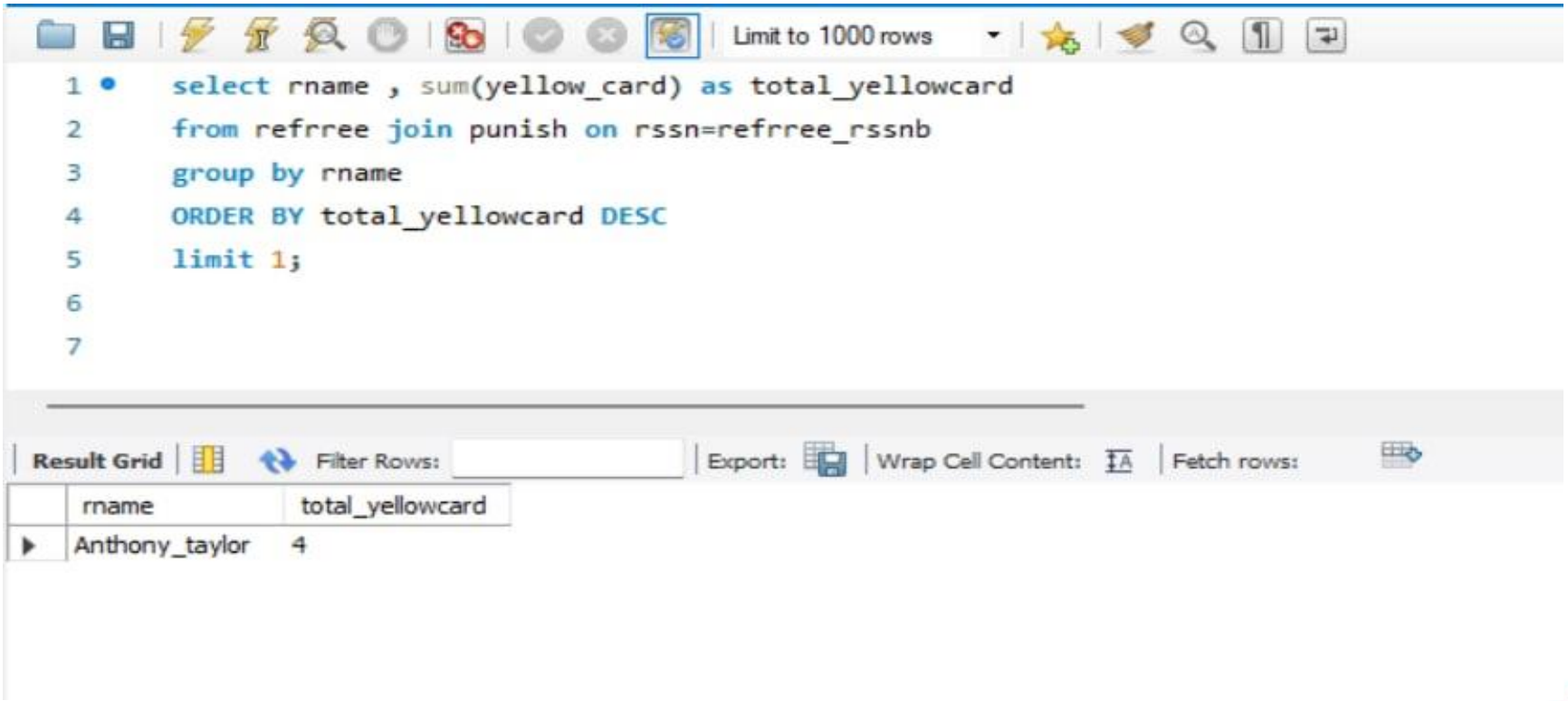
```java
public static void main(String[] args)
{
    int tArsenal = 0, tCity = 0 , tunited =0, tscince = 0;
    String name = "";
    EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
    em.getTransaction().begin();
    List<Player> avg = em.createNamedQuery(string: "Player.findAll").getResultList();
    System.out.format(format: "%20s %20s\n",args: "Average Salary" , args: "Team Name" );
    for(Player a : avg)
    {
    String str = a.getPssn().substring(beginIndex:0,endIndex: 1);
    try {
        int salary = a.getPsalary();
        if(str.equals(anObject: "1"))
        {
            tArsenal += salary;
        }
        else if(str.equals(anObject: "2"))
        {
            tCity += salary;
        }
        else if(str.equals(anObject: "3"))
        {
            tunited += salary;
        }
        else if(str.equals(anObject: "4"))
        {
            tscince += salary;
        }
        } catch (NumberFormatException e)
        {
        }
    }
    if(tArsenal / 5 > 40000)
    {
        System.out.format(format: "%20s %20s\n",tArsenal / 5 , args: "Arsenal" );
    }
    if(tCity / 5 > 40000)
    {
        System.out.format(format: "%20s %20s\n",tCity / 5, args: "Man_city" );
    }
    if(tunited / 5 > 40000)
    {
        System.out.format(format: "%20s %20s\n",tunited / 5 , args: "Man_united" );
    }
    if(tscince / 5 > 40000)
    {
        System.out.format(format: "%20s %20s\n",tscince / 5 , args: "Scient_soccer" );
    }
}
```

```
run:
[EL Info]: 2024-05-01 18:43:57.438--ServerSess
      Average Salary            Team Name
               42400              Arsenal
              322800             Man_city
              304000           Man_united
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- Retrieve name of referee and The Most sum of yellow card which the one referee give for players



Premier League

QUERIES IN JAVA

