FOOTBALL ANALYTICS

By Sriram Chandramouli

CONTENTS

Sriram Chandramouli

Advanced Database Management Systems

CASE SETUP:

Football is one of the most popular games played throughout the world. The viewership and the demand is increasing rapidly by the day and maintaining the information in a highly secured environment is one of the primary needs for any league conducting the events. In order to achieve this and to amplify the performance I am implementing Football Analytics that aims to:

1. Build a robust system that can store and retrieve data effectively and conveniently.
2. Create “constraints” to minimize data duplications.
3. Provide improved data security.

Football Analytics focuses on developing a database that maintains various events and information revolving around Football maintenance such as player information, event details etc. Though there are many databases that maintain this information in a secured level, my database deals with an effective/normalized way of storing and retrieving the data. Maintaining the records of various actors in Football is a cumbersome process and the presence of ambiguity and redundancy in the data further aggravates the difficulty. Hence, automating the entire system by building a robust database, makes data maintenance a less difficult job. The main users for accessing my database system will be Football players, supporting staff of the team. Apart from that, registered users will also be using the system in order to learn about various details of the players, events and other information involved in Football.

The main objective is to develop a database that would maintain records of various components related to Football. It is only logical if system is defined in terms of relationships rather than a mere representation of the components. The various components would include club information, club details, and player information among others. Each component has a distinct functionality and is inter-related. So, having a system that supports concurrent updating and minimizes duplicity is absolutely essential. Moreover, since, the league has planned to expand its base to areas that have seen a recent spurt in Football subscriptions. To accommodate the throngs of audience and supporters, the system has to have a robust system. My database holds top-level data of the football matches such as match date, competition name, participating teams, venues and environmental conditions. It also captures all macro-events occurring during the match such as goals, penalties, and disciplinary incidents. The biographical and physical data of players, clubs, club managers and supporting staff are also available in the database.

Finally, we intend to improve upon the proposed model as Football Analytics system is more evolutionary than revolutionary.

BUSINESS AND DATA REQUIREMENT:

The most essential requirement for maintaining the database is to have all the information about various events in one common place. Football has many leagues conducted throughout the world and it is ideal if all this information can be stored and retrieved in a single database. This prevents Football Franchisees to visit multiple systems resulting in time saving and as well as saving cost.

The system covers two leagues and each league has four teams. There are 5 players per team and the database was developed adhering to the common norms and establishing atomicity while meeting the business requirements. The person table holds the information of various players involved in the league, supporting staff, owner details and common users. The user category includes managers, players, club chairman, club secretary, club doctor, and common user. This table basically holds the information about all the active persons involved in accessing the Football database. This table captures all demographic information such as name, age, nationality etc.

The league table holds the information about various leagues whose information needs to be stored. The club directory table holds the information about various clubs and their history. The club table has various information such as its position in the league, number of games played/won/drawn/lost and its points for a particular league season. The most important table of all is the Squad, which holds the information of various players of a particular club. This table holds the information of player's position in the field, whether forward, defense, midfield or attacking, goals scored, matches played, cards received for foul play, and wages of individual players. This information enables managers and various other clubs in the world to look into player's profile and cast their interest for buying new player. They can look into each and every activity of the players, track their information match by match and even know about the past clubs where they played. As we all know, players form the basic foundation of Football and this system enables various users to knowing the information about players at the touch of single button.

There are other tables such as attacking, defending and goals which captures various information about the clubs such as total number of shots passed, number of crosses made, offside, goals scored, goals conceded etc. This information helps various teams to plan and build a strategy when they face the opponent.

Other important tables are Fixtures and View Results where all the details about the matches played in the past and matches in future will be available. These are helpful to gather the information about the past matches for common users and as well as other actors accessing the database. The Franchise table holds the information about sponsors name, their budget and contract year with a particular club. This enables rest of the franchises to bid for sponsorship for any club in future.

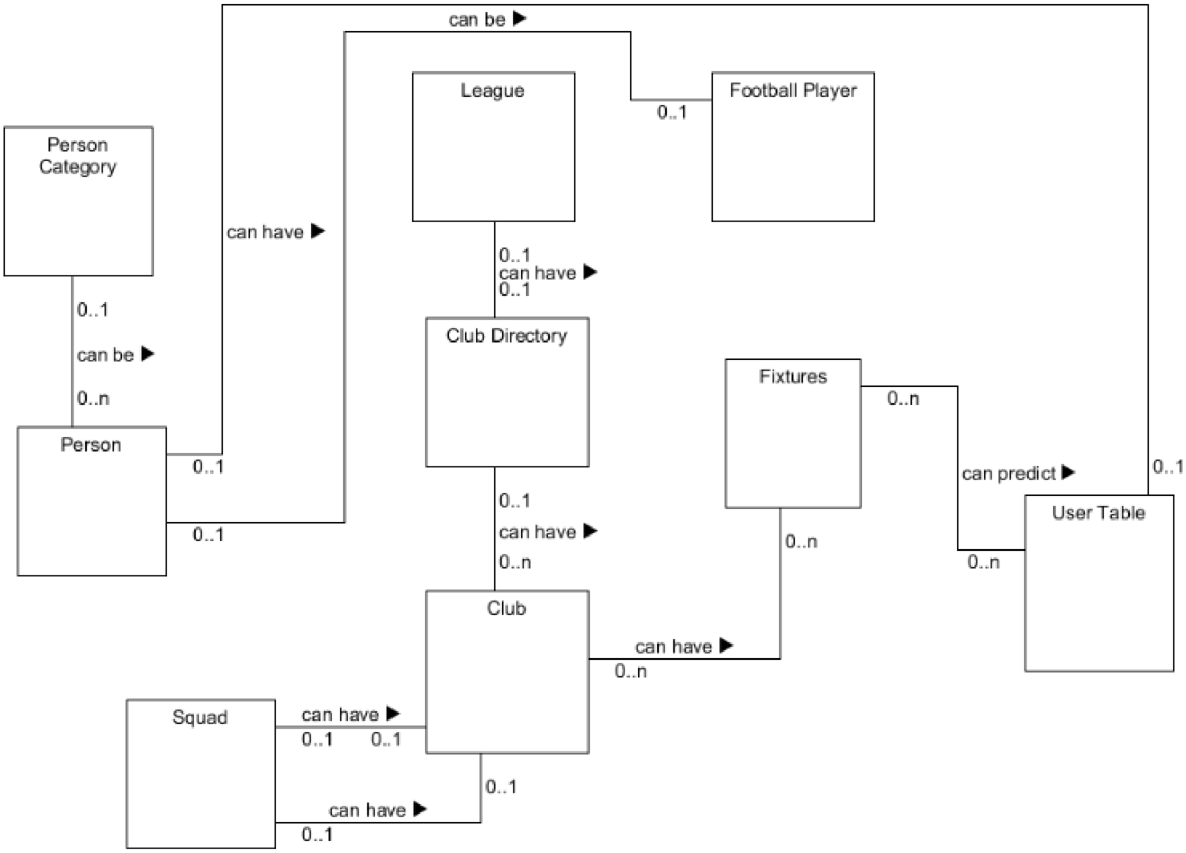
There is a unique table for users to attract their attention. The users can forecast the winner of a match by predicting the goals scored in a match.

All table covers both super type and sub type. Tables like Club directory, Fixtures, League, and Person Category are super types. Tables such as club, squad are sub types that are dependent on super type tables and the link between these two tables are created using foreign keys.

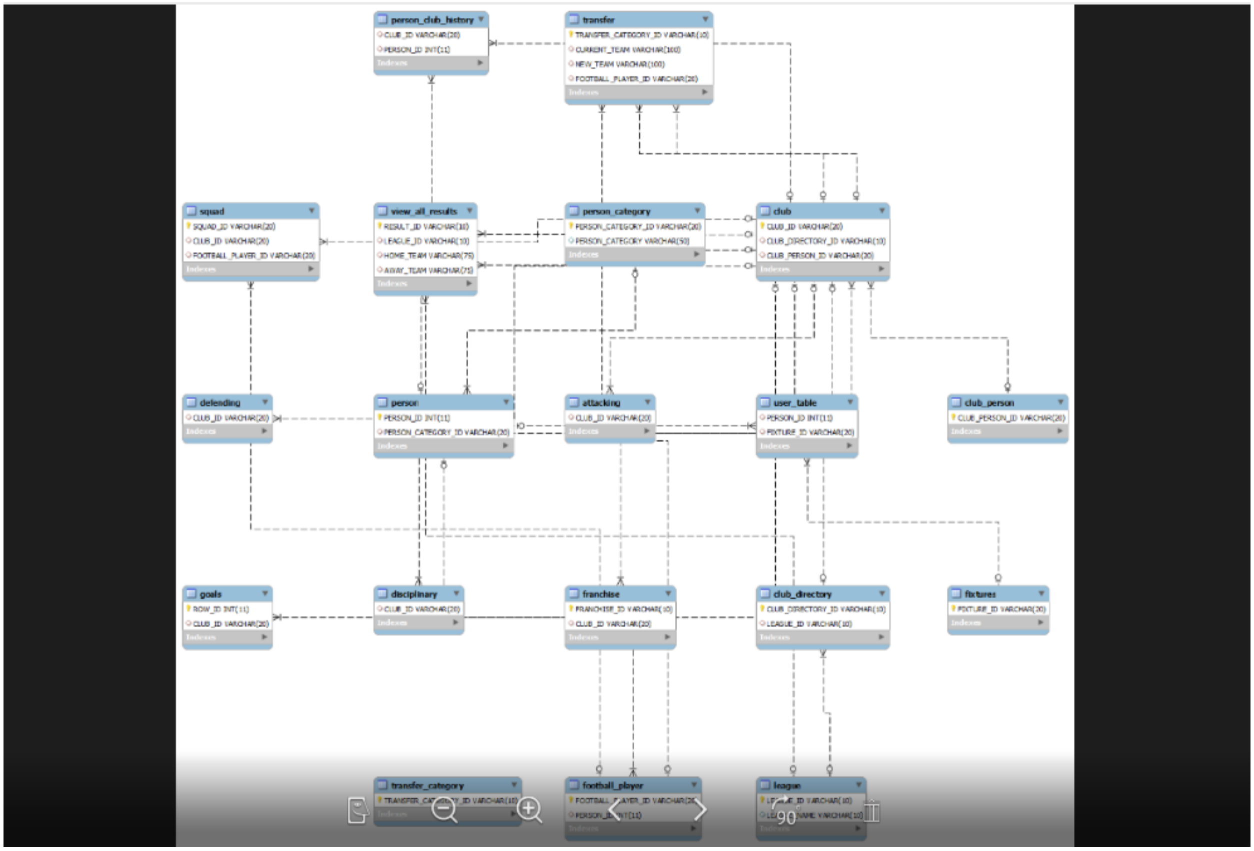
There are various roles created in database and these roles have several privileges set to access the database. All the tables are normalized making sure the data redundancy, data complexity are reduced and faster retrieval of data can be achieved. Example, Player information are stored in a separate table named Football_Player. The Squad table has the primary key of Football_Player set as foreign key in it, so that the player information can be obtained by joining two table. The player may be changing the club from club to club and it would be ideal if their data are maintained in separate table. Similar pattern is followed for the creation of other tables in order to ensure that the tables are normalized.

