

# **DATABASE** **MANAGEMENT** **SYSTEMS**

ITE1003-PROJECT

Project By:

BHARATH RAJ G.N - 19BIT0364

PRAVIN G - 19BIT0393

SRIDHARAN V.R- 19BIT0396

Slot : A1

L41+L42

TOPIC:

**FOOTBALL CLUB  
MANAGEMENT  
SYSTEM**

# Review -1

## Football Club Management System

### Introduction:

This Project is about an FOOTBALL CLUB MANAGEMENT. It is a project which is helpful in the areas of Football. Our main aim is maintain records and make analysis based on reports generated. Admin can login as administrator . this will allow admin to get full control to the System. The actual need for the new Football Club Management System is to reduce paper work and Time. This System can Store Details of Players, Mangers, Clubs , Transfer Details during Bidding etc . This system is built to reduce complexity of system for the users handling the system. This system is a web-based application.

Football Club Management System Users to get all latest information About football and training. Admin can access all Information About Player, Teams and their Performance. Easy to Update and Search any Record.

### Data Requirements:

Entities -

- 1) **User** is an entity type which has many attributes like ,
  - ✓ Name is a user's Original Name of which is not null.
  - ✓ User Name is a User's wise name given which is not null, unique.
  - ✓ Password which users password and not null.
  - ✓ Every User has been given a User ID which is unique and not null.
  - ✓ DoB which is not null to store Date of birth of user.
  - ✓ Nationality which is user's nationality and cannot be null.
  - ✓ Status which says that a user is a banned or not (i.e null).
- 2) **Players** is an entity type which has many attributes like ,
  - ✓ Every Player has been given a Player ID which is unique and not null.
  - ✓ Position which is not null and this describes the position in which a player plays during the match.
  - ✓ It has a foreign key, User ID which is the primary key of the entity, User.
  - ✓ Bid amount which cannot be null.
  - ✓ Sale which is the minimum amount a player fixed and cannot be null.
  - ✓ Rating which describes the skill rating of a player and not null.
  - ✓ Skill-star and Weak-foot which describes the statistics of Player and not null.
- 3) **Mangers** is an entity type which has many attributes like ,
  - ✓ Every Manager has been given a Mgr ID which is unique and not null.
  - ✓ No. Of Matches played and Years of Exp which describes the statistics of manager.

- 4) **Clubs** is an entity type which has many attributes like ,
- ✓ Club Name which is a name of Club/Team and not null, unique.
  - ✓ Each Club has been given a Club ID which is unique and not null.
  - ✓ Trophies and No. of Goals which describes the statistics of Club and not null.
  - ✓ Win, Loss, Draws, Matches Played and Total points are those attributes which cannot be null.
- 5) **Bids** is an entity type which has many attributes like ,
- ✓ It has a foreign key, Mgr ID and Player ID which is the primary key of the entity, Managers and players respectively.
  - ✓ Every Bid is placed by the managers and on the players so a Mgr ID and Player\_ID will be unique and not null.
  - ✓ Bid ID has a type not null unique.
  - ✓ Bid Amount which cannot be null.
- 6) **Training** is an entity type which has many attributes like ,
- ✓ Club Name which is foreign key and not null.
  - ✓ Date and Time is when a player's training is scheduled and not null.

## Relationships:

### **I. Users are Players :**

User must be either Player or Manager. So, Many Users may be Players and all Players are users.

### **II. Users are Managers :**

User must be either Manager or Player. So, Many Users may be Managers and all Managers are users

### **III. Managers places Bids :**

A Manager can place a many Bid and also many Managers can place on a same Bid. And no total participation.

### **IV. Manager manages Club :**

Club must have one Manager. So, A Manager can manage only one Club or a Club can be managed by only one Manager.

### **V. Manager organizes Training :**

A Manager can organize many Training to players but a single training can be organized by only one Manager. And no total participation.

### **VI. Bids On Players :**

A Bid can be placed on many Players and many Bids can be placed on a single Player And no total participation.

### **VII. Player belongs Clubs :**

Club must have a Player. A Club can have many Players but a Player can belong to only one Club.

## Functional Requirements:

### 1) Administrator :

- Administrator has all the privileges of the user but has the authority to add and remove data from the database which the user cannot do because he/she is in charge of creating the website which is used to access the database.
- He/she is responsible for creating different user accounts and assigning the id and password.
- Administrators are the one who keep updating the each club details like trophies, win, loss, draws, matches played etc in the database.
- They should be allowed to enter the club name of home and away team. He should have the authority to enter and modify the match details like time and venue in case the need to be changed.
- If any player/manager has been banned for some reasons the player/manger's user status has to be modified as ban from the database.
- The point table of every club must keep being modified after each match.

### BASIC ANALOGY:

- ✧ Creates the Football Club Management website
- ✧ Gives a unique user-ID for viewers
- ✧ Well designed website
- ✧ Display menus
- ✧ Schedule Auction
- ✧ Display Player Name
- ✧ Display Player's Bid amount
- ✧ Display Club Manager's Bid on a Player

### View Players information per Bid:

- ✓ Name
- ✓ Nationality
- ✓ Skills/Position
- ✓ Bid amount

### REMOVAL OF OLD DATA:

- a) During Biding if the Bid is finalized then the Bid data has to be removed.
- b) If the training is cancelled then that training has to be removed
- c) If any Club's Manger gets changed then the previous Manager's data has to removed.
- d) If any Player announces his retirement then that player data has to be removed from database.