```
em.getTransaction().begin();
List<Punish> totalcard = em.createNamedQuery("Punish.findAll").getResultList();
System.out.format("%20s %20s\n","Refrree Name" , "Most Yellpw Card" );
for(Punish t : totalcard)
{
   String str = t.getRefrree().getRssn().substring(2,3);
try {
    int yellowcard = Integer.parseInt(t.getYellowCard());
    if(str.equals("0"))
    {
        Anthony_taylor += yellowcard;
        if(Anthony_taylor > max)
        {
            max = Anthony_taylor;
            max_name = "Anthony_taylor";
        }
    }
    else if(str.equals("1"))
    {
        Michael_oliver += yellowcard;
        if(Michael_oliver > max)
        {
            max = Michael_oliver;
            max_name = "Michael_oliver";
        }
    }
    else if(str.equals("2"))
    {
        Craig_bawson += yellowcard;
        if(Craig_bawson > max)
        {
            max = Craig_bawson;
            max_name = "Craig_bawson";
        }
    }
    else if(str.equals("3"))
    {
        Jarred_jilletg += yellowcard;
        if(Jarred_jilletg > max)
        {
            max = Jarred_jilletg;
            max_name = "Jarred_jilletg";
        }
    }
    } catch (NumberFormatException e)
    {
    }
}
System.out.format("%20s %20s\n",max_name , max );
}
```

```
run:
[EL Info]: 2024-05-01 18:57:32.447--ServerSession
        Refrree Name      Most Yellpw Card
      Anthony_taylor                     4
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- list the number of players have the same salary

```
1 •  select Psalary , count(*) as no_of_players
2    from player
3    GROUP BY psalary;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Psalary	no_of_players
1000	1
100000	1
1000000	1
104000	1
115000	1
13000	1
15000	1
150000	1
180000	2
200000	1
225000	1
240000	1
30000	1
375000	1
480000	1
5000	1
50000	1
6000	1
63000	1

Premier League

QUERIES IN JAVA

```java
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        ArrayList<Integer> sal = new ArrayList<>();
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").crea
        em.getTransaction().begin();
        List<Player> play = em.createNamedQuery(string:"Player.findAll").getResultList();
        System.out.format(format:"%20s %20s\n",args: "Salary" , args: "Number of Player" );
        for(Player p : play)
        {
            sal.add(e:p.getPsalary());
            //System.out.format("%20s %20s\n",p.getPsalary() , p );
        }
         HashMap<Integer, Integer> countMap = new HashMap<>();
        for (Integer num : sal)
        {
            countMap.put(key: num, countMap.getOrDefault(key: num, defaultValue:0) + 1);
        }

        for (Integer num : countMap.keySet())
        {
            System.out.format(format:"%20s %20s\n",args: num , args: countMap.get(key: num) );
        }
    }
```

```
run:
[EL Info]: 2024-05-01 21:24:26.739--ServerSess
              Salary     Number of Player
              104000                    1
              100000                    1
              180000                    2
              200000                    1
              240000                    1
              480000                    1
               13000                    1
                1000                    1
                5000                    1
              225000                    1
             1000000                    1
                6000                    1
               50000                    1
               30000                    1
              150000                    1
               63000                    1
               15000                    1
              115000                    1
              375000                    1
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

- list the number of matches of players with team name and nationality

```
Limit to 1000 rows

1  select pname as player, tname as team , Pnationalty as nationality , count(*) as no_of_games
2  from player join player_has_game on pssn = player_pssn
3  join game on gno = game_gno
4  join team on Team_Tname = tname
5  group by pname;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

player	team	nationality	no_of_games
Belal	Scient_soccer	Egyption	6
Ben_white	Arsenal	England	6
Bruno	Man_united	Portuguese	2
Cancelo	Man_city	Portuguese	6
Depruyne	Man_city	Belgium	5
De_gea	Man_united	Spansih	6
Ederson	Man_city	Braziliane	6
Halaand	Man_city	Norway	6
Jesus	Arsenal	Braziliane	4
Maguire	Man_united	England	6
Mo3tesm	Scient_soccer	Egyption	4
Odegard	Arsenal	Norway	3
Omar_elqady	Scient_soccer	Egyption	4
Ramsdale	Arsenal	England	6
Rashford	Man_united	England	4
Ronaldo	Man_united	Portuguese	6
Silva	Man_city	Portuguese	1
Xhaka	Arsenal	Swiss	4
Zenhom	Scient_soccer	Egyption	5

Premier League

QUERIES IN JAVA

```
Source  History
 9    class Project
10
11   blic static void main(String[] args)
12
13       ArrayList<Player> ssn = new ArrayList<>();
14       EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").crea
15       em.getTransaction().begin();
16       List<PlayerHasGame> play = em.createNamedQuery(string: "PlayerHasGame.findAll").getResultLis
17       System.out.format(format: "%20s %20s %20s %15s\n",args:"Player Name" , args:"Team Name", args:
18       for(PlayerHasGame p : play)
19       {
20           if(p.getPlayer() != null)
21           {
22               ssn.add(e:p.getPlayer());
23           }
24
25
26       }
27       HashMap<Player, Integer> countMap = new HashMap<>();
28       for (Player num : ssn)
29       {
30           countMap.put(key:num, countMap.getOrDefault(key:num, defaultValue:0) + 1);
31       }
32
34       for (Player num : countMap.keySet())
35       {
36           System.out.format(format: "%20s %20s %20s %15s\n",
37                   args:num.getPname(), args:num.getTeamTname().getTname(),
38                   args:num.getPnationalty(), args:countMap.get(key:num)
39           );
40       }
```

```
run:
[EL Info]: 2024-05-01 22:29:38.077--ServerSession(375466577)--EclipseLink, version: Ecl
         Player Name            Team Name         Nationality Number of Games
             Ramsdale              Arsenal             England                6
           Ben_white              Arsenal             England                6
             Ederson             Man_city          Braziliane                6
             Odegard              Arsenal              Norway                3
             Cancelo             Man_city          Portuguese                6
              De_gea           Man_united             Spansih                6
               Xhaka              Arsenal               Swiss                4
            Depruyne             Man_city             Belgium                5
             Maguire           Man_united             England                6
          Ahmed_emad         Scient_soccer             Egyption                6
               Jesus              Arsenal          Braziliane                4
               Silva             Man_city          Portuguese                1
               Bruno           Man_united          Portuguese                2
             Mo3tesm         Scient_soccer             Egyption                4
             Halaand             Man_city              Norway                6
            Rashford           Man_united             England                4
               Belal         Scient_soccer             Egyption                6
             Ronaldo           Man_united          Portuguese                6
          Omar_elqady         Scient_soccer             Egyption                4
              Zenhom         Scient_soccer             Egyption                5
BUILD SUCCESSFUL (total time: 1 second)
```

Premier League

QUERIES

5. Some Nested Queries :

- **Retrive Player name , Player salary , Team name , player ssn whose player Score Goal.**