DATABASE DIAGRAMS:

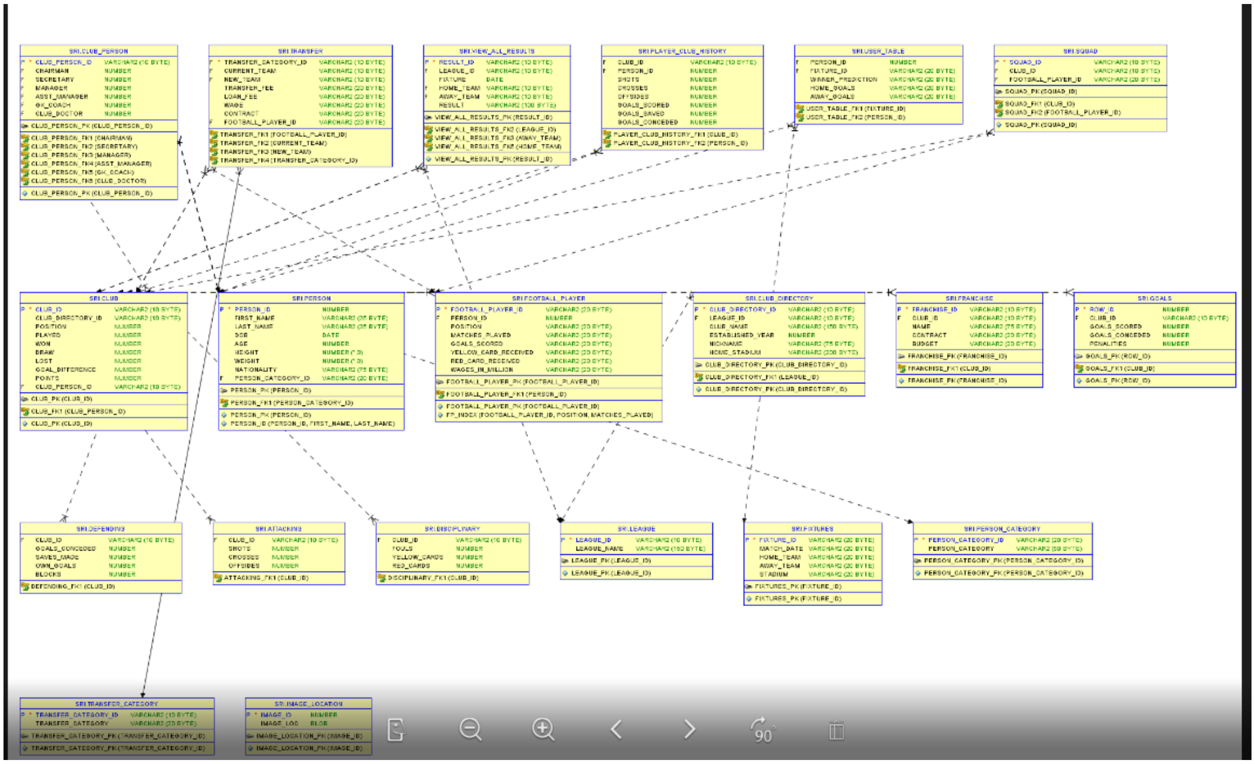
CONCEPTUAL MODEL:



LOGICAL MODEL:



PHYSICAL MODEL:


SRI.CLUB_PERSON
P * CLUB_PERSON_ID VARCHAR2 (10 BYTE)
F CHAIRMAN NUMBER
F SECRETARY NUMBER
F MANAGER NUMBER
F ASST_MANAGER NUMBER
F GK_COACH NUMBER
F CLUB_DOCTOR NUMBER
CLUB_PERSON_PK (CLUB_PERSON_ID)
CLUB_PERSON_FK1 (CHAIRMAN)
CLUB_PERSON_FK2 (SECRETARY)
CLUB_PERSON_FK3 (MANAGER)
CLUB_PERSON_FK4 (ASST_MANAGER)
CLUB_PERSON_FK5 (GK_COACH)
CLUB_PERSON_FK6 (CLUB_DOCTOR)
CLUB_PERSON_PK (CLUB_PERSON_ID)
SRI.TRANSFER
P * TRANSFER_CATEGORY_ID VARCHAR2 (10 BYTE)
F CURRENT_TEAM VARCHAR2 (10 BYTE)
F NEW_TEAM VARCHAR2 (10 BYTE)
TRANSFER_FEE VARCHAR2 (20 BYTE)
LOAN_FEE VARCHAR2 (20 BYTE)
WAGE VARCHAR2 (20 BYTE)
CONTRACT VARCHAR2 (20 BYTE)
F FOOTBALL_PLAYER_ID VARCHAR2 (20 BYTE)
TRANSFER_FK1 (FOOTBALL_PLAYER_ID)
TRANSFER_FK2 (CURRENT_TEAM)
TRANSFER_FK3 (NEW_TEAM)
TRANSFER_FK4 (TRANSFER_CATEGORY_ID)
SRI.VIEW_ALL_RESULTS
P * RESULT_ID VARCHAR2 (10 BYTE)
F LEAGUE_ID VARCHAR2 (10 BYTE)
FIXTURE DATE
F HOME_TEAM VARCHAR2 (10 BYTE)
F AWAY_TEAM VARCHAR2 (10 BYTE)
RESULT VARCHAR2 (100 BYTE)
VIEW_ALL_RESULTS_PK (RESULT_ID)
VIEW_ALL_RESULTS_FK2 (LEAGUE_ID)
VIEW_ALL_RESULTS_FK3 (AWAY_TEAM)
VIEW_ALL_RESULTS_FK6 (HOME_TEAM)
VIEW_ALL_RESULTS_PK (RESULT_ID)
SRI.PLAYER_CLUB_HISTORY
F CLUB_ID VARCHAR2 (10 BYTE)
F PERSON_ID NUMBER
SHOTS NUMBER
CROSSES NUMBER
OFFSIDES NUMBER
GOALS_SCORED NUMBER
GOALS_SAVED NUMBER
GOALS_CONCEDED NUMBER
PLAYER_CLUB_HISTORY_FK1 (CLUB_ID)
PLAYER_CLUB_HISTORY_FK2 (PERSON_ID)
SRI.USER_TABLE
F PERSON_ID NUMBER
F FIXTURE_ID VARCHAR2 (20 BYTE)
WINNER_PREDICTION VARCHAR2 (20 BYTE)
HOME_GOALS VARCHAR2 (20 BYTE)
AWAY_GOALS VARCHAR2 (20 BYTE)
USER_TABLE_FK1 (FIXTURE_ID)
USER_TABLE_FK2 (PERSON_ID)
SRI.SQUAD
P * SQUAD_ID VARCHAR2 (10 BYTE)
F CLUB_ID VARCHAR2 (10 BYTE)
F FOOTBALL_PLAYER_ID VARCHAR2 (20 BYTE)
SQUAD_PK (SQUAD_ID)
SQUAD_FK1 (CLUB_ID)
SQUAD_FK2 (FOOTBALL_PLAYER_ID)
SQUAD_PK (SQUAD_ID)
SRI.CLUB
P * CLUB_ID VARCHAR2 (10 BYTE)
CLUB_DIRECTORY_ID VARCHAR2 (10 BYTE)
POSITION NUMBER
PLAYED NUMBER
WON NUMBER
DRAW NUMBER
LOST NUMBER
GOAL_DIFFERENCE NUMBER
POINTS NUMBER
F CLUB_PERSON_ID VARCHAR2 (10 BYTE)
CLUB_PK (CLUB_ID)
CLUB_FK1 (CLUB_PERSON_ID)
CLUB_PK (CLUB_ID)
SRI.PERSON
P * PERSON_ID NUMBER
FIRST_NAME VARCHAR2 (25 BYTE)
LAST_NAME VARCHAR2 (25 BYTE)
DOB DATE
AGE NUMBER
HEIGHT NUMBER (*,0)
WEIGHT NUMBER (*,0)
NATIONALITY VARCHAR2 (75 BYTE)
F PERSON_CATEGORY_ID VARCHAR2 (20 BYTE)
PERSON_PK (PERSON_ID)
PERSON_FK1 (PERSON_CATEGORY_ID)
PERSON_PK (PERSON_ID)
PERSON_ID (PERSON_ID, FIRST_NAME, LAST_NAME)
SRI.FOOTBALL_PLAYER
P * FOOTBALL_PLAYER_ID VARCHAR2 (20 BYTE)
F PERSON_ID NUMBER
POSITION VARCHAR2 (20 BYTE)
MATCHES_PLAYED VARCHAR2 (20 BYTE)
GOALS_SCORED VARCHAR2 (20 BYTE)
YELLOW_CARD_RECEIVED VARCHAR2 (20 BYTE)
RED_CARD_RECEIVED VARCHAR2 (20 BYTE)
WAGES_IN_MILLION VARCHAR2 (20 BYTE)
FOOTBALL_PLAYER_PK (FOOTBALL_PLAYER_ID)
FOOTBALL_PLAYER_FK1 (PERSON_ID)
FOOTBALL_PLAYER_PK (FOOTBALL_PLAYER_ID)
FP_INDEX (FOOTBALL_PLAYER_ID, POSITION, MATCHES_PLAYED)
SRI.CLUB_DIRECTORY
P * CLUB_DIRECTORY_ID VARCHAR2 (10 BYTE)
F LEAGUE_ID VARCHAR2 (10 BYTE)
CLUB_NAME VARCHAR2 (150 BYTE)
ESTABLISHED_YEAR NUMBER
NICKNAME VARCHAR2 (75 BYTE)
HOME_STADIUM VARCHAR2 (200 BYTE)
CLUB_DIRECTORY_PK (CLUB_DIRECTORY_ID)
CLUB_DIRECTORY_FK1 (LEAGUE_ID)
CLUB_DIRECTORY_PK (CLUB_DIRECTORY_ID)
SRI.FRANCHISE
P * FRANCHISE_ID VARCHAR2 (10 BYTE)
F CLUB_ID VARCHAR2 (10 BYTE)
NAME VARCHAR2 (75 BYTE)
CONTRACT VARCHAR2 (20 BYTE)
BUDGET VARCHAR2 (20 BYTE)
FRANCHISE_PK (FRANCHISE_ID)
FRANCHISE_FK1 (CLUB_ID)
FRANCHISE_PK (FRANCHISE_ID)
SRI.GOALS
P * ROW_ID NUMBER
F CLUB_ID VARCHAR2 (10 BYTE)
GOALS_SCORED NUMBER
GOALS_CONCEDED NUMBER
PENALITIES NUMBER
GOALS_PK (ROW_ID)
GOALS_FK1 (CLUB_ID)
GOALS_PK (ROW_ID)
SRI.DEFENDING
F CLUB_ID VARCHAR2 (10 BYTE)
GOALS_CONCEDED NUMBER
SAVES_MADE NUMBER
OWN_GOALS NUMBER
BLOCKS NUMBER
DEFENDING_FK1 (CLUB_ID)
SRI.ATTACKING
F CLUB_ID VARCHAR2 (10 BYTE)
SHOTS NUMBER
CROSSES NUMBER
OFFSIDES NUMBER
ATTACKING_FK1 (CLUB_ID)
SRI.DISCIPLINARY
F CLUB_ID VARCHAR2 (10 BYTE)
FOULS NUMBER
YELLOW_CARDS NUMBER
RED_CARDS NUMBER
DISCIPLINARY_FK1 (CLUB_ID)
SRI.LEAGUE
P * LEAGUE_ID VARCHAR2 (10 BYTE)
LEAGUE_NAME VARCHAR2 (150 BYTE)
LEAGUE_PK (LEAGUE_ID)
LEAGUE_PK (LEAGUE_ID)
SRI.FIXTURES
P * FIXTURE_ID VARCHAR2 (20 BYTE)
MATCH_DATE VARCHAR2 (20 BYTE)
HOME_TEAM VARCHAR2 (20 BYTE)
AWAY_TEAM VARCHAR2 (20 BYTE)
STADIUM VARCHAR2 (20 BYTE)
FIXTURES_PK (FIXTURE_ID)
FIXTURES_PK (FIXTURE_ID)
SRI.PERSON_CATEGORY
P * PERSON_CATEGORY_ID VARCHAR2 (20 BYTE)
PERSON_CATEGORY VARCHAR2 (50 BYTE)
PERSON_CATEGORY_PK (PERSON_CATEGORY_ID)
PERSON_CATEGORY_PK (PERSON_CATEGORY_ID)
SRI.TRANSFER_CATEGORY
P * TRANSFER_CATEGORY_ID VARCHAR2 (10 BYTE)
TRANSFER_CATEGORY VARCHAR2 (20 BYTE)
TRANSFER_CATEGORY_PK (TRANSFER_CATEGORY_ID)
TRANSFER_CATEGORY_PK (TRANSFER_CATEGORY_ID)
SRI.IMAGE_LOCATION
P * IMAGE_ID NUMBER
IMAGE_LOC BLOB
IMAGE_LOCATION_PK (IMAGE_ID)
IMAGE_LOCATION_PK (IMAGE_ID)
90