### MODIFICATION OF DATA:

- a) If any Club's Manger or a Player gets ban( i.e RED CARD) then, his status has to be modified for next some matches.
- b) If Training schedule is change then it has to be modified.
- c) The Points table of each Club has to be modified after each match.
- d) Bid amount have to be modified if some other manager places bid more than earlier

## RETRIEVAL OF DATA:

### a) View information of every Bid:

Before a Bid on a Player, we have to retrieve the Player record like:

- a. Name
- b. Nationality
- c. Skills
- d. Position
- e. Bid Amount

### b) View information of every Team:

Before the start of a new match, we have to retrieve the Team record like:

- a. Club Name
- b. Goals
- c. Win
- d. Loss
- e. Draws
- f. Trophies

### c) View Score Board:

After every match, we have to retrieve the ranking order of Clubs:

- a. Points
- b. Club name

### d) View the Managers of each Club:

We need to retrieve the data of Manager of particular club:

- a. Name
- b. Years Experienced

### e) View members of particular team:

Before every match we need to retrieve Manger name and Player names of Particular team

- a. Name of managers and players
- b. Role of the member

### f) View Skills information of each player:

If a Player learns new skills or scores many Goals, then we need to retrieve the data of Players Skill:

- a. Rating
- b. Skill-Star
- c. Player ID

### g) View information of trainings:

For Management purpose we may need to retrieve training information

- a. Training ID

b. Total Players

h) View Club information per Goal:

If a Team scores a goal .We need to retrieve the data of club. If a person scores a goal in a match we need to retrieve the data of club:

- a. Club\_Name
- b. Goals
- c. Trophies

2) Viewer:

- First of a system should allow users to login if they enter correct User Name and Password.
- The details of their Manager and Club must also be available to the Players.
- The Players should be able to update their skills and other details.
- System should display the complete roster of a Club including the Manager and the Players playing and the current rank of the Club.
- Each Manager's statistics should also be available like total goals, number of matches played ,won,loss etc.
- System should display data on each Bid which has been placed during the Auction.
- System should allow Players to be searched and the Bid been placed.

BASIC ANALOGY:

- ✧ View the Football Management website
- ✧ Login to it
- ✧ View their Manager details
- ✧ View the Player's details
- ✧ View their Club details
- ✧ View their On going Biding details
- ✧ View Rank
- ✧ View statistics of Mangers or Clubs
- ✧ View the Points table

View Players information per Bid:

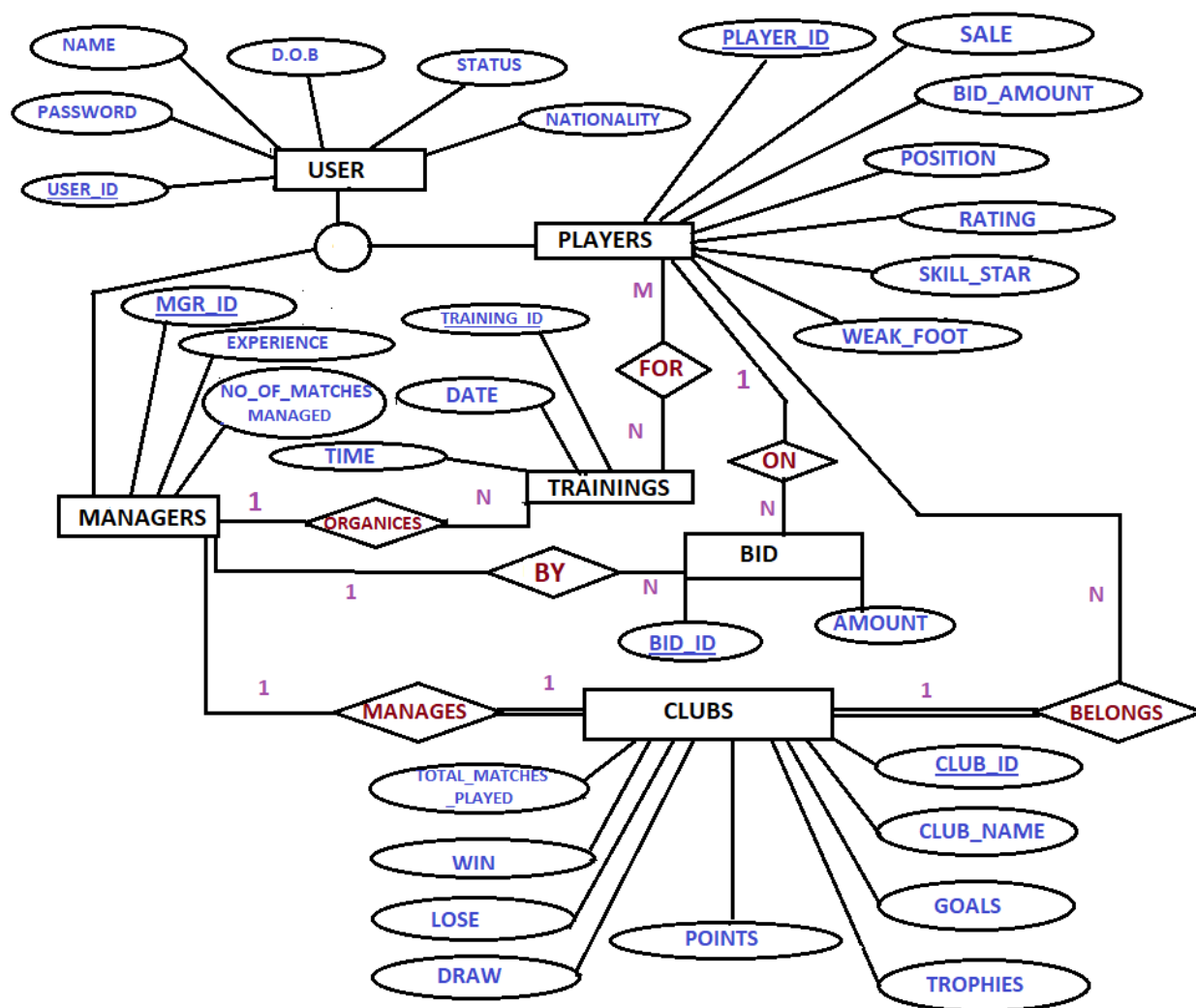
- ✓ Name
- ✓ Nationality
- ✓ Skills/Position
- ✓ Bid amount