```
1  select psalary , pname, team_Tname , Pssn
2  from player
3  where pssn in
4  (
5  select Player_Pssn
6  from score
7  group by Player_Pssn
8  );
9
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

psalary	pname	team_Tname	Pssn
6000	Ben_white	Arsenal	103
115000	Odegard	Arsenal	104
15000	Xhaka	Arsenal	105
13000	Jesus	Arsenal	106
104000	Depruyne	Man_city	204
1000000	Halaand	Man_city	206
225000	Rashford	Man_united	305
480000	Ronaldo	Man_united	306
5000	Mo3tesm	Scient_soccer	403
50000	Belal	Scient_soccer	404
30000	Omar_elq...	Scient_soccer	405
100000	Zenhom	Scient_soccer	406
NULL	NULL	NULL	NULL

Premier League

QUERIES IN JAVA

5. Some Nested Queries :

```
Source  History
javax.persistence.EntityManager;
javax.persistence.Persistence;

class Project

lic static void main(String[] args)

 ArrayList<Player> ssn = new ArrayList<>();
 EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").create
 em.getTransaction().begin();
 List<Score> play = em.createNamedQuery(string:"Score.findAll").getResultList();
 System.out.format(format:"%20s %20s %20s %15s\n",args: "Salary" , args: "Player Name", args: "Team
 for(Score p : play)
 {
     if(p.getPlayerPssn() != null)
     {
         ssn.add(e: p.getPlayerPssn());
     }
 }
 HashMap<Player, Integer> countMap = new HashMap<>();
 for (Player num : ssn)
 {
     countMap.put(key:num, countMap.getOrDefault(key:num, defaultValue: 0) + 1);
 }

 for (Player num : countMap.keySet())
 {
     System.out.format(format:"%20s %20s %20s %15s\n",
             args: num.getPsalary(), args: num.getPname(),
             args: num.getTeamTname().getTname(), args: num.getPssn()
     );
 }
```

```
run:
[EL Info]: 2024-05-01 22:35:22.456--ServerSession(1095273238)--EclipseLink, ver
              Salary          Player Name             Team Name       Player SSN
                6000           Ben_white               Arsenal             103
              115000             Odegard               Arsenal             104
              104000            Depruyne              Man_city             204
               15000               Xhaka               Arsenal             105
               13000               Jesus               Arsenal             106
                5000            Mo3tesm         Scient_soccer             403
              225000            Rashford            Man_united             305
             1000000             Halaand              Man_city             206
               50000              Belal         Scient_soccer             404
              480000             Ronaldo            Man_united             306
               30000          Omar_elqady         Scient_soccer             405
              100000              Zenhom         Scient_soccer             406
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- - list the oldest club ever established with stadium , president and coach name

```
1  select tname as name , tstart_date as founded_date , sname as stadium , mname as president , cname as coach_name
2  from team join president on mssn = president_mssn
3  join coach on cssn = coche_Cssn
4  join stadium on team_tname = tname
5  where tstart_date=
6  (
7  select min(tstart_date)
8  from team) ;
9
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	founded_date	stadium	president	coach_name
Man_united	1878-03-03	Old_trafford	Joel_glazer	Erik_ten_hag

Premier League

QUERIES IN JAVA

```java
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Persistence;

public class Project
{
    public static void main(String[] args)
    {
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
        em.getTransaction().begin();
        List sdate = em.createNamedQuery(string:"Team.findmindateTeam").getResultList();
        List<Stadium> te = em.createNamedQuery(string:"Stadium.findAll").getResultList();
        System.out.format(format:"%10s %30s %20s %20s %20s\n",
                args:"Team Name" , args:"Start Date", args:"Stadium Name",
                args:"President Name" , args:"Coach Name" );
        for(Stadium p : te)
        {
            if(p.getTeamTname().getTstartdate().equals(obj:sdate.get(index: 0)))
            {
                System.out.format(format:"%10s %30s %20s %20s %20s\n",
                args:p.getTeamTname().getTname() , args:p.getTeamTname().getTstartdate(), args:p.getSname(),
                args:p.getTeamTname().getPresidentMssn().getMname() ,
                args:p.getTeamTname().getCocheCssn().getCname() );
            }
        }
```

Output - project (run)

```
run:
[EL Info]: 2024-05-01 22:53:22.99--ServerSession(2029680286)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
 Team Name                      Start Date         Stadium Name       President Name           Coach Name
Man_united   Sun Mar 03 00:00:00 EET 1878         Old_trafford          Joel_glazer         Erik_ten_hag
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

6. Some Random Queries:

- list the name of players have same salary

```
1 •  select p1.pname , p1.psalary
2    from player p1 join player p2
3    on p1.psalary = p2.psalary and p1.pname <> p2.pname;
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pname	psalary
Cancelo	180000
Ederson	180000

Premier League

QUERIES IN JAVA

6.Some Random Queries in java:

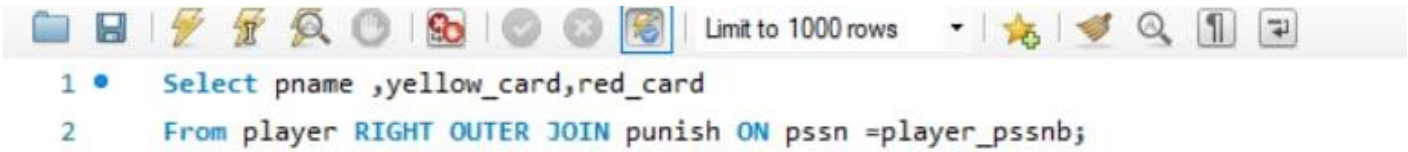
```
Source  History
    public static void main(String[] args)
    { int x = 0;
        ArrayList<Integer> sal = new ArrayList<>();
        EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName: "projectPU").createEntityManager();
        em.getTransaction().begin();
        List<Player> play = em.createNamedQuery(string: "Player.findAll").getResultList();
        System.out.format(format: "%20s %20s\n",args:"Player Name" , args:"Salary" );
        for(Player p : play)
        {

            sal.add(e: p.getPsalary());
        }
         HashMap<Integer, Integer> countMap = new HashMap<>();
        for (Integer num : sal)
        {
            countMap.put(key: num, countMap.getOrDefault(key: num, defaultValue: 0) + 1);
        }

        for (Integer num : countMap.keySet())
        {
            if(countMap.get(key: num) > 1)
            {
                x = num;
                List<Player> plaay = em.createNamedQuery(string: "Player.findByPsalary").setParameter(string: "psalary", o: x).getResultList();
                 for(Player p : plaay)
                {
                    System.out.format(format: "%20s %20s\n",args:p.getPname(), args:p.getPsalary() );
                }
            }
```

```
Output - project (run)
run:
[EL Info]: 2024-05-01 23:42:02.991--ServerSession(787122337)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v20230209-e5c4074ef3
         Player Name               Salary
             Ederson               180000
             Cancelo               180000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

QUERIES

- Retrieve each name of player and his red card and yellow card (if exist)



```
Limit to 1000 rows

1 •  Select pname ,yellow_card,red_card
2    From player RIGHT OUTER JOIN punish ON pssn =player_pssnb;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pname	yellow_card	red_card
Jesus	1	NULL
Halaand	1	NULL
Ronaldo	1	NULL
Ahmed_emad	1	NULL
Maguire	1	NULL
Ahmed_emad	1	NULL
Maguire	1	NULL
Belal	1	NULL
Zenhom	NULL	1
Cancelo	1	NULL
Omar_elqady	1	NULL
Maguire	1	NULL
Halaand	1	NULL

Premier League

QUERIES IN JAVA

Source History

```
4  import javax.persistence.EntityManager;
5  import javax.persistence.Persistence;
6
7  public class Project
8  {
9      public static void main(String[] args)
10     {
11         EntityManager em = Persistence.createEntityManagerFactory(persistenceUnitName:"projectPU").createEntityManager();
12         em.getTransaction().begin();
13         List<Punish> punh = em.createNamedQuery(string:"Punish.findAll").getResultList();
14         System.out.format(format:"%20s %10s %10s\n",
15                 args: "Player Name" , args: "Yellow Card", args: "Red Card");
16         for(Punish p : punh)
17         {
18             System.out.format(format:"%20s %10s %10s\n",
19                 args: p.getPlayer().getPname() , args: p.getYellowCard(), args: p.getRedcard());
20         }
```

Output - project (run)

```
run:
[EL Info]: 2024-05-01 23:00:33.9--ServerSession(36657658)--EclipseLink, version: Eclipse Persistence Services - 2.7.12.v2023020
         Player Name Yellow Card   Red Card
               Jesus           1          0
             Halaand           1          0
             Ronaldo           1          0
           Ahmed_emad          1          0
              Maguire          1          0
           Ahmed_emad          1          0
              Maguire          1          0
                Belal          1          0
               Zenhom          0          1
             Cancelo           1          0
          Omar_elqady          1          0
              Maguire          1          0
              Halaand          1          0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

NOW WE REPRESENT SOME TRIGGERS

TRIGGER

1 . NOW SOME TRIGGER IN PLAYER TABLE :

THE UPDATE AFTER TRIGGER IN PLAYER TABLE

```
create table Playeraudit (
    ID int auto_increment primary key,
    pname varchar(15) not null,
    pno varchar(3) not null,
    psalary int not null,
    changedate datetime default null,
    action varchar(50) default null
);

DELIMITER //
CREATE TRIGGER After_Player_Update
AFTER Update ON player
FOR EACH ROW
BEGIN
    INSERT INTO playeraudit
    Set pname = NEW.Pname,
        pno = NEW.Pno,
        psalary =NEW.Psalary,
        changedate = NOW(),
        action = 'UPDATE';
END //
DELIMITER ;
```

Output

Action Output

#	Time	Action	Message
7	00:29:37	SELECT * FROM premier_league.teamaudit LIMIT 0, 1000	3 row(s) returned
8	00:33:26	create table Playeraudit (ID int auto_increment primary key, pname varchar(15) not null, pno varch...	Error Code: 1064. You have an error in your SQL syntax; check
9	00:33:39	create table Playeraudit (ID int auto_increment primary key, pname varchar(15) not null, pno varch...	0 row(s) affected
10	00:37:33	CREATE TRIGGER After_Player_Update AFTER Update ON player FOR EACH ROW BEGIN INSER...	0 row(s) affected

Premier League

TRIGGER

- THAT'S THE INSERT AFTER TRIGGER IN PLAYER TABLE

```
1   DELIMITER //
2   CREATE TRIGGER After_Player_Insert
3   After insert ON player
4   FOR EACH ROW
5   BEGIN
6       INSERT INTO playeraudit (pname, pno, psalary, changedate, action)
7           value (NEW.Pname, NEW.Pno, NEW.Psalary, NOW(), 'Insert');
8   END //
9   DELIMITER ;
10
```

Output

Action Output

#	Time	Action	Message
14	00:41:37	CREATE TRIGGER Before_Player_Insert before insert ON player FOR EACH ROW BEGIN INSERT I...	Error Code: 1363. There is no OLD row in on INSERT trigger
15	00:42:10	CREATE TRIGGER After_Player_Insert After insert ON player FOR EACH ROW BEGIN INSERT INTO...	0 row(s) affected
16	00:42:20	SELECT * FROM premier_league.player LIMIT 0, 1000	20 row(s) returned
17	00:43:02	SELECT * FROM premier_league.playeraudit LIMIT 0, 1000	3 row(s) returned

Premier League

TRIGGER

- THAT'S THE DELETE BEFORE TRIGGER IN PLAYER TABLE