For better clarity, all these three models are provided as separate image files.

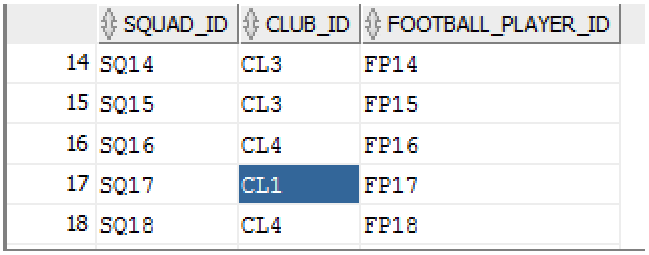
Queries Output:

Whenever a player is transferred from one club to another club (bought by a new club), a trigger is 'after_transfer_insert' executed which will call a procedure 'update_squad'.

This procedure will change the club name in the club table to new club name. Apart from that, there is another table 'PLAYER_CLUB_HISTORY' which tracks the players statistics club wise. Here a new record will be inserted for the player for new team to which he joined.

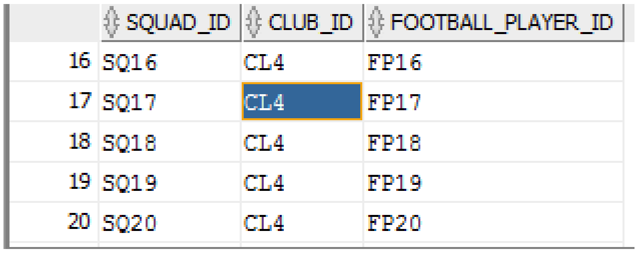
Here, in this example, football player of ID FP17 is transferred from Club1(CL1) to club4(CL4).

Before Trigger, the player is in club CL1.



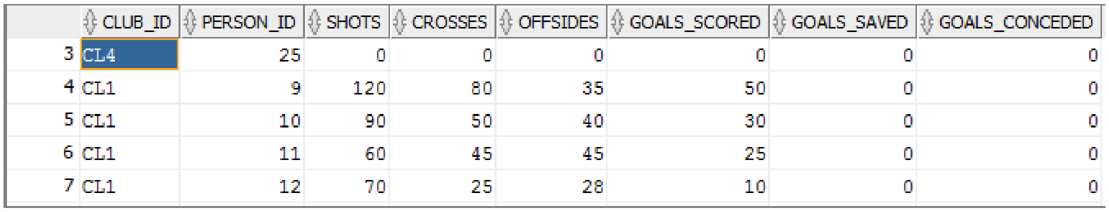
	SQUAD_ID	CLUB_ID	FOOTBALL_PLAYER_ID
14	SQ14	CL3	FP14
15	SQ15	CL3	FP15
16	SQ16	CL4	FP16
17	SQ17	CL1	FP17
18	SQ18	CL4	FP18

After Trigger, the player has been changed to new club CL4.



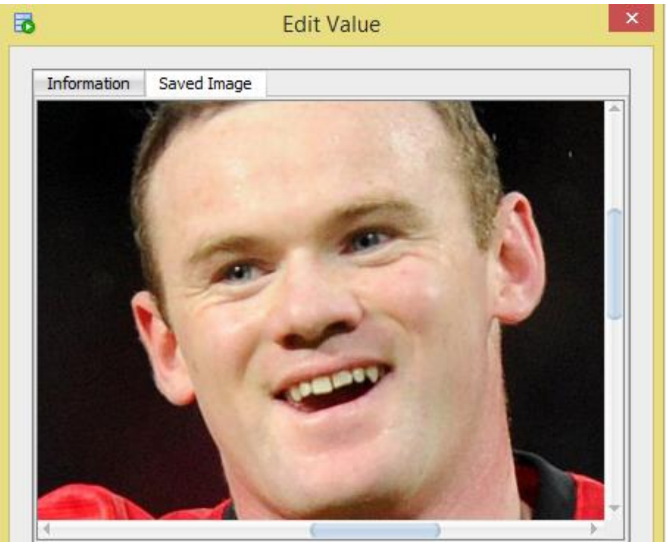
	SQUAD_ID	CLUB_ID	FOOTBALL_PLAYER_ID
16	SQ16	CL4	FP16
17	SQ17	CL4	FP17
18	SQ18	CL4	FP18
19	SQ19	CL4	FP19
20	SQ20	CL4	FP20

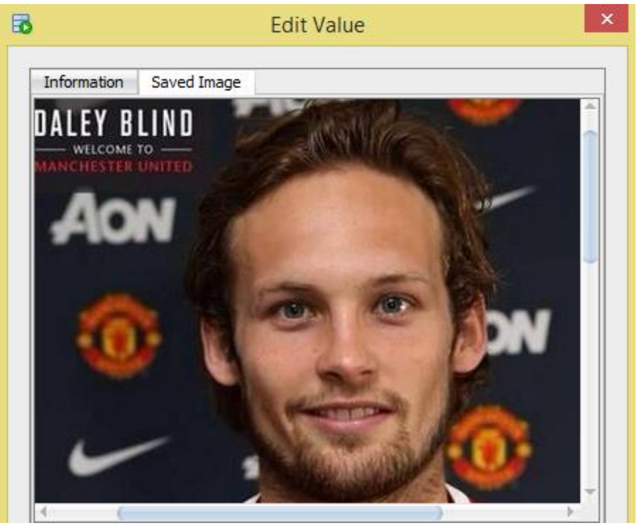
As the player is moved to new club CL4, his record (Goals, shots etc.) should be tracked for that particular club. Hence, the procedure updates the table with new club with all the record set to 0, so that it can be tracked from now on.

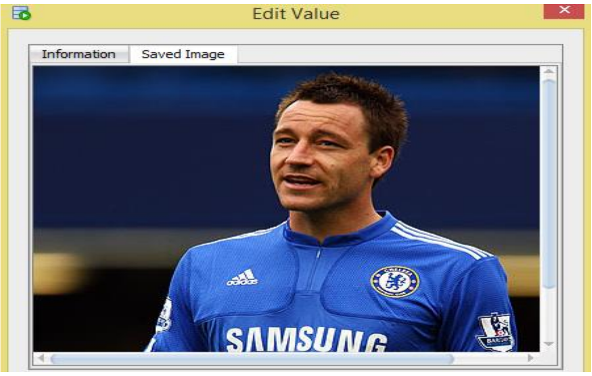


	CLUB_ID	PERSON_ID	SHOTS	CROSSES	OFFSIDES	GOALS_SCORED	GOALS_SAVED	GOALS_CONCEDED
3	CL4	25	0	0	0	0	0	0
4	CL1	9	120	80	35	50	0	0
5	CL1	10	90	50	40	30	0	0
6	CL1	11	60	45	45	25	0	0
7	CL1	12	70	25	28	10	0	0

Another Procedure IMAGE_INSERTION has been created to store the images of the players.