View Bid information per sale:

- ✓ Manger Name
- ✓ Player Name ( i.e SOLD)
- ✓ Bid amount



## EER MODEL along with key constraints, participation constraints and cardinality constraints:



# Review -2

## Football Club Management System

Relational database schema diagram:

USERS

<u>USER_ID</u>	PASSWORD	NAME	DOB	STATUS	NATIONALITY
----------------	----------	------	-----	--------	-------------

MANAGERS

<u>MGR_ID</u>	EXPERIENCE	NO_OF_MATCHES_MANAGED	USER_ID
---------------	------------	-----------------------	---------

CLUBS

<u>CLUB_ID</u>	CLUB_NAME	GOALS	TROPHIES	TOT_MATCHES_PLAYED	WIN	LOSE	DRAW	POINTS	MGR_ID
----------------	-----------	-------	----------	--------------------	-----	------	------	--------	--------

PLAYERS

<u>PLAYER_ID</u>	SALE	BID_AMOUNT	POSITION	RATING	SKILL_STAR	WEAK_FOOT	USER_ID	CLUB_ID
------------------	------	------------	----------	--------	------------	-----------	---------	---------

TRAINING\_FOR

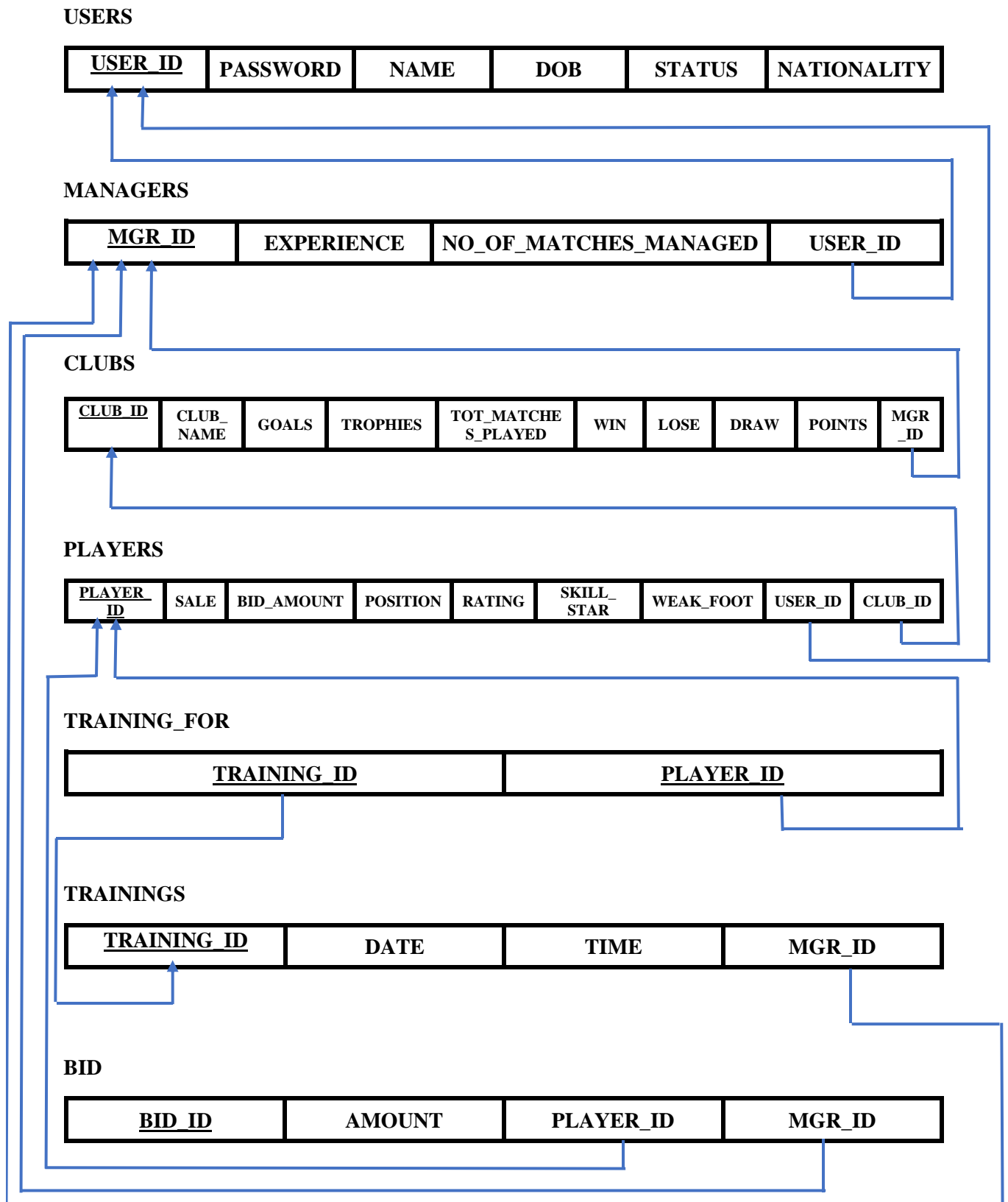
<u>TRAINING_ID</u>	<u>PLAYER_ID</u>
--------------------	------------------