```
1   DELIMITER //
2   CREATE TRIGGER Before_Player_Delete
3   before delete ON player
4   FOR EACH ROW
5   BEGIN
6       INSERT INTO playeraudit (pname, pno, psalary, changedate, action)
7           value (OLD.Pname, OLD.Pno, OLD.Psalary, NOW(), 'Delete');
8   END //
9   DELIMITER ;
10
```

Output

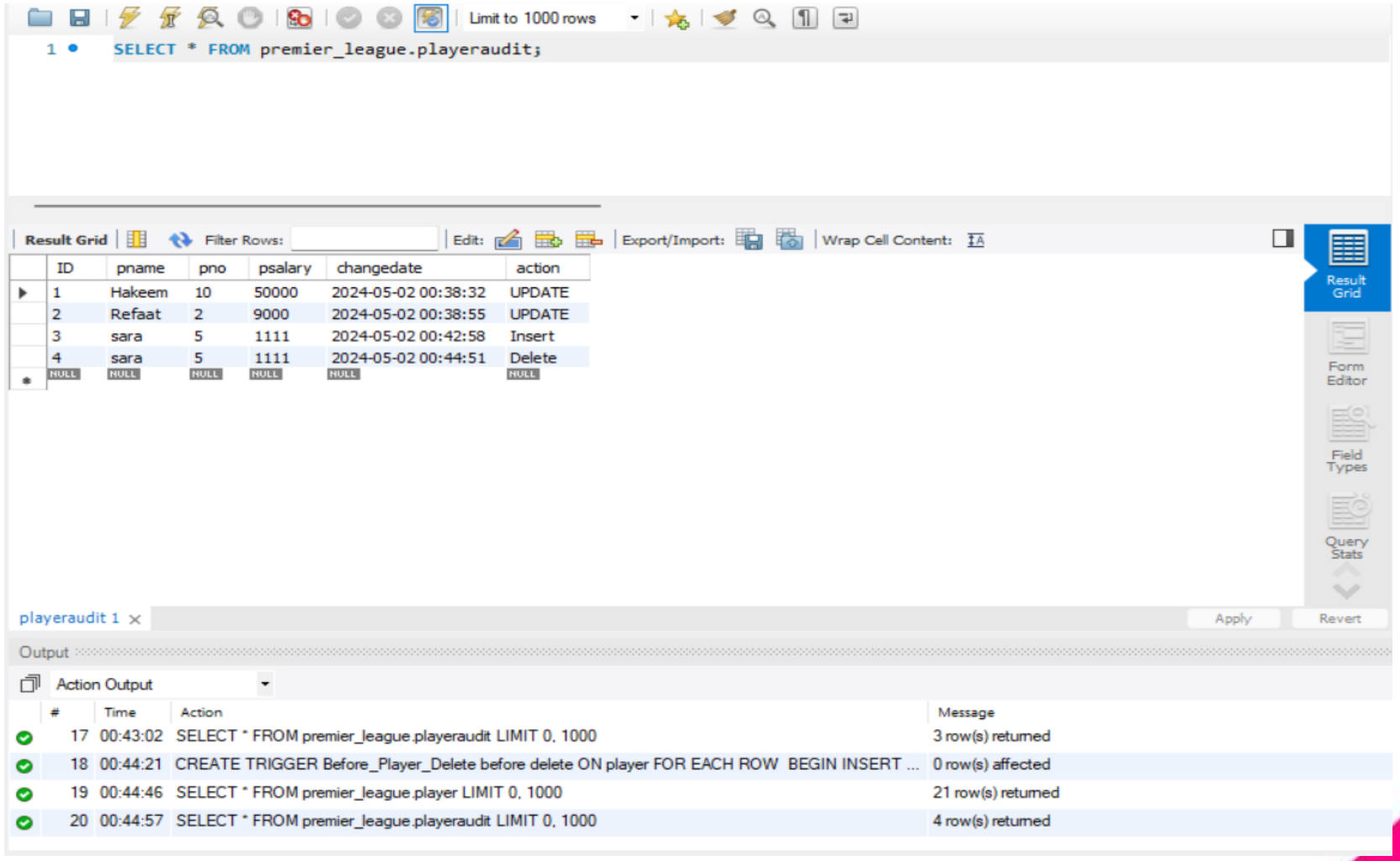
Action Output

#	Time	Action	Message
15	00:42:10	CREATE TRIGGER After_Player_Insert After insert ON player FOR EACH ROW BEGIN INSERT INTO...	0 row(s) affected
16	00:42:20	SELECT * FROM premier_league.player LIMIT 0, 1000	20 row(s) returned
17	00:43:02	SELECT * FROM premier_league.playeraudit LIMIT 0, 1000	3 row(s) returned
18	00:44:21	CREATE TRIGGER Before_Player_Delete before delete ON player FOR EACH ROW BEGIN INSERT ...	0 row(s) affected

Premier League

TRIGGER

- THAT'S THE RUN OF TRIGGER IN PLAYER TABLE



Premier League

TRIGGER

2 . NOW SOME TRIGGER IN TEAM TABLE :

THE UPDATE BEFORE TRIGGER IN TEAM TABLE

```
create Table TeamAudit (
    ID int auto_increment primary key,
    Teamname int not null,
    ThomeKit varchar(10) not null,
    Tcaptin varchar(15) not null,
    changedate datetime default null,
    action varchar(50)  default null
);