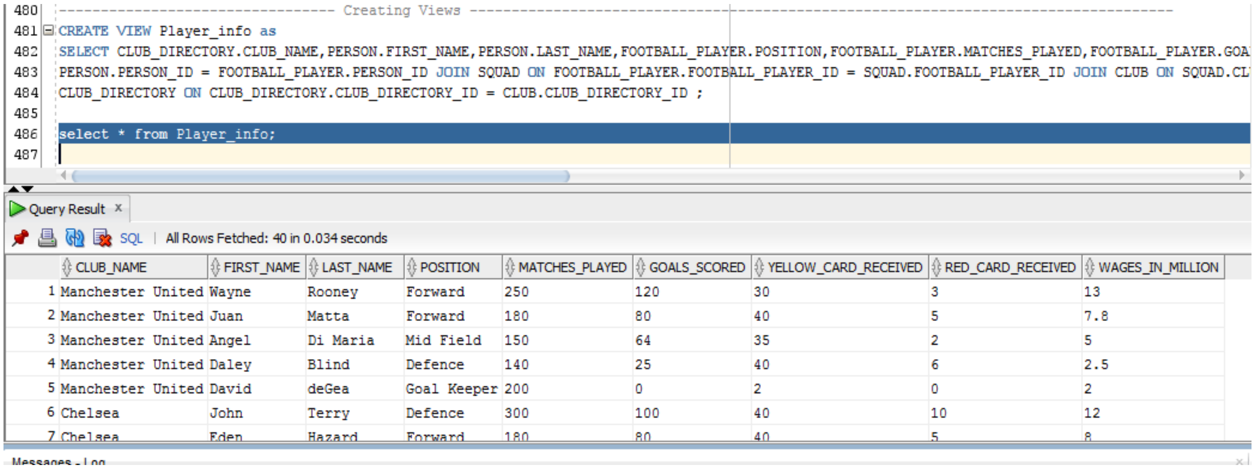






VIEWS:

A view Player_info has been created to get the information of players and their club history.



```
------------------------------- Creating Views ------------------------------------------------------------------------------
CREATE VIEW Player_info as
SELECT CLUB_DIRECTORY.CLUB_NAME,PERSON.FIRST_NAME,PERSON.LAST_NAME,FOOTBALL_PLAYER.POSITION,FOOTBALL_PLAYER.MATCHES_PLAYED,FOOTBALL_PLAYER.GOA
PERSON.PERSON_ID = FOOTBALL_PLAYER.PERSON_ID JOIN SQUAD ON FOOTBALL_PLAYER.FOOTBALL_PLAYER_ID = SQUAD.FOOTBALL_PLAYER_ID JOIN CLUB ON SQUAD.CL
CLUB_DIRECTORY ON CLUB_DIRECTORY.CLUB_DIRECTORY_ID = CLUB.CLUB_DIRECTORY_ID ;

select * from Player_info;
```

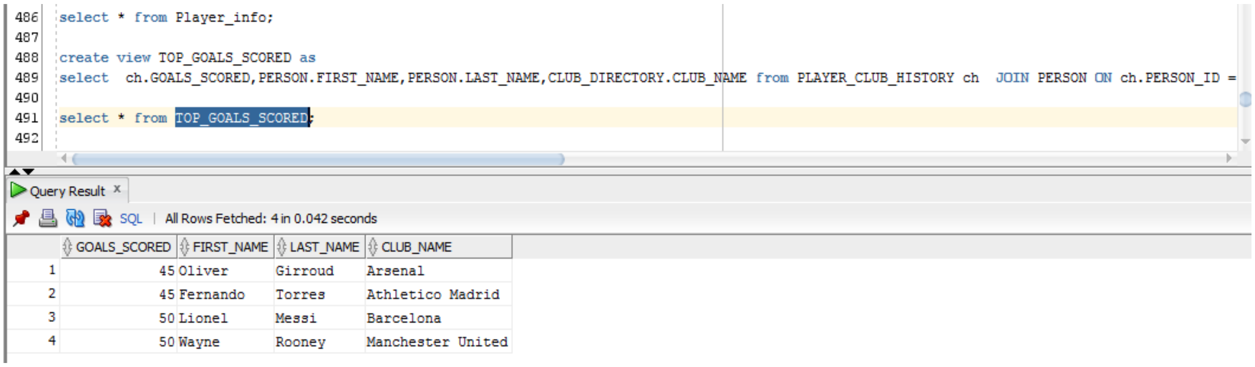
Query Result

SQL | All Rows Fetched: 40 in 0.034 seconds

	CLUB_NAME	FIRST_NAME	LAST_NAME	POSITION	MATCHES_PLAYED	GOALS_SCORED	YELLOW_CARD_RECEIVED	RED_CARD_RECEIVED	WAGES_IN_MILLION
1	Manchester United	Wayne	Rooney	Forward	250	120	30	3	13
2	Manchester United	Juan	Matta	Forward	180	80	40	5	7.8
3	Manchester United	Angel	Di Maria	Mid Field	150	64	35	2	5
4	Manchester United	Daley	Blind	Defence	140	25	40	6	2.5
5	Manchester United	David	deGea	Goal Keeper	200	0	2	0	2
6	Chelsea	John	Terry	Defence	300	100	40	10	12
7	Chelsea	Eden	Hazard	Forward	180	80	40	5	8

Messages - Log

Another view TOP_GOALS_SCORED has been created to identify top 4 goals scored by player belonging to both the leagues.



```
select * from Player_info;

create view TOP_GOALS_SCORED as
select  ch.GOALS_SCORED,PERSON.FIRST_NAME,PERSON.LAST_NAME,CLUB_DIRECTORY.CLUB_NAME from PLAYER_CLUB_HISTORY ch  JOIN PERSON ON ch.PERSON_ID =

select * from TOP_GOALS_SCORED;
```

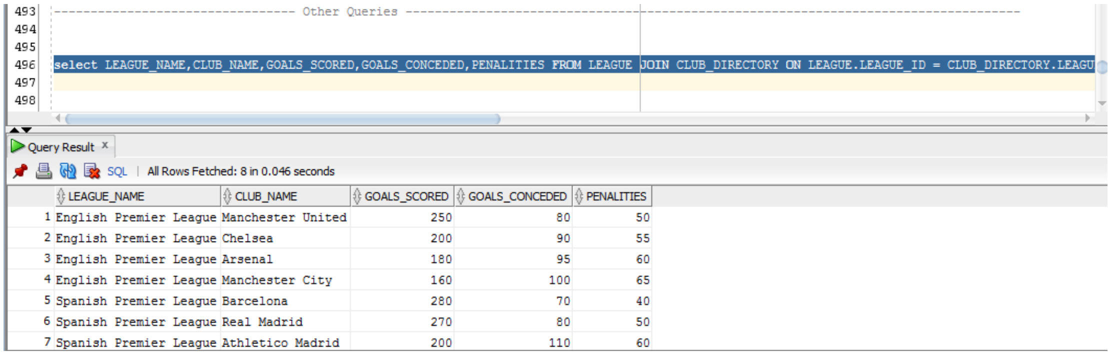
Query Result

SQL | All Rows Fetched: 4 in 0.042 seconds

	GOALS_SCORED	FIRST_NAME	LAST_NAME	CLUB_NAME
1	45	Oliver	Girroud	Arsenal
2	45	Fernando	Torres	Athletico Madrid
3	50	Lionel	Messi	Barcelona
4	50	Wayne	Rooney	Manchester United

Other Queries:

Below query provides the detail of individual clubs and their records.



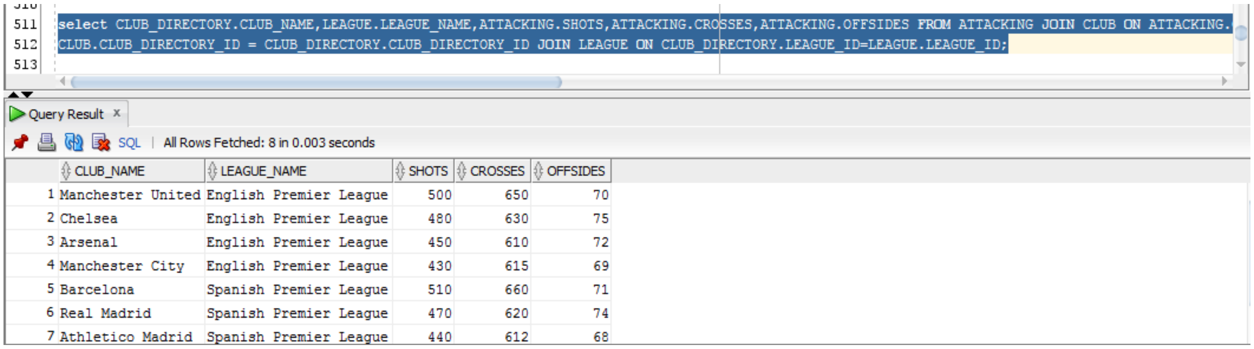
```
------------------------------- Other Queries -----------------------------------------------------------------------------------


select LEAGUE_NAME,CLUB_NAME,GOALS_SCORED,GOALS_CONCEDED,PENALITIES FROM LEAGUE JOIN CLUB_DIRECTORY ON LEAGUE.LEAGUE_ID = CLUB_DIRECTORY.LEAGU
```

Query Result

All Rows Fetched: 8 in 0.046 seconds

	LEAGUE_NAME	CLUB_NAME	GOALS_SCORED	GOALS_CONCEDED	PENALITIES
1	English Premier League	Manchester United	250	80	50
2	English Premier League	Chelsea	200	90	55
3	English Premier League	Arsenal	180	95	60
4	English Premier League	Manchester City	160	100	65
5	Spanish Premier League	Barcelona	280	70	40
6	Spanish Premier League	Real Madrid	270	80	50
7	Spanish Premier League	Athletico Madrid	200	110	60

Below query tracks another record of the clubs from Attacking perspective.



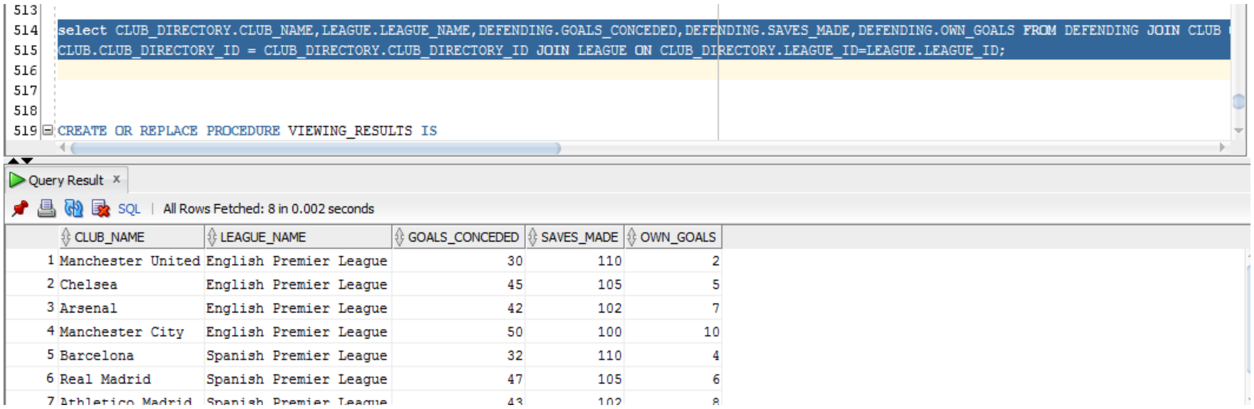
```
select CLUB_DIRECTORY.CLUB_NAME,LEAGUE.LEAGUE_NAME,ATTACKING.SHOTS,ATTACKING.CROSSES,ATTACKING.OFFSIDES FROM ATTACKING JOIN CLUB ON ATTACKING.
CLUB.CLUB_DIRECTORY_ID = CLUB_DIRECTORY.CLUB_DIRECTORY_ID JOIN LEAGUE ON CLUB_DIRECTORY.LEAGUE_ID=LEAGUE.LEAGUE_ID;
```