TRAININGS

<u>TRAINING_ID</u>	DATE	TIME	MGR_ID
--------------------	------	------	--------

BID

<u>BID_ID</u>	AMOUNT	PLAYER_ID	MGR_ID
---------------	--------	-----------	--------



## Implementation of the relational database schema:-

### **TABLES:-**

- ❖ **USERS**
- ❖ **MANAGERS**
- ❖ **CLUBS**
- ❖ **PLAYERS**
- ❖ **TRAININGS**
- ❖ **TRAINING FOR**
- ❖ **BID**

# TABLE USERS

```
create table users(  
    user_id          varchar(5)  
                    constraint usr_pk primary key,  
    password         varchar(16)      constraint usr_nn not null,  
    name             char(20)         constraint usr_nn2 not null,  
    dob              date,  
    status           char(10),  
    nationality       char(20)  
);
```

```
SQL> create table users(  
  2  user_id      varchar(5)      constraint usr_pk primary key,  
  3  password    varchar(16)      constraint usr_nn not null,  
  4  name        char(20)         constraint usr_nn2 not null,  
  5  dob         date,  
  6  status      char(10),  
  7  nationality  char(20)  
  8  );
```

Table created.

```
SQL> desc users;
```

Name	Null?	Type
USER_ID	NOT NULL	VARCHAR2(5)
PASSWORD	NOT NULL	VARCHAR2(16)
NAME	NOT NULL	CHAR(20)
DOB		DATE
STATUS		CHAR(10)
NATIONALITY		CHAR(20)

```
insert into users values('U#23','password1','Joe Gomez',  
TO_DATE('23/05/1997', 'DD/MM/YYYY'),'Active','England');  
insert into users values('U#07','password2','Mohamed Salah',  
TO_DATE('15/06/1992', 'DD/MM/YYYY'),'Active','Egypt');  
insert into users values('U#99','password3','David Silva',  
TO_DATE('08/01/1986', 'DD/MM/YYYY'),'Active','Spain');  
insert into users values('U#34','password4','Sergio Aguero',  
TO_DATE('02/06/1988', 'DD/MM/YYYY'),'Active','Argentina');  
insert into users values('U#86','password5','Toni Kroos',  
TO_DATE('04/01/1990', 'DD/MM/YYYY'),'Active','German');  
insert into users values('U#49','password6','Raphhael Varane',
```

```

TO_DATE('25/04/1993', 'DD/MM/YYYY'), 'Active', 'French');
insert into users values('U#27', 'password7', 'Ivo Grbic',
TO_DATE('18/01/1996', 'DD/MM/YYYY'), 'Active', 'Croatian');
insert into users values('U#64', 'password8', 'Jose Grimenez',
TO_DATE('20/01/1995', 'DD/MM/YYYY'), 'Active', 'Uruguayan');
insert into users values('U#14', 'password9', 'Diego Godin',
TO_DATE('16/02/1986', 'DD/MM/YYYY'), 'Active', 'Uruguayan');
insert into users values('U#04', 'password10', 'Alexis Sanchez',
TO_DATE('19/12/1988', 'DD/MM/YYYY'), 'Active', 'Chilean');
insert into users values('U#38', 'password11', 'Antonio Conte',
TO_DATE('31/07/1969', 'DD/MM/YYYY'), 'Active', 'Italian');
insert into users values('U#91', 'password12', 'Diego Simeone',
TO_DATE('28/04/1970', 'DD/MM/YYYY'), 'Active', 'Argentine');
insert into users values('U#11', 'password13', 'Zinedine Zidane',
TO_DATE('23/06/1972', 'DD/MM/YYYY'), 'Active', 'Algerian');
insert into users values('U#41', 'password14', 'Pep Guardiola',
TO_DATE('18/01/1971', 'DD/MM/YYYY'), 'Active', 'Spanish');
insert into users values('U#77', 'password15', 'Jurgen Klopp',
TO_DATE('16/06/1967', 'DD/MM/YYYY'), 'Active', 'German');

```

USER_ID	PASSWORD	NAME	DOB	STATUS	NATIONALITY
U#23	password1	Joe Gomez	23-MAY-97	Active	England
U#07	password2	Mohamed Salah	15-JUN-92	Active	Egypt
U#99	password3	David Silva	08-JAN-86	Active	Spain
U#34	password4	Sergio Aguero	02-JUN-88	Active	Argentina
U#86	password5	Toni Kroos	04-JAN-90	Active	German
U#49	password6	Raphhael Varane	25-APR-93	Active	French
U#27	password7	Ivo Grbic	18-JAN-96	Active	Croatian
U#64	password8	Jose Grimenez	20-JAN-95	Active	Uruguayan
U#14	password9	Diego Godin	16-FEB-86	Active	Uruguayan
U#04	password10	Alexis Sanchez	19-DEC-88	Active	Chilean
U#38	password11	Antonio Conte	31-JUL-69	Active	Italian
U#91	password12	Diego Simeone	28-APR-70	Active	Argentine
U#11	password13	Zinedine Zidane	23-JUN-72	Active	Algerian
U#41	password14	Pep Guardiola	18-JAN-71	Active	Spanish
U#77	password15	Jurgen Klopp	16-JUN-67	Active	German

