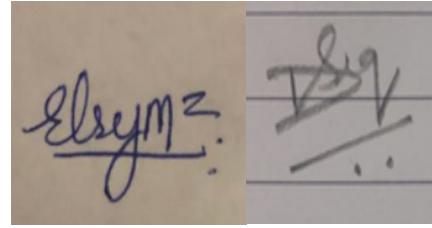


**HONOR CODE**

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

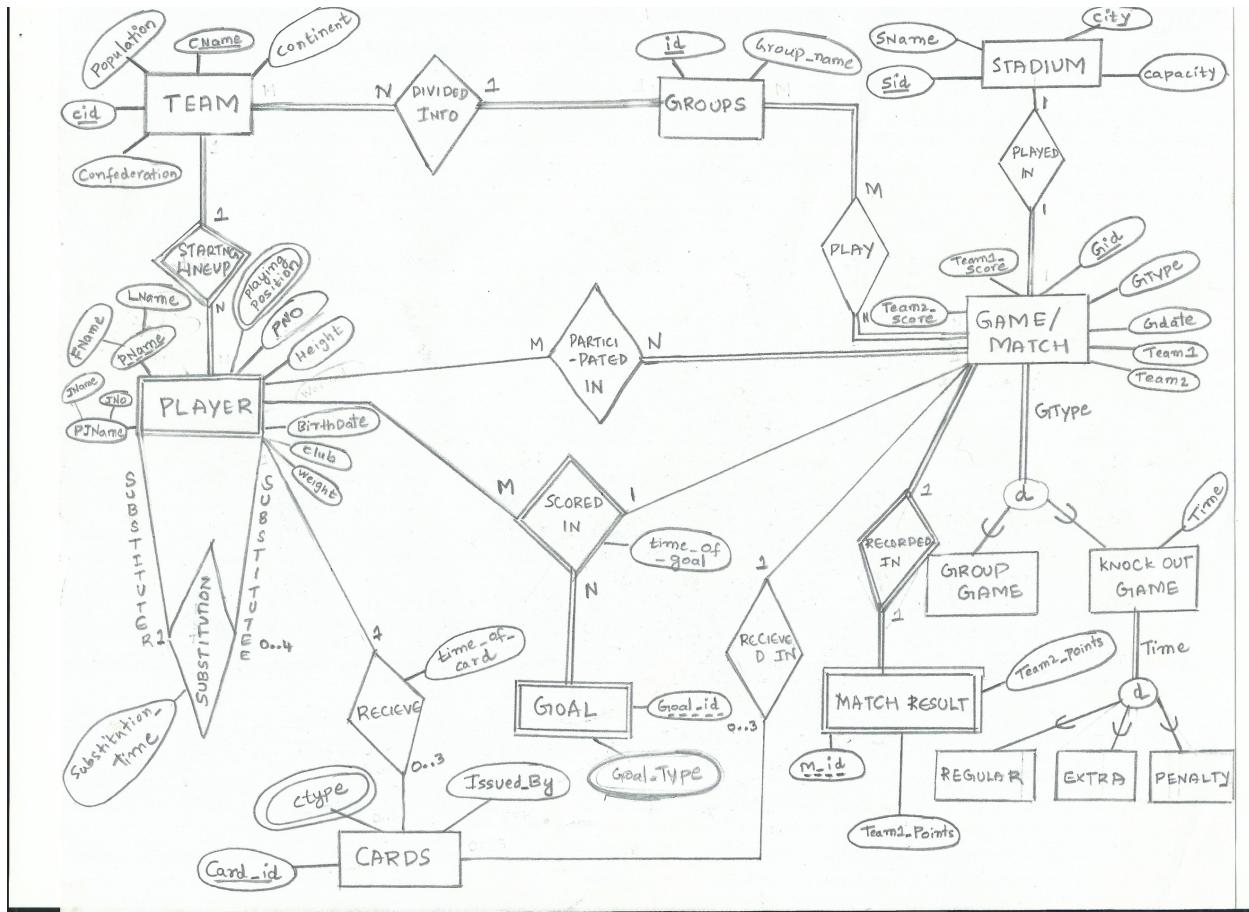
**TEAM MEMBERS:**

---

ELSY FERNANDES  
MAV ID: 1001602253

---

DIANA NEVILLE MASCARENHAS  
MAV ID: 1001550719

EER Diagram

### Design Choices

This Project involves creating the following:

1. Conceptual database design using ER/EER model
2. Converting EER model to a relational schema using rules for ER/EER-to-relational mapping
3. Designing a relational database.

**This project mainly involves storing information of a SOCCER game in the Database.**

Our EER design starts with an entity type ‘TEAM’. TEAM basically involves all the 32 countries which participate in this tournament. ‘TEAM’ has a relationship with entity type ‘GROUP’ and a ‘PLAYER’. Every TEAM participates in ‘STARTING LINEUP’ relationship atleast once. (Every TEAM has atleast one player) So participation of TEAM in a STARTING LINEUP relationship is Total.

We have ‘STARTING LINEUP’ Relationship between a TEAM and a PLAYER because the Original Team has 23 players, but out of 23 players, only 11 players participate in a match, at any given time. So the number of players in PLAYER entity type participating in a match, are 11/23 and remaining 12/23 players are the substitutes. We have an attribute ‘Substitute\_PNo’ in PLAYER table to determine which player out of 11 players substituted by which player (out of 12). Out of those 11 players, they can substitute upto 3 players during the 90-minute game and 1-more player if the game is KNOCKOUT.