Query Result

All Rows Fetched: 8 in 0.003 seconds

	CLUB_NAME	LEAGUE_NAME	SHOTS	CROSSES	OFFSIDES
1	Manchester United	English Premier League	500	650	70
2	Chelsea	English Premier League	480	630	75
3	Arsenal	English Premier League	450	610	72
4	Manchester City	English Premier League	430	615	69
5	Barcelona	Spanish Premier League	510	660	71
6	Real Madrid	Spanish Premier League	470	620	74
7	Athletico Madrid	Spanish Premier League	440	612	68

From Defending perspective:



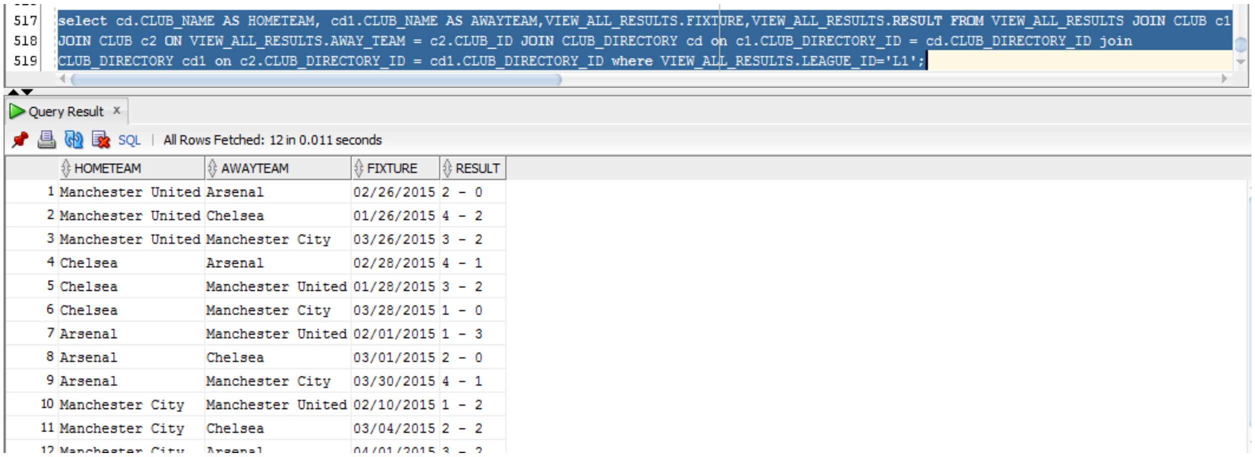
```
select CLUB_DIRECTORY.CLUB_NAME,LEAGUE.LEAGUE_NAME,DEFENDING.GOALS_CONCEDED,DEFENDING.SAVES_MADE,DEFENDING.OWN_GOALS FROM DEFENDING JOIN CLUB
CLUB.CLUB_DIRECTORY_ID = CLUB_DIRECTORY.CLUB_DIRECTORY_ID JOIN LEAGUE ON CLUB_DIRECTORY.LEAGUE_ID=LEAGUE.LEAGUE_ID;



CREATE OR REPLACE PROCEDURE VIEWING_RESULTS IS
```

Query Result

All Rows Fetched: 8 in 0.002 seconds

	CLUB_NAME	LEAGUE_NAME	GOALS_CONCEDED	SAVES_MADE	OWN_GOALS
1	Manchester United	English Premier League	30	110	2
2	Chelsea	English Premier League	45	105	5
3	Arsenal	English Premier League	42	102	7
4	Manchester City	English Premier League	50	100	10
5	Barcelona	Spanish Premier League	32	110	4
6	Real Madrid	Spanish Premier League	47	105	6
7	Athletico Madrid	Spanish Premier League	43	102	8

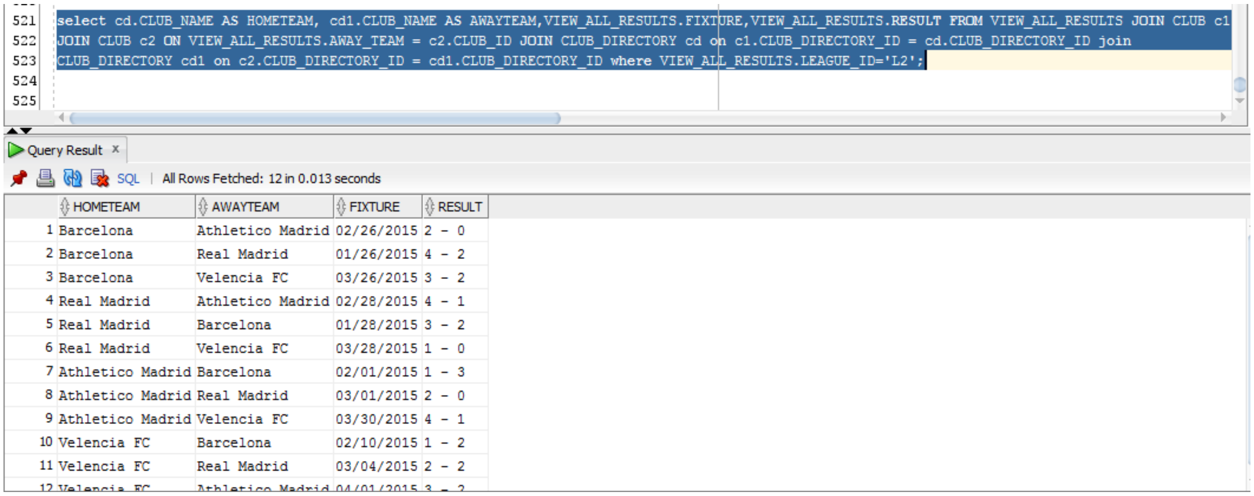
Below query retrieves all the results of the games played in League L1(English Premiere League)



```
select cd.CLUB_NAME AS HOMETEAM, cd1.CLUB_NAME AS AWAYTEAM,VIEW_ALL_RESULTS.FIXTURE,VIEW_ALL_RESULTS.RESULT FROM VIEW_ALL_RESULTS JOIN CLUB c1
JOIN CLUB c2 ON VIEW_ALL_RESULTS.AWAY_TEAM = c2.CLUB_ID JOIN CLUB_DIRECTORY cd on c1.CLUB_DIRECTORY_ID = cd.CLUB_DIRECTORY_ID join
CLUB_DIRECTORY cd1 on c2.CLUB_DIRECTORY_ID = cd1.CLUB_DIRECTORY_ID where VIEW_ALL_RESULTS.LEAGUE_ID='L1';
```

	HOMETEAM	AWAYTEAM	FIXTURE	RESULT
1	Manchester United	Arsenal	02/26/2015	2 - 0
2	Manchester United	Chelsea	01/26/2015	4 - 2
3	Manchester United	Manchester City	03/26/2015	3 - 2
4	Chelsea	Arsenal	02/28/2015	4 - 1
5	Chelsea	Manchester United	01/28/2015	3 - 2
6	Chelsea	Manchester City	03/28/2015	1 - 0
7	Arsenal	Manchester United	02/01/2015	1 - 3
8	Arsenal	Chelsea	03/01/2015	2 - 0
9	Arsenal	Manchester City	03/30/2015	4 - 1
10	Manchester City	Manchester United	02/10/2015	1 - 2
11	Manchester City	Chelsea	03/04/2015	2 - 2
12	Manchester City	Arsenal	04/01/2015	3 - 2

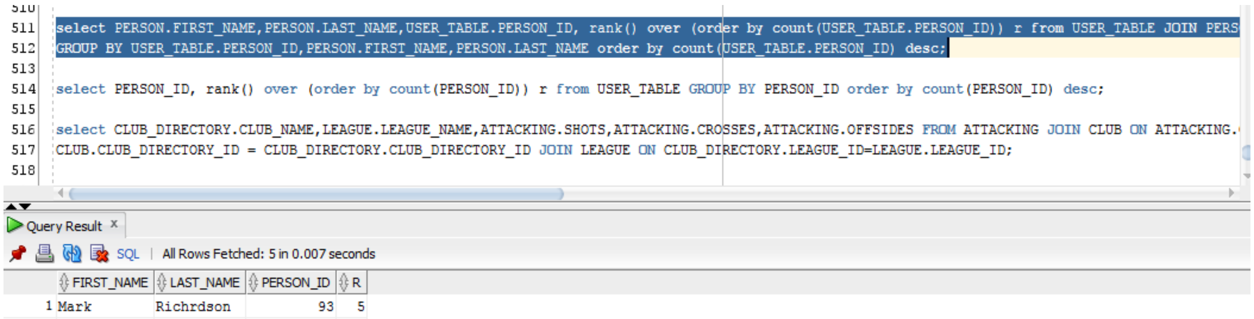
Below query retrieves all the results of the games played in League L2(Spanish Premiere League)



```
select cd.CLUB_NAME AS HOMETEAM, cd1.CLUB_NAME AS AWAYTEAM,VIEW_ALL_RESULTS.FIXTURE,VIEW_ALL_RESULTS.RESULT FROM VIEW_ALL_RESULTS JOIN CLUB c1
JOIN CLUB c2 ON VIEW_ALL_RESULTS.AWAY_TEAM = c2.CLUB_ID JOIN CLUB_DIRECTORY cd on c1.CLUB_DIRECTORY_ID = cd.CLUB_DIRECTORY_ID join
CLUB_DIRECTORY cd1 on c2.CLUB_DIRECTORY_ID = cd1.CLUB_DIRECTORY_ID where VIEW_ALL_RESULTS.LEAGUE_ID='L2';
```

	HOMETEAM	AWAYTEAM	FIXTURE	RESULT
1	Barcelona	Athletico Madrid	02/26/2015	2 - 0
2	Barcelona	Real Madrid	01/26/2015	4 - 2
3	Barcelona	Velencia FC	03/26/2015	3 - 2
4	Real Madrid	Athletico Madrid	02/28/2015	4 - 1
5	Real Madrid	Barcelona	01/28/2015	3 - 2
6	Real Madrid	Velencia FC	03/28/2015	1 - 0
7	Athletico Madrid	Barcelona	02/01/2015	1 - 3
8	Athletico Madrid	Real Madrid	03/01/2015	2 - 0
9	Athletico Madrid	Velencia FC	03/30/2015	4 - 1
10	Velencia FC	Barcelona	02/10/2015	1 - 2
11	Velencia FC	Real Madrid	03/04/2015	2 - 2
12	Velencia FC	Athletico Madrid	04/01/2015	3 - 2