DELIMITER $$
create Trigger Before_Team_Update
    Before Update on team
    FOR EACH ROW
BEGIN
    insert into teamaudit
    set action = 'update',
    Teamname = OLD.Tname,
    ThomeKit = OLD.Thome_kit,
    Tcaptin = OLD.Tcaptain,
    changedate = NOW();
END $$
DELIMITER ;
```

Output

Action Output

#	Time	Action	Message
11	00:09:46	SELECT * FROM premier_league.teamaudit LIMIT 0, 1000	0 row(s) returned
12	00:10:05	SELECT * FROM premier_league.team LIMIT 0, 1000	4 row(s) returned
13	00:10:24	SELECT * FROM premier_league.team LIMIT 0, 1000	4 row(s) returned
14	00:10:58	SELECT * FROM premier_league.teamaudit LIMIT 0, 1000	1 row(s) returned

Premier League

TRIGGER

- THAT'S THE INSERT AFTER TRIGGER IN TEAM TABLE

```
1  DELIMITER //
2  CREATE TRIGGER After_Team_Insert
3  AFTER INSERT ON team
4  FOR EACH ROW
5  BEGIN
6      INSERT INTO teamaudit (Teamname, ThomeKit, Tcaptin, changedate, action)
7      VALUE (NEW.Tname, NEW.Thome_kit, NEW.Tcaptain, NOW(), 'Insert');
8  END //
9  DELIMITER ;
```

Output

Action Output

#	Time	Action	Message
1	00:23:59	SELECT * FROM premier_league.teamaudit LIMIT 0, 1000	2 row(s) returned
2	00:27:23	CREATE TRIGGER After_Team_Insert AFTER INSERT ON team FOR EACH ROW BEGIN INSERT IN...	Error Code: 1359. Trigger 'premier_league.After_Team_Insert' a

Premier League

TRIGGER

- THAT'S THE DELETE AFTER TRIGGER IN TEAM TABLE

Query 1 | team | teamaudit

Limit to 1000 rows

```
1  DELIMITER //
2  CREATE TRIGGER After_Team_Delete
3  AFTER Delete ON team
4  FOR EACH ROW
5  BEGIN
6      INSERT INTO teamaudit (Teamname, ThomeKit, Tcaptin, changedate, action)
7      VALUE (OLD.Tname, OLD.Thome_kit, OLD.Tcaptain, NOW(), 'Delete');
8  END //
9  DELIMITER ;
```

Output

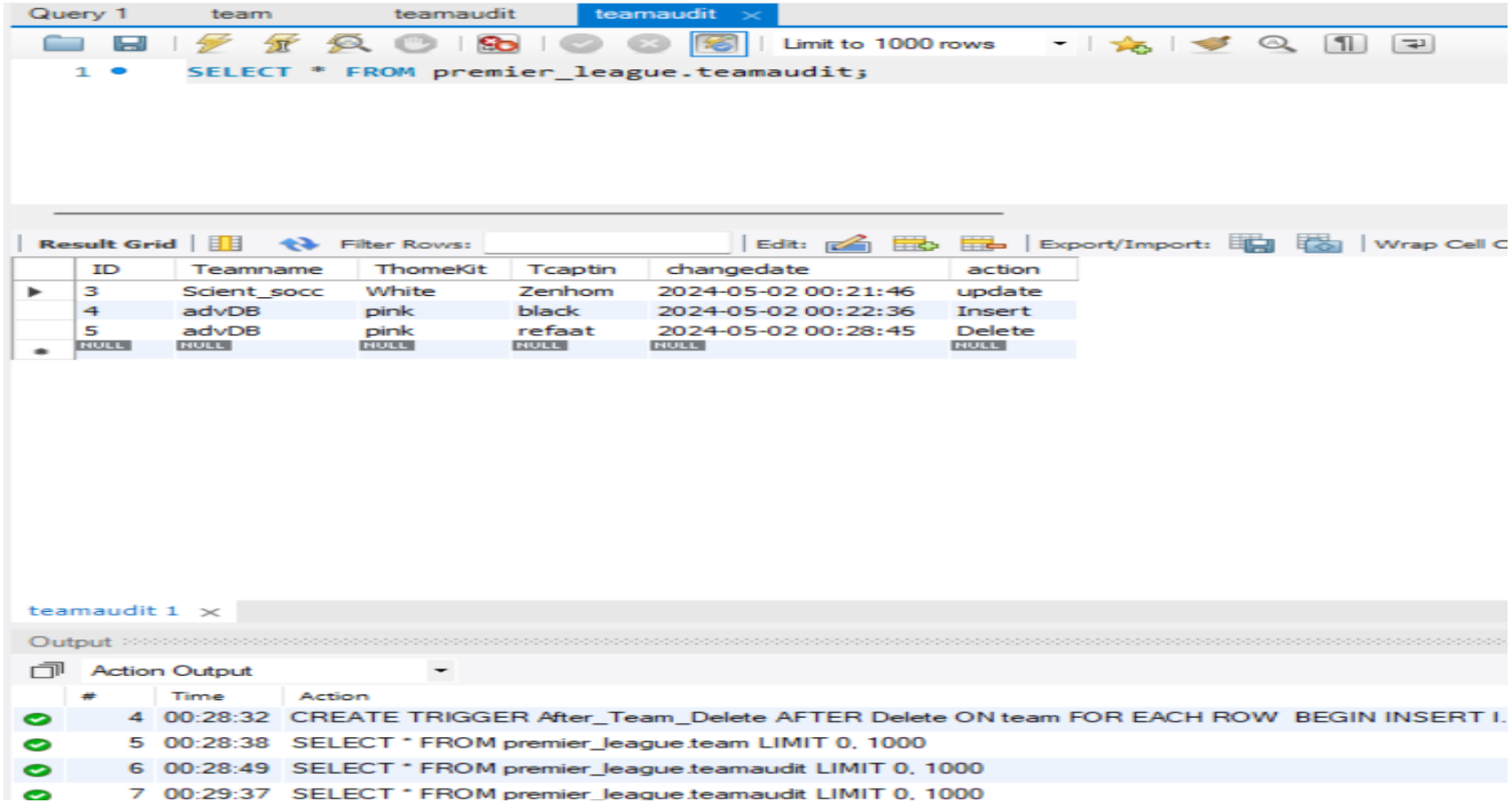
Action Output

#	Time	Action	Message
3	00:28:25	CREATE TRIGGER After_Team_Delete AFTER Delete ON team FOR EACH ROW BEGIN INSERT I...	Error Code: 1054. Unknown column 'Tcaptainx' in 'OLD'
4	00:28:32	CREATE TRIGGER After_Team_Delete AFTER Delete ON team FOR EACH ROW BEGIN INSERT I...	0 row(s) affected
5	00:28:38	SELECT * FROM premier_league.team LIMIT 0, 1000	5 row(s) returned
6	00:28:49	SELECT * FROM premier_league.teamaudit LIMIT 0, 1000	3 row(s) returned

Premier League

TRIGGER

- THAT'S THE RUN OF TRIGGER IN TEAM TABLE



Premier League

NOW WE REPRESENT SOME PROCEDURES

PROCEDURES

1 .PROCEDURES :

Procedure to retrieve all player in each position

Query 1

Limit to 1000 rows