# TABLE MANAGERS

```
create table managers(  
    mgr_id          varchar(5)  
                   constraint mgr_pk primary key,  
    experience      number(2),  
    no_of_matches_managed number(4),  
    user_id         constraint mgr_fk references users  
);
```

```
SQL> create table managers(  
2  mgr_id          varchar(5)          constraint mgr_pk primary key,  
3  experience      number(2),  
4  no_of_matches_managed number(4),  
5  user_id         constraint mgr_fk references users  
6  );
```

Table created.

```
SQL> desc managers;
```

Name	Null?	Type
MGR_ID	NOT NULL	VARCHAR2(5)
EXPERIENCE		NUMBER(2)
NO_OF_MATCHES_MANAGED		NUMBER(4)
USER_ID		VARCHAR2(5)

```
insert into managers values ('JK11M',11,186,'U#77');  
insert into managers values ('AC23M',5,76,'U#38');  
insert into managers values ('DS56M',13,213,'U#91');  
insert into managers values ('ZZ01M',2,66,'U#11');  
insert into managers values ('PG15M',4,155,'U#41');
```

MGR_ID	EXPERIENCE	NO_OF_MATCHES_MANAGED	USER_ID
JK11M	11	186	U#77
AC23M	5	76	U#38
DS56M	13	213	U#91
ZZ01M	2	66	U#11
PG15M	4	155	U#41

## TABLE CLUBS:-

```
create table clubs(  
    club_id          varchar(5) constraint clb_pk primary key,  
    club_name        char(20)   constraint clb_nn not null,  
    goals            number(4),  
    trophies          number(3),  
    total_matches_played number(4),  
    win               number(4),  
    lose              number(4),  
    draw              number(4),  
    points            number(4),  
    mgr_id            constraint clb_fk references managers  
);
```

```
SQL> create table clubs(  
2  club_id          varchar(5)          constraint clb_pk primary key,  
3  club_name        char(20)           constraint clb_nn not null,  
4  goals            number(4),  
5  trophies          number(3),  
6  total_matches_played number(4),  
7  win               number(4),  
8  lose              number(4),  
9  draw              number(4),  
10 points            number(4),  
11 mgr_id            constraint clb_fk references managers  
12 );
```

Table created.

```
SQL> desc clubs;
```

Name	Null?	Type
CLUB_ID	NOT NULL	VARCHAR2(5)
CLUB_NAME	NOT NULL	CHAR(20)
GOALS		NUMBER(4)
TROPHIES		NUMBER(3)
TOTAL_MATCHES_PLAYED		NUMBER(4)
WIN		NUMBER(4)
LOSE		NUMBER(4)
DRAW		NUMBER(4)
POINTS		NUMBER(4)
MGR_ID		VARCHAR2(5)

```
insert into clubs values ('IM#24','Inter Milan',284,30,1289,648,418,  
223,2167,'AC23M');  
insert into clubs values ('AM#07','Atletico Madrid',320,24,547,296,1  
35,116,1004,'DS56M');  
insert into clubs values ('RM#46','Real Madrid',368,46,596,362,120,1  
14,1200,'ZZ01M');  
insert into clubs values ('MC#13','Man City',232,28,620,312,180,128,  
1988,'PG15M');  
insert into clubs values ('LV#72','Liverpool',297,29,1081,564,251,26  
6,1958,'JK11M');
```

CLUB_ID	CLUB_NAME	GOALS	TROPHIES	TOTAL_MATCHES_PLAYED	WIN	LOSE	DRAW	POINTS	MGR_ID
IM#24	Inter Milan	284	30	1289	648	418	223	2167	AC23M
AM#07	Atletico Madrid	320	24	547	296	135	116	1004	DS56M
RM#46	Real Madrid	368	46	596	362	120	114	1200	ZZ01M
MC#13	Man City	232	28	620	312	180	128	1988	PG15M
LV#72	Liverpool	297	29	1081	564	251	266	1958	JK11M

## TABLE PLAYERS

```
create table players(
  player_id      varchar(5) constraint plar_pk primary key,
  sale           varchar(8) constraint plar_nn not null,
  bid_amount     varchar(8),
  position       varchar(10),
  rating         number(5),
  skill_star     char(20),
  weak_foot      char(20),
  user_id        constraint plar_fk references users,
  club_id        constraint plar_fk2 references clubs
);
```

```
SQL> create table players(
  2  player_id      varchar(5)      constraint plar_pk primary key,
  3  sale           varchar(8)      constraint plar_nn not null,
  4  bid_amount     varchar(8),
  5  position       varchar(10),
  6  rating         number(5),
  7  skill_star     char(20),
  8  weak_foot      char(20),
  9  user_id        constraint plar_fk references users,
 10  club_id        constraint plar_fk2 references clubs
 11 );
```