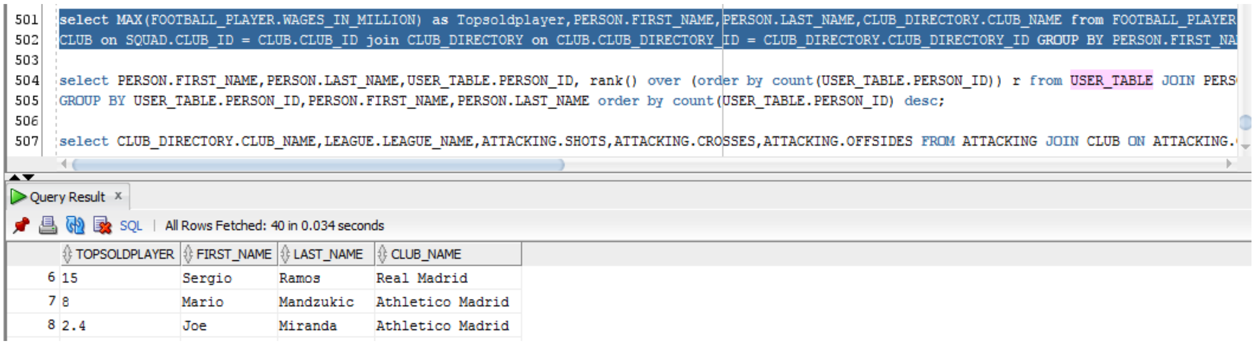
Query to find the user who predicted the result many times:



```
select PERSON.FIRST_NAME,PERSON.LAST_NAME,USER_TABLE.PERSON_ID, rank() over (order by count(USER_TABLE.PERSON_ID)) r from USER_TABLE JOIN PERS
GROUP BY USER_TABLE.PERSON_ID,PERSON.FIRST_NAME,PERSON.LAST_NAME order by count(USER_TABLE.PERSON_ID) desc;

select PERSON_ID, rank() over (order by count(PERSON_ID)) r from USER_TABLE GROUP BY PERSON_ID order by count(PERSON_ID) desc;

select CLUB_DIRECTORY.CLUB_NAME,LEAGUE.LEAGUE_NAME,ATTACKING.SHOTS,ATTACKING.CROSSES,ATTACKING.OFFSIDES FROM ATTACKING JOIN CLUB ON ATTACKING.
CLUB.CLUB_DIRECTORY_ID = CLUB_DIRECTORY.CLUB_DIRECTORY_ID JOIN LEAGUE ON CLUB_DIRECTORY.LEAGUE_ID=LEAGUE.LEAGUE_ID;
```

	FIRST_NAME	LAST_NAME	PERSON_ID	R
1	Mark	Richrdson	93	5

Query to find the wages of every Foot Ball.



```
select MAX(FOOTBALL_PLAYER.WAGES_IN_MILLION) as Topsoldplayer,PERSON.FIRST_NAME,PERSON.LAST_NAME,CLUB_DIRECTORY.CLUB_NAME from FOOTBALL_PLAYER
CLUB on SQUAD.CLUB_ID = CLUB.CLUB_ID join CLUB_DIRECTORY on CLUB.CLUB_DIRECTORY_ID = CLUB_DIRECTORY.CLUB_DIRECTORY_ID GROUP BY PERSON.FIRST_NA

select PERSON.FIRST_NAME,PERSON.LAST_NAME,USER_TABLE.PERSON_ID, rank() over (order by count(USER_TABLE.PERSON_ID)) r from USER_TABLE JOIN PERS
GROUP BY USER_TABLE.PERSON_ID,PERSON.FIRST_NAME,PERSON.LAST_NAME order by count(USER_TABLE.PERSON_ID) desc;

select CLUB_DIRECTORY.CLUB_NAME,LEAGUE.LEAGUE_NAME,ATTACKING.SHOTS,ATTACKING.CROSSES,ATTACKING.OFFSIDES FROM ATTACKING JOIN CLUB ON ATTACKING.
```

All Rows Fetched: 40 in 0.034 seconds

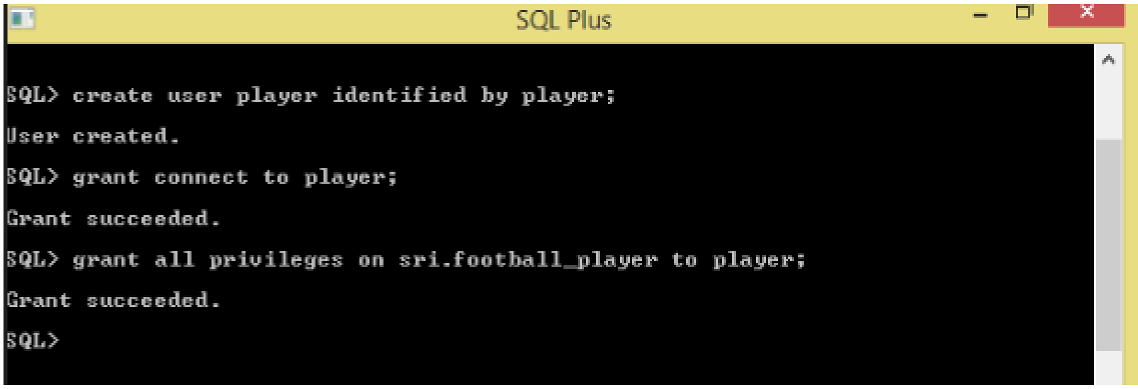
	TOPSOLDPLAYER	FIRST_NAME	LAST_NAME	CLUB_NAME
6	15	Sergio	Ramos	Real Madrid
7	8	Mario	Mandzukic	Athletico Madrid
8	2.4	Joe	Miranda	Athletico Madrid

INDEXES:

Indexes person_id and fp_index have been created for the tables PERSON and FOOTBALL_PLAYER as these two tables have more number of records and will be helpful for the faster retrieval of data from these tables.

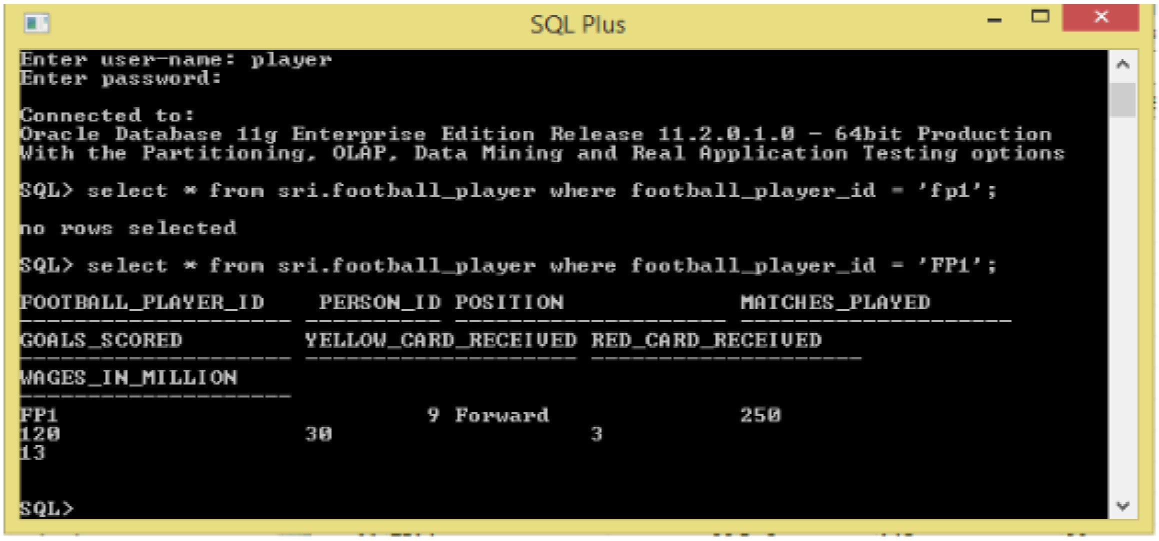
GRANTS:

New user Player has been created and he has been given all privileges to FOOTBALL_PLAYER table.



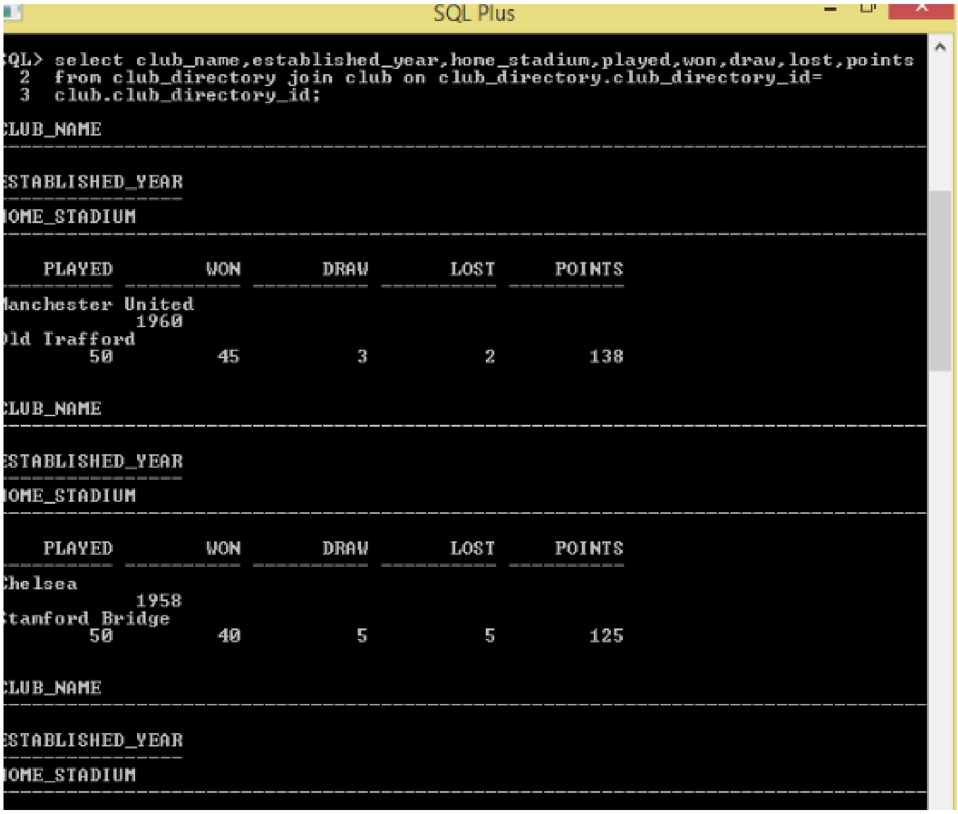
```
SQL> create user player identified by player;
User created.
SQL> grant connect to player;
Grant succeeded.
SQL> grant all privileges on sri.football_player to player;
Grant succeeded.
SQL>
```

He now has access to FOOTBALL_PLAYER table.