```
1   DELIMITER $$
2   Drop procedure if exists get_number_by_position$$
3   CREATE procedure get_number_by_position(OUT GK INT, OUT CB INT, OUT LB INT, OUT CM INT, OUT CAM INT, OUT CDM INT, OUT LM INT, OUT ST INT)
4   begin
5       SELECT count(*) into GK FROM player where Position = 'GK';
6       SELECT count(*) into CB FROM player where Position = 'CB';
7       SELECT count(*) into LB FROM player where Position = 'LB';
8       SELECT count(*) into CM FROM player where Position = 'CM';
9       SELECT count(*) into CAM FROM player where Position = 'CAM';
10      SELECT count(*) into CDM FROM player where Position = 'CDM';
11      SELECT count(*) into LM FROM player where Position = 'LM';
12      SELECT count(*) into ST FROM player where Position = 'ST';
13  End$$
14  DELIMITER ;
15
16  call get_number_by_position(@GK, @CB, @LB, @CM, @CAM, @CDM, @LM, @ST);
17  select @GK, @CB, @LB, @CM, @CAM, @CDM, @LM, @ST;
18
19
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

@GK	@CB	@LB	@CM	@CAM	@CDM	@LM	@ST
4	3	1	1	5	1	1	4

Premier League

PROCEDURES IN JAVA

- THAT'S THE RUN OF PROCEDURE IN JAVA

Start Page | DB.java

Source | History

```
21          try {
22              Class.forName(className:myDriver);
                Connection conn = DriverManager.getConnection(url: myUrl, user:userName, password);
24              String procedureCall = "{call get_number_by_position(?, ?, ?, ?, ?, ?, ?, ?)}";
25              CallableStatement callableStatement = conn.prepareCall(string: procedureCall);
26              callableStatement.registerOutParameter(i: 1, il: Types.INTEGER);
27              callableStatement.registerOutParameter(i: 2, il: Types.INTEGER);
28              callableStatement.registerOutParameter(i: 3, il: Types.INTEGER);
29              callableStatement.registerOutParameter(i: 4, il: Types.INTEGER);
30              callableStatement.registerOutParameter(i: 5, il: Types.INTEGER);
31              callableStatement.registerOutParameter(i: 6, il: Types.INTEGER);
32              callableStatement.registerOutParameter(i: 7, il: Types.INTEGER);
33              callableStatement.registerOutParameter(i: 8, il: Types.INTEGER);
34
35              callableStatement.execute();
36
37              int GK = callableStatement.getInt(i: 1);
38              int CB = callableStatement.getInt(i: 2);
                int LB = callableStatement.getInt(i: 3);
40              int CM = callableStatement.getInt(i: 4);
41              int CAM = callableStatement.getInt(i: 5);
42              int CDM = callableStatement.getInt(i: 6);
43              int LM = callableStatement.getInt(i: 7);
44              int ST = callableStatement.getInt(i: 8);
45
46              System.out.println("Number of GK: " + GK);
47              System.out.println("Number of CB: " + CB);
48              System.out.println("Number of LB: " + LB);
49              System.out.println("Number of CM: " + CM);
50              System.out.println("Number of CAM: " + CAM);
51              System.out.println("Number of CDM: " + CDM);
52              System.out.println("Number of LM: " + LM);
53              System.out.println("Number of ST: " + ST);
54
55              callableStatement.close();
56              conn.close();
57          } catch (ClassNotFoundException ex) {
58              Logger.getLogger(name:DB.class.getName()).log(level: Level.SEVERE, msg: null, thrown: e
```

Output - DB (run)

```
run:
Number of GK: 4
Number of CB: 3
Number of LB: 1
Number of CM: 1
Number of CAM: 5
Number of CDM: 1
Number of LM: 1
Number of ST: 4
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

FUNCTIONS

2 . FUNCTION :

FUNCTION TO GET THE KIND OF THE PLAYER

Query 1 | player | teamaudit | playeraudit | player | playeraudit | player | playeraudit

Limit to 1000 rows

```
1   DELIMITER $$
2   CREATE FUNCTION playerKind(salary INT)
3   RETURNS varchar(10)
4   DETERMINISTIC
5   BEGIN
6       DECLARE player_Level varchar(10);
7
8       IF salary < 60000 THEN
9           SET player_Level = 'Bronz';
10      ELSEIF salary >= 60000 AND salary < 100000 THEN
11          SET player_Level = 'Silver';
12      ELSEIF salary >= 100000 THEN
13          SET player_Level = 'Gold';
14      END IF;
15
16      RETURN player_Level;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Pname	playerKind(player.Psalary)
Ramsdale	Silver
Ben_white	Bronz
Odegard	Gold
Xhaka	Bronz
Jesus	Bronz
Ederson	Gold
Cancelo	Gold
Depruyne	Gold
Silva	Gold
Halaand	Gold
De_gea	Gold