Table created.

```
SQL> desc players;
```

Name	Null?	Type
PLAYER_ID	NOT NULL	VARCHAR2(5)
SALE	NOT NULL	VARCHAR2(8)
BID_AMOUNT		VARCHAR2(8)
POSITION		VARCHAR2(10)
RATING		NUMBER(5)
SKILL_STAR		CHAR(20)
WEAK_FOOT		CHAR(20)
USER_ID		VARCHAR2(5)
CLUB_ID		VARCHAR2(5)



```

insert into players values ('P#23','€105 M','€125 M','FC',8,'Cruyff
Turn','Left','U#23','LV#72');
insert into players values ('P#04','€245 M','€260 M','RW',9,'Gaucho
Snake','Right','U#07','LV#72');
insert into players values ('P#97','€130 M','€145 M','AM',7,'Puskas
Pullback','Right','U#99','MC#13');
insert into players values ('P#56','€110 M','€130 M','S',8,'Shotpowe
r','Left','U#34','MC#13');
insert into players values ('P#73','€220 M','€235 M','CM',9,'Boxtobo
x','Left','U#86','RM#46');
insert into players values ('P#24','€167 M','€180 M','CB',8,'Scissor
Kick','Left','U#49','RM#46');
insert into players values ('P#15','€160 M','€175 M','GK',7,'Jumping
Reach','Left','U#27','AM#07');
insert into players values ('P#09','€133 M','€140 M','DF',7,'Step-
over','Right','U#64','AM#07');
insert into players values ('P#70','€205 M','€210 M','CB',8,'Seal Dr
ibble','Right','U#14','IM#24');
insert into players values ('P#32','€170 M','€195 M','ST',9,'Jay-
Jay Okocha','Right','U#04','IM#24');

```

PLAYER_ID	SALE	BID_AMOUNT	POSITION	RATING	SKILL_STAR	WEAK_FOOT	USER_ID	CLUB_ID
P#23	€105 M	€125 M	FC	8	Cruyff Turn	Left	U#23	LV#72
P#04	€245 M	€260 M	RW	9	Gaucho Snake	Right	U#07	LV#72
P#97	€130 M	€145 M	AM	7	Puskas Pullback	Right	U#99	MC#13
P#56	€110 M	€130 M	S	8	Shotpower	Left	U#34	MC#13
P#73	€220 M	€235 M	CM	9	Boxtobox	Left	U#86	RM#46
P#24	€167 M	€180 M	CB	8	Scissor Kick	Left	U#49	RM#46
P#15	€160 M	€175 M	GK	7	Jumping Reach	Left	U#27	AM#07
P#09	€133 M	€140 M	DF	7	Step-over	Right	U#64	AM#07
P#70	€205 M	€210 M	CB	8	Seal Dribble	Right	U#14	IM#24
P#32	€170 M	€195 M	ST	9	Jay-Jay Okocha	Right	U#04	IM#24

# TABLE TRAININGS

```
alter session set nls_timestamp_format='HH12:Mi AM';
create table trainings(
    training_id    varchar(5)
                  constraint trng_pk primary key,
    tdate          date,
    ttime          timestamp(0),
    mgr_id         constraint trng_fk references managers not null
);
```

```
SQL> alter session set nls_timestamp_format='HH12:Mi AM';

Session altered.

SQL> create table trainings(
  2  training_id    varchar(5)    constraint trng_pk primary key,
  3  tdate          date,
  4  ttime          timestamp(0),
  5  mgr_id         constraint trng_fk    references managers not null
  6  );
```

Table created.

```
SQL> desc trainings;
Name                                Null?    Type
-----
TRAINING_ID                        NOT NULL VARCHAR2(5)
TDATE                              DATE
TTIME                              TIMESTAMP(0)
MGR_ID                             NOT NULL VARCHAR2(5)
```

```
insert into trainings values ('T#03',TO_DATE('20/01/2020', 'DD/MM/YY
YY'),to_timestamp('06:30 PM'),'AC23M');
insert into trainings values ('T#11',TO_DATE('03/02/2020', 'DD/MM/YY
YY'),to_timestamp('05:00 AM'),'DS56M');
insert into trainings values ('T#35',TO_DATE('31/01/2020', 'DD/MM/YY
YY'),to_timestamp('07:30 AM'),'ZZ01M');
insert into trainings values ('T#57',TO_DATE('17/03/2020', 'DD/MM/YY
YY'),to_timestamp('07:00 PM'),'PG15M');
select * from trainings;
```

TRAINING_ID	TDATE	TTIME	MGR_ID
T#03	20-JAN-20	06:30 PM	AC23M
T#11	03-FEB-20	05:00 AM	DS56M
T#35	31-JAN-20	07:30 AM	ZZ01M
T#57	17-MAR-20	07:00 PM	PG15M

# TABLE TRAINING\_FOR

```
create table training_for(  
    training_id      constraint trng_for_fk  
                      references trainings,  
    player_id        constraint trng_for_fk2  
                      references players,  
    constraint trng_for_pk primary key(training_id,player_id)  
);
```

```
SQL> create table training_for(  
2  training_id      constraint trng_for_fk  references trainings,  
3  player_id        constraint trng_for_fk2 references players,  
4  constraint trng_for_pk primary key(training_id,player_id)  
5  );
```

Table created.

```
SQL> desc training_for;
```

Name	Null?	Type
-----	-----	-----
TRAINING_ID	NOT NULL	VARCHAR2(5)
PLAYER_ID	NOT NULL	VARCHAR2(5)

```
insert into training_for values ('T#03','P#70');  
insert into training_for values ('T#11','P#09');  
insert into training_for values ('T#35','P#24');  
insert into training_for values ('T#57','P#97');
```

TRAINING_ID	PLAYER_ID
T#03	P#70
T#11	P#09
T#35	P#24
T#57	P#97

# TABLE BID

```
create table bid(  
    bid_id      varchar(5) constraint bid_pk primary key,  
    amount      varchar(8),  
    player_id   constraint bid_fkunn  
                references players unique not null,  
    mgr_id      constraint bid_by_fk2  
                references managers not null  
);
```

```
SQL> create table bid(  
2  bid_id      varchar(5)      constraint bid_pk primary key,  
3  amount      varchar(8),  
4  player_id   constraint bid_fkunn references players unique not null,  
5  mgr_id      constraint bid_by_fk2 references managers not null  
6  );
```