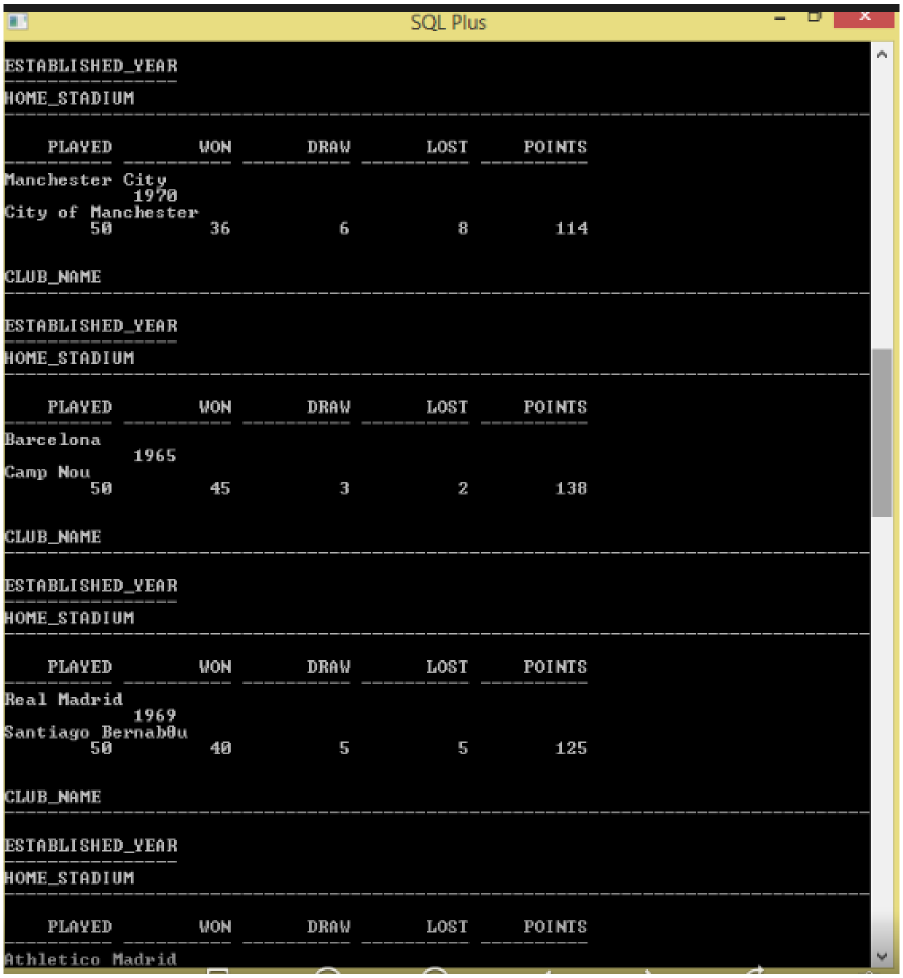


```
Enter user-name: player
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> select * from sri.football_player where football_player_id = 'fp1';

no rows selected

SQL> select * from sri.football_player where football_player_id = 'FP1';

FOOTBALL_PLAYER_ID    PERSON_ID POSITION             MATCHES_PLAYED
-------------------- ---------- -------------------- -------------------
GOALS_SCORED         YELLOW_CARD_RECEIVED RED_CARD_RECEIVED
-------------------- -------------------- --------------------
WAGES_IN_MILLION
--------------------
FP1                           9 Forward              250
120                  30                   3
13


SQL>
```

Another generic query executed on SQLPLUS:

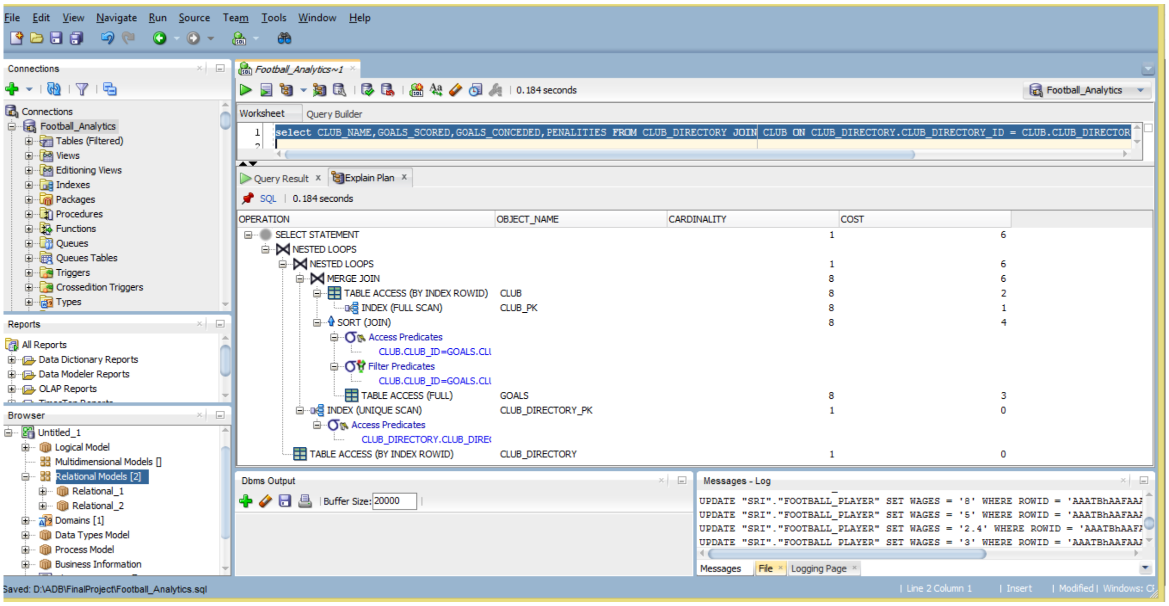


```
QL> select club_name,established_year,home_stadium,played,won,draw,lost,points
 2  from club_directory join club on club_directory.club_directory_id=
 3  club.club_directory_id;

LUB_NAME
--------------------------------------------------------------------------------

STABLISHED_YEAR
----------------
OME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
anchester United
            1960
ld Trafford
        50         45          3          2        138


LUB_NAME
--------------------------------------------------------------------------------

STABLISHED_YEAR
----------------
OME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
helsea
            1958
tanford Bridge
        50         40          5          5        125


LUB_NAME
--------------------------------------------------------------------------------

STABLISHED_YEAR
----------------
OME_STADIUM
--------------------------------------------------------------------------------
```



```
ESTABLISHED_YEAR
----------------
HOME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
Manchester City
            1970
City of Manchester
        50         36          6          8        114

CLUB_NAME
--------------------------------------------------------------------------------

ESTABLISHED_YEAR
----------------
HOME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
Barcelona
            1965
Camp Nou
        50         45          3          2        138

CLUB_NAME
--------------------------------------------------------------------------------

ESTABLISHED_YEAR
----------------
HOME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
Real Madrid
            1969
Santiago Bernab0u
        50         40          5          5        125

CLUB_NAME
--------------------------------------------------------------------------------

ESTABLISHED_YEAR
----------------
HOME_STADIUM
--------------------------------------------------------------------------------

    PLAYED        WON       DRAW       LOST     POINTS
---------- ---------- ---------- ---------- ----------
Athletico Madrid
```

EXPLAIN PLAN SAMPLE:



```
select CLUB_NAME,GOALS_SCORED,GOALS_CONCEDED,PENALITIES FROM CLUB_DIRECTORY JOIN CLUB ON CLUB_DIRECTORY.CLUB_DIRECTORY_ID = CLUB.CLUB_DIRECTOR
```

```
OPERATION                                   OBJECT_NAME          CARDINALITY   COST
SELECT STATEMENT                                                           1      6
  NESTED LOOPS
    NESTED LOOPS                                                           1      6
      MERGE JOIN                                                           8      6
        TABLE ACCESS (BY INDEX ROWID)       CLUB                           8      2
          INDEX (FULL SCAN)                 CLUB_PK                        8      1
        SORT (JOIN)                                                        8      4
          Access Predicates
            CLUB.CLUB_ID=GOALS.CLU
          Filter Predicates
            CLUB.CLUB_ID=GOALS.CLU
          TABLE ACCESS (FULL)               GOALS                          8      3
      INDEX (UNIQUE SCAN)                   CLUB_DIRECTORY_PK              1      0
        Access Predicates
          CLUB_DIRECTORY.CLUB_DIREC
    TABLE ACCESS (BY INDEX ROWID)           CLUB_DIRECTORY                 1      0
```

```
UPDATE "SRI"."FOOTBALL_PLAYER" SET WAGES = '8' WHERE ROWID = 'AAATBhAAFAAA
UPDATE "SRI"."FOOTBALL_PLAYER" SET WAGES = '5' WHERE ROWID = 'AAATBhAAFAAA
UPDATE "SRI"."FOOTBALL_PLAYER" SET WAGES = '2.4' WHERE ROWID = 'AAATBhAAFA
UPDATE "SRI"."FOOTBALL_PLAYER" SET WAGES = '3' WHERE ROWID = 'AAATBhAAFAAA
```

Saved: D:\ADB\FinalProject\Football_Analytics.sql

Database Maintenance plan:

The database should be up and available most of the times in order to make sure that all the users accessing it should be benefited. The database maintenance plan should be designed and installed so that there would be always as backup plan in case of any real time disaster or any other database calamities.

Oracle Database has several automated maintenance tasks that are performed by database administrators. Normally these maintenance are executed whenever the system load is light. These tasks start automatically at regular intervals in order to perform maintenance operation in database. The time intervals for running the tasks are predefined and these tasks run in this window. These task are achieved by:

Automatic Optimizer Statistics Collection

Automatic Segment Advisor

Automatic SQL Tuning Advisor.

These days in a real time production environment, a technique named “Data Mirroring” is followed where all the production data will be mirrored into an alternate back up environment. All the data will be updated and saved in this environment similar to that of production. The mirroring will be taken in a particular interval but during recovery, last available image will be used. Apart from that, there are image copies which takes the backup of the database. These image copies in the form jobs or any other means can be executed prior to updating the database and it contains all the information of database prior to current day. Once after the execution of these image copies, the current day database can be updated. In this manner, back up can be made on a daily basis.

Whenever a disaster occurs, the recent image prior to the disaster timing will be recovered in the Business Continuation site. Verification will be made base on the time of disaster and whatever transactions that got hung up will be restarted. The database can also be recovered from recent images taken by the image copies and can be brought up and available in production. In order to make sure that the recovery process takes place in a normal manner, business continuation test should be executed in a test environment to make sure the proper recovery is done. These tests can be conducted twice or thrice a year.