Premier League

PROCEDURES

3 . PROCEDURES WITH FUNCTION:

Procedure to retrieve player with gold kind

Query 1 | player | teamaudit | playeraudit | player | playeraudit | player | playeraudit

Limit to 1000 rows

```
1   DELIMITER $$
2
3 • CREATE procedure get_player_Gold()
4   BEGIN
5       select player.Pname, player.Psalary, playerKind(player.Psalary)
6       FROM player
7       where playerKind(player.Psalary) = 'GOLD';
8   END $$
9
10  DELIMITER ;
11
12 • Call get_player_Gold();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Pname	Psalary	playerKind(player.Psalary)
Ederson	180000	Gold
Cancelo	180000	Gold
Depruyne	104000	Gold
Silva	150000	Gold
Halaand	1000000	Gold
De_gea	375000	Gold
Maguire	200000	Gold
Bruno	240000	Gold
Rashford	225000	Gold
Ronaldo	480000	Gold
Zenhom	100000	Gold

Premier League

PROCEDURES AND FUNCTION IN JAVA

- THAT'S THE RUN OF PROCEDURE AND FUNCTION IN JAVA

```
Source  History
16      {
17          String myDriver = "com.mysql.cj.jdbc.Driver";
18          String myUrl = "jdbc:mysql://localhost:3306/premier_league";
19          String userName = "root";
20          String password = "";
21          try {
22              Class.forName(className:myDriver);
23              Connection conn = DriverManager.getConnection(url: myUrl, user:userName, password);
24              System.out.println(x: "Retrive Gold Player Using Procedure");
25              System.out.format(format: "%25s %25s %25s\n",args:"Player Name",args:"PLayer Salary",args:"Play
26              String Procedurecall = "{call get_player_Gold()}";
                CallableStatement Stmt = conn.prepareCall(string: Procedurecall);
28              Stmt.execute();
29              ResultSet re = Stmt.getResultSet();
30              while(re.next())
31              {
32                  String name = re.getString(string: "Pname");
33                  int number = re.getInt(string: "Psalary");
                    String Kind = re.getString(string: "playerKind(player.Psalary)");
35                  System.out.format(format: "%25s %25s %25s\n",args:name,args:number,args:Kind);
36              }
37              Stmt.close();
38          } catch (ClassNotFoundException ex) {
39              Logger.getLogger(name:DB.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
40          } catch (SQLException ex) {
41              Logger.getLogger(name:DB.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
42          }
43      }
44  }
45
```

```
run:
Retrive Gold Player Using Procedure
              Player Name              PLayer Salary              Player Kind
                  Odegard                     115000                     Gold
                  Ederson                     180000                     Gold
                  Cancelo                     180000                     Gold
                 Depruyne                     104000                     Gold
                    Silva                     150000                     Gold
                  Halaand                    1000000                     Gold
                   De_gea                     375000                     Gold
                  Maguire                     200000                     Gold
                    Bruno                     240000                     Gold
                 Rashford                     225000                     Gold
                  Ronaldo                     480000                     Gold
                   Zenhom                     100000                     Gold
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

PROCEDURES AND CURSOR

4 .PROCEDURES WITH CURSOR:

Procedure to retrieve the ssn of all player splited by ;

Query 1

Limit to 1000 rows

```
2  CREATE procedure Build_SsnPlayer_List(INOUT SSN_LIST varchar(3000))
3  begin
4      declare v_finished INTEGER default 0;
5      declare v_SsnPlayer varchar(10) default "";
6
7      declare SsnPlayer_cursor cursor for
8          select player.Pssn from player;
9      declare continue handler for
10         not found SET v_finished = 1;
11
12     open SsnPlayer_cursor;
13     get_SsnPlayer: LOOP
14         fetch SsnPlayer_cursor into v_SsnPlayer;
15         if v_finished = 1 then
16             leave get_SsnPlayer;
17         end if;
18
19         set SSN_LIST = concat(v_SsnPlayer, ";" , SSN_LIST);
20
21     END LOOP get_SsnPlayer;
22
23     close SsnPlayer_cursor;
24 End$$
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

@SsnPlayer_list
406;405;404;403;402;306;305;304;303;302;206;205;204;203;202;106;105;104;103;102;

PROCEDURES AND CURSOR IN JAVA

- THAT'S THE RUN OF PROCEDURE AND CURSOR IN JAVA

```
Source  History
16      {
17          String myDriver = "com.mysql.cj.jdbc.Driver";
18          String myUrl = "jdbc:mysql://localhost:3306/premier_league";
19          String userName = "root";
20          String password = "";
21          try {
22              Class.forName(className: myDriver);
            Connection conn = DriverManager.getConnection(url: myUrl, user: userName, password);
24              System.out.println(x: "Retrive All SSN Player with , Using Procedure Cursor");
25              String Procedurecall = "{call Build_SsnPlayer_List(?)}";
26              CallableStatement callableStatement  = conn.prepareCall(string: Procedurecall);
27              callableStatement.setString(i: 1, string: "");
28              callableStatement.registerOutParameter(i: 1, i1: java.sql.Types.VARCHAR);
29              callableStatement.execute();
30              String ssnList = callableStatement.getString(i: 1);
31              System.out.println("SSN Player List ==> " + ssnList);
32              callableStatement.close();
33              conn.close();
            } catch (ClassNotFoundException ex) {
35              Logger.getLogger(name: DB.class.getName()).log(level:Level.SEVERE, msg: null, thrown: ex);
36          } catch (SQLException ex) {
37              Logger.getLogger(name: DB.class.getName()).log(level:Level.SEVERE, msg: null, thrown: ex);
38          }
```

```
Output - DB (run)
run:
Retrive All SSN Player with , Using Procedure Cursor
SSN Player List ==> 406;405;404;403;402;306;305;304;303;302;206;205;204;203;202;106;105;104;103;102;
BUILD SUCCESSFUL (total time: 0 seconds)
```

Premier League

Thank You

MMASH