Table created.

```
SQL> desc bid;
```

Name	Null?	Type
BID_ID	NOT NULL	VARCHAR2(5)
AMOUNT		VARCHAR2(8)
PLAYER_ID	NOT NULL	VARCHAR2(5)
MGR_ID	NOT NULL	VARCHAR2(5)

```
insert into bid values('B#01', '€135 M', 'P#23', 'PG15M');  
insert into bid values('B#21', '€145 M', 'P#56', 'AC23M');  
insert into bid values('B#15', '€190 M', 'P#24', 'JK11M');  
insert into bid values('B#38', '€155 M', 'P#09', 'JK11M');  
insert into bid values('B#07', '€200 M', 'P#32', 'DS56M');
```

BID_ID	AMOUNT	PLAYER_ID	MGR_ID
B#01	€135 M	P#23	PG15M
B#21	€145 M	P#56	AC23M
B#15	€190 M	P#24	JK11M
B#38	€155 M	P#09	JK11M
B#07	€200 M	P#32	DS56M

# Review -3

## Football Club Management System

### SQL statements for implementation of functional requirements:

#### DELETE

1.Remove particular BID data from BID table

```
delete from bid
where bid_id='B#01';
```

2.Remove particular Training schedule from TRAININGS table

```
delete from trainings
where training_id='T#03';
```

3.Remove Manager of club “RM#46” detail from Manager table

```
delete from managers
where mgr_id in (
    select mgr_id from clubs
    where club_id='RM#46'
);
```

4.Remove particular player data from PLAYERS table

```
delete from players
where player_id='P#04';
```

## UPDATE

1.Change status of user to 'Banned' for given Player ID.

```
update users set status='Banned'
where user_id =(select user_id from players
                 where player_id='P#23');
```

2.Modify training schedule for particular Training

```
alter session set nls_timestamp_format='HH12:Mi AM';

update trainings
set tdate= TO_DATE('21/01/2020', 'DD/MM/YYYY'),ttime=to_timestamp('0
5:00 AM')
where training_id='T#11';
```

3.Update Points table of a Club.

```
update clubs
set total_matches_played =(total_matches_played+1),win =(win+1)
where club_id='IM#24';
```

4.Modify Bid amount for particular bid

```
update bid
set amount='U160 M'
where bid_id='B#21';
```

## SELECT

1. Display details of players using **NVL** function **OUTER JOIN** query

```
select name,nationality,skill_star,position,  
nvl(bid_amount,'not bid by any club')  
from users right outer join players  
on users.user_id=players.user_id;
```

NAME	NATIONALITY	SKILL_STAR	POSITION	NVL(BID_AMOUNT,'NOTBIDBYANYCLUB')
Joe Gomez	England	Cruyff Turn	FC	€125 M
David Silva	Spain	Puskas Pullback	AM	€145 M
Sergio Agüero	Argentina	Shotpower	S	€130 M
Toni Kroos	German	Boxtobox	CM	€235 M
Raphael Varane	French	Scissor Kick	CB	€180 M
Ivo Grbic	Croatian	Jumping Reach	GK	€175 M
Jose Grimenez	Uruguayan	Step-over	DF	€140 M
Diego Godin	Uruguayan	Seal Dribble	CB	€210 M
Alexis Sanchez	Chilean	Jay-Jay Okocha	ST	€195 M

2. Display details of club using **NULLIF** function

```
select club_name,nullif(win,0),nullif(lose,0),nullif(draw,0),nullif(  
goals,0),nullif(trophies,0) from clubs;
```

CLUB_NAME	NULLIF(WIN,0)	NULLIF(LOSE,0)	NULLIF(DRAW,0)	NULLIF(GOALS,0)	NULLIF(TROPHIES,0)
Inter Milan	649	418	223	284	30
Atletico Madrid	296	135	116	320	24
Real Madrid	362	120	114	368	46
Man City	312	180	128	232	28
Liverpool	564	251	266	297	29

3. Display Score board of club in ranking order using **ORDER BY** clause

```
select club_name,points
from clubs order by points desc;
```

CLUB_NAME	POINTS
Inter Milan	2167
Man City	1988
Liverpool	1958
Real Madrid	1200
Atletico Madrid	1004

4. Retrieve the manager data of particular club using **UNCORRELATED NESTED** query

```
select name,experience
from users natural join managers
where mgr_id in (select mgr_id from clubs
                 where club_id='MC#13');
```

NAME	EXPERIENCE
Pep Guardiola	4

5. Display all members of a team(Managers,Players) with their Role using **SET** operations.

```
select name,'Manager' as Role from users natural join managers
where mgr_id in (select mgr_id from clubs
                 where club_id='MC#13')
union
select name,'Player' as Role from users natural join players
where club_id='MC#13';
```

NAME	ROLE
David Silva	Player
Pep Guardiola	Manager
Sergio Aguero	Player



6. Display particular player's skill using **WHERE** clause

```
select player_id, rating, skill_star from players
where player_id='P#23';
```

PLAYER_ID	RATING	SKILL_STAR
P#23	8	Cruyff Turn

7. Display all training id with total players for that training is less than 2 using **GROUP BY** and **HAVING** clause.

```
select training_id, count(player_id)
from training_for
group by training_id
having count(player_id)<2;
```