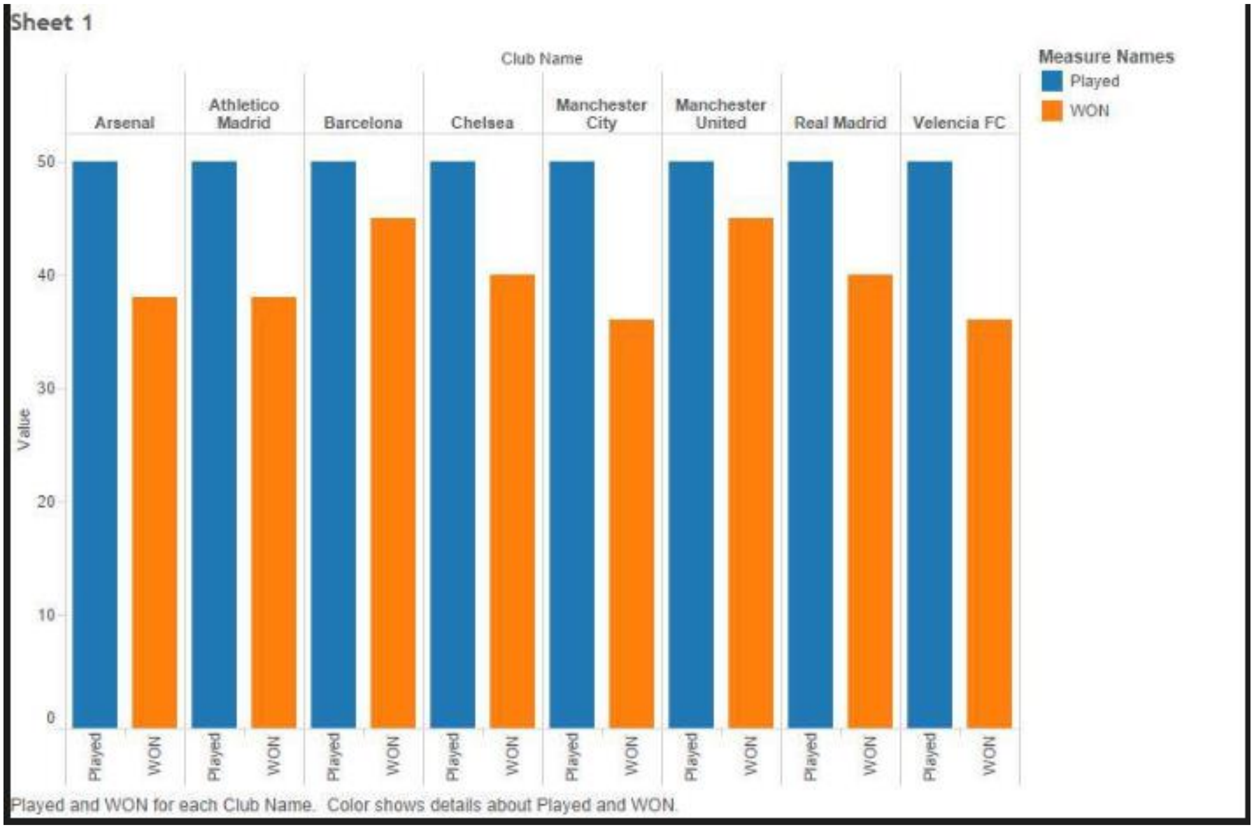
A failover is when primary database fails and one of the standby database is transitioned to take the primary role. Failover is performed whenever primary database fails and primary database cannot be recovered in timely manner. While performing a fail over, the standby database that has most archived data in it should be chosen. By doing this, amount of data loss can be minimized. If the configuration has both physical and logical databases, it is always effective to choose a physical standby database because all standby databases in the configuration will still be available as standby databases to the new primary database after the failover operation completes.

BI TOOLS:

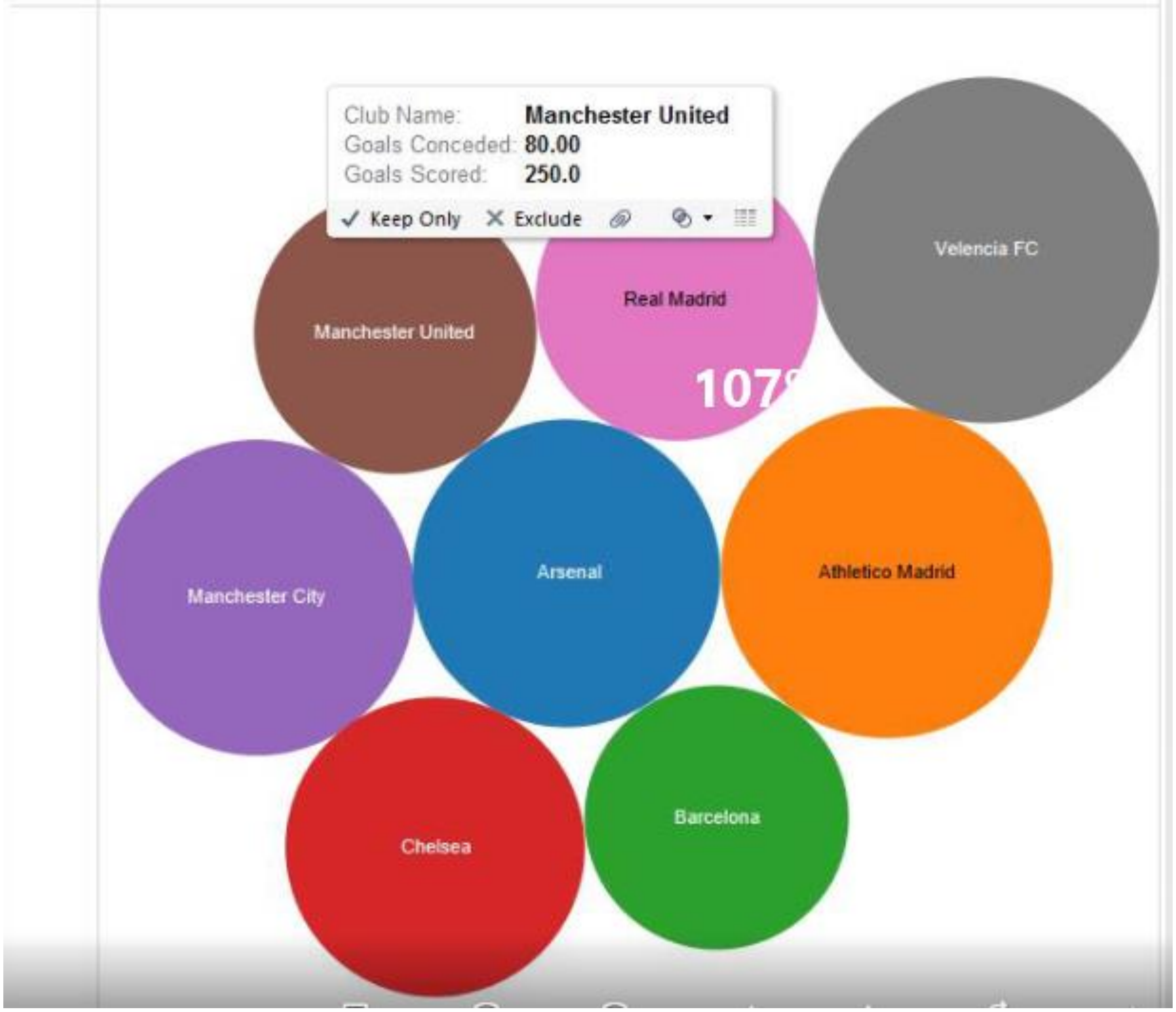
The data from the tables can be loaded into BI tools and we can do various analysis using that.

Here I have used Tableao tool to generate the below mentioned charts.

Below graph shows the number of matches won by each club in all the two leagues.

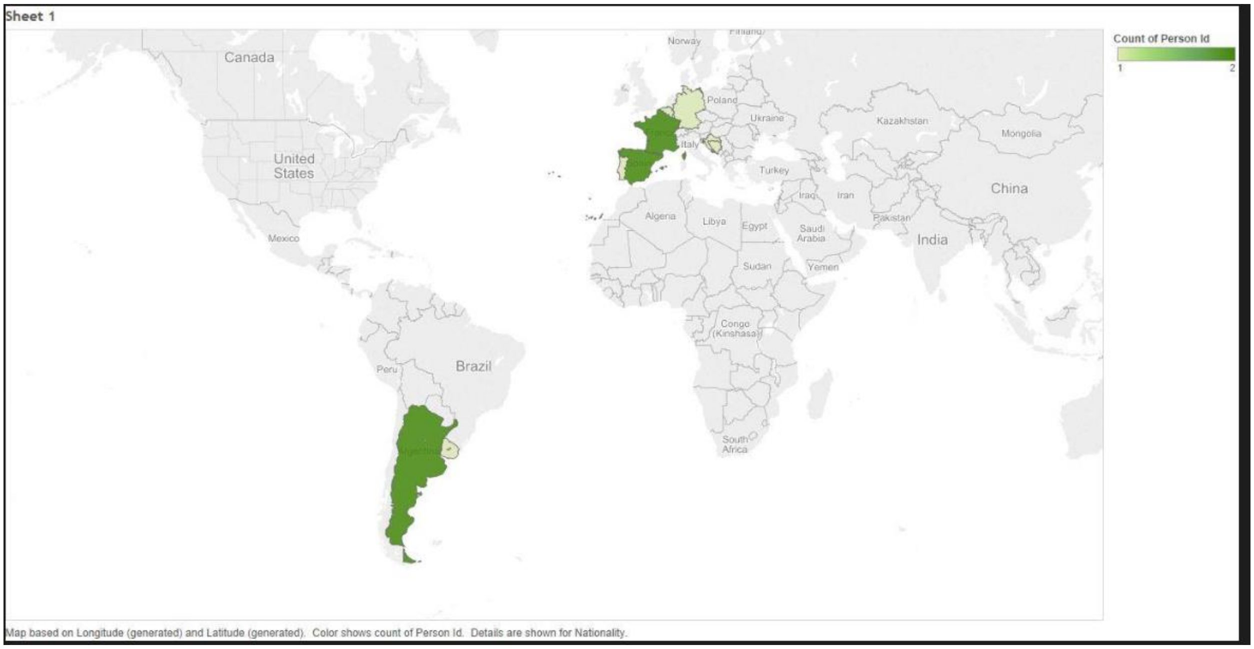


The below graph shows the number of goals scored and conceded by each team in the league.



Club Name: Manchester United
Goals Conceded: 80.00
Goals Scored: 250.0

The below graph shows the distribution of players across the world. The players are from region that has green shades.



Map based on Longitude (generated) and Latitude (generated). Color shows count of Person Id. Details are shown for Nationality.

REFERENCES:

http://docs.oracle.com/cd/B19306_01/server.102/b14230/sofo.htm

http://docs.oracle.com/cd/E11882_01/server.112/e25494/tasks.htm#ADMIN11836