TRAINING_ID	COUNT(PLAYER_ID)
T#03	1
T#11	1
T#35	1
T#57	1

8. Display goals and trophies of particular team using **CORRELATED NESTED** query (use manager table)

```
select club_name, goals, trophies from clubs
where exists (
    select mgr_id from managers
    where managers.mgr_id=clubs.mgr_id and club_id='IM#24'
);
```

CLUB_NAME	GOALS	TROPHIES
Inter Milan	284	30

## Implementation of two PL/SQL function involving cursor and two PL/SQL procedure involving cursor for the database under consideration :-

### PL/SQL FUNCTIONS

1. Implementation of a function which gets club id as parameter and returns manager of that club

#### Function definition

```
create or replace function managerofaclub(clb clubs.club_id%type) return users.name%type
  cursor mgr_det is select users.name from users,clubs,managers
                    where clubs.club_id=clb and clubs.mgr_id=managers.mgr_id
                    and managers.user_id=users.user_id;
  mgr_rec mgr_det%rowtype;

begin
  open mgr_det;
  fetch mgr_det into mgr_rec;
  return mgr_rec.name;
end;
```

#### Function execution

```
variable mgrname varchar2;

exec :mgrname:=managerofaclub('LV#72');

print mgrname;
```

#### Output

Function MANAGEROFACLUB compiled

PL/SQL procedure successfully completed.

MGRNAME

-----  
Jurgen Klopp

## 2. Implementation of function which gets player id and return active status of player

### Function definition

```
create or replace function players_status(plr_id players.player_id%type) return users.status%type is
  cursor      plyr_crs is select player_id,status
                        from players,users
                        where players.user_id=users.user_id;
  plyr_rec     plyr_crs%rowtype;

begin
  open plyr_crs;
  loop
    fetch plyr_crs into plyr_rec;
    exit when plyr_crs%notfound;
    if plr_id=plyr_rec.player_id then
      return plyr_rec.status;
    end if;
  end loop;
  close plyr_crs;
end;
```

### Function execution

```
variable status_var varchar2;

exec :status_var:= players_status('P#23');

print status_var;
```

### Output

Function PLAYERS\_STATUS compiled

PL/SQL procedure successfully completed.

STATUS\_VAR

-----  
Active

## PL/SQL PROCEDURES

### 1.Implementation of procedure which gets club id as parameter and prints players of that club

#### Procedure definition

```
create or replace procedure playersofclub (clb clubs.club_id%type) is
  cursor plr_crs is select name from players,users
                    where players.club_id=clb and players.user_id=users.user_id;
  plr_rec plr_crs%rowtype;

begin
  open plr_crs;
  dbms_output.put_line('Players of club '||clb||' are :');
  loop
    fetch plr_crs into plr_rec;
    exit when plr_crs%notfound;
    dbms_output.put_line(plr_rec.name);
  end loop;
  close plr_crs;
end;
```

#### Procedure execution

```
exec playersofclub('RM#46');
```

#### output

Procedure PLAYERSOFCLUB compiled

Players of club RM#46 are :

Toni Kroos

Raphhael Varane

PL/SQL procedure successfully completed.

## 2. Implementation of procedure which checks login credentials

### Procedure definition

```
create or replace procedure loginacc(id users.user_id%type,pwd users.password%type) is
    cursor    user_det is select user_id,password from users;
    user_rec   user_det%rowtype;
    datanotfound number := 0;

begin
    open user_det;
    loop
        fetch user_det into user_rec;
        exit when user_det%notfound;
        if id=user_rec.user_id then
            datanotfound:=1;
            if pwd=user_rec.password then
                dbms_output.put_line('Welcome '||id||' You are successfully logged in');
            else
                dbms_output.put_line('Invalid User id or password');
            end if;
        end if;
    end loop;
    if datanotfound=0 then
        dbms_output.put_line('No such user found');
    end if;
    close user_det;
end;
```

### Procedure execution

```
exec loginacc('U#23','password1');
```

### Output

Procedure LOGINACC compiled

Welcome U#23 You are successfully logged in

PL/SQL procedure successfully completed.

## Implementation of three business rules appropriate for the database under consideration and using trigger

### TRIGGER 1

#### Rules

If user id of user is modified then user id has to be changed in players table for players and managers table for managers.

#### Code

```
create or replace trigger useridupdate

  after update of user_id on users
  for each row
  begin

    update players
    set user_id = :new.user_id
    where user_id= :old.user_id;

    update managers
    set user_id = :new.user_id
    where user_id = :old.user_id;

  end;

-----

update users
set user_id='U#01'
where user_id='U#23';
```

#### Output

```
Trigger USERIDUPDATE compiled

1 row updated.
```

## TRIGGER 2

### Rules

Clubs will get 3 points for an each win, 1 point for a draw and will not get any points for loss.

### Code

```
create or replace trigger pointtable
after insert or update of total_matches_played on clubs
begin
    update clubs
    set points=win * 3+draw;
end;

-----
update clubs
set total_matches_played =(total_matches_played+1),win =(win+1)
where club_id='IM#24';
```

### Output

Trigger POINTTABLE compiled

1 row updated.

## TRIGGER 3

### Rules

If a training is cancelled Player have to be notified. So that training have to be removed from training\_for table.

### Code

```
create or replace trigger tranining_schedule
    before delete on trainings
    for each row
    begin
        delete from training_for
        where training_id= :old.training_id;
    end;
-----
delete from trainings
where training_id='T#11';
```

### Output

```
Trigger TRANINING_SCHEDULE compiled
```

```
1 row deleted.
```