Entity PLAYER is a weak entity and its identifying relationship to the owner ‘STARTING LINEUP’ is also weak. Attribute ‘PName’ is a partial key as shown in the EER diagram. PLAYER has a self-referencing relationship ‘SUBSTITUTION’, where we have a substitute, and a substitute, both belong to an entity type PLAYER. Each PLAYER can substitute upto 0 or 3 players if its 90-minute game and 1-more player if the game is knockout. So, they can substitute upto 0 and 3 (or 4). Hence the cardinality ratio is 1 to 0..4. We have a table for PLAYER in our database along with the substitution info.

Every TEAM is divided into atleast one GROUPS. So, participation of TEAM in DIVIDED BY Relationship is Total. Every group has atleast one team associated with it. So, the participation of GROUPS in a DIVIDED BY relationship is Total.

TEAM has its own attributes as mentioned in the project description. It has Cid as a surrogate key, we have set it as a PRIMARY KEY in our design choices. And CName is also unique. So, we have set it as UNIQUE Key in our database. We have a ‘TEAM’ table in our database for the entity type ‘TEAM’.

Cardinality ratio between TEAM and GROUPS is N:1(Each TEAM divided into atmost One GROUPS. One GROUP has many numbers of TEAMS). Cardinality ratio between TEAM and PLAYER is 1: N (Each PLAYER has atmost 1 team.one TEAM has many PLAYERS).

We have a ‘GROUPS’ Table in the database and also as an entity in the EER Diagram. The reason we created this Entity type, is to have a list of all groups (groups belong to Group game (A, B, C, D, E, F, G,H) and a knockout game(X,Q,S,L,T) so that we can associate a country to each group in our database.

We have a ‘GROUP\_COUNTRY’ table in the database which references GROUPS and COUNTRY Table to map which country belongs to which group initially, before beginning with the game. This is an extra table in the database, where we have not used an entity type for this. This operation has to be done internally.

‘GROUPS’ entity type has a relationship with ‘GAME/MATCH’ entity type. Each GROUP play many numbers of GAME/MATCH. Each GAME/MATCH is played by many GROUPS. Cardinality ratio is M: N. Participation constraint is total between GROUPS and MATCH and vice versa.

Entity type MATCH has two subclasses. GROUP GAME and KNOCKOUT GAME. This specialization has a **disjoint** constraint. Here both subclasses, in the specialization have their membership condition on the same attribute of the superclass, MATCH. This specialization is “**Attribute defined**” specialization. Here attribute used for the specialization is GType.

We have a total participation between a MATCH/GAME and its subclasses that is because Every entity MATCH/GAME must be a member of atleast one subclasses. So it is disjoint, Total.

KNOCK\_OUT GAME has a local attribute ‘Time’. Based on attribute Time, we have specialized the KNOCKOUT GAME to REGULAR, EXTRA, PENALTY. This is also an “Attribute defined” specialization. We have used a **disjoint** constraint here because an entity of a superclass can be a member of at most one of the subclasses of the specialization. So, this is disjoint, Total participation.

We have several attributes for GAME/MATCH entity as mentioned in the project description, but we have an attribute Gid as a primary key in our database. We have tables for GAME, GROUP\_GAME, KNOCKOUT\_GAME, REGULAR, EXTRA, PENALTY in our database.

Entity type MATCH has a relationship with entity type STADIUM. Every MATCH is played in atleast one STADIUM. So, participation of MATCH in a PLAYED IN relationship is Total. But the participation of STADIUM in PLAYED IN relationship is Partial. Cardinality ratio between STADIUM and GAME/MATCH is 1:1. Each STADIUM has atmost one MATCH/GAME. MATCH/GAME has atmost one STADIUM. Entity STADIUM has several attributes mentioned in the requirement. Attribute Sid is set as a primary key. We also have a table STADIUM in the database.

After every MATCH (weather it is a KNOCKOUT or GROUPGAME), GROUP\_COUNTRY table mentioned above gets updated by updating the new group to each country. For example, USA belongs to GROUP A and it plays a groupgame with another team. If USA wins the match, Groupname for USA in ‘GROUP\_TABLE’ gets updated with the next valid group (USA goes to the next round). Applies to all the country which participates in a MATCH.

Entity type MATCH has a ‘RECORDED IN’ Relationship with ‘MATCH\_RESULT’. Entity type MATCH participation in a ‘RECORDED IN’ relationship is Total. ‘MATCH\_RESULT’ doesn’t exist without MATCH. So, MATCH\_RESULT is a weak entity and the relationship ‘RECORDED IN’ is an identifying relationship. We have defined attributes in MATCH\_RESULT. Different attributes are M\_id which is a partial key because it is a weak entity. Team1\_Points, Team2\_points are the other two attributes.

MATCH\_RESULTS table has information on winner team of the tournament, along with the points scored by Team1 and Team2. We used Attribute ‘Team1\_Points’ and ‘Team2\_Points’. As mentioned in the requirement ‘GROUP\_GAME’ has a draw match. If in any match, winning team gets 2 points and a losing team doesn’t get any point. But if it’s a draw match then both the teams get 1 point each. So, it’s easier to determine a winner with points specially in these cases.

Entity type CARD related to a PLAYER and a GAME. We need to identify which PLAYER received a CARD in which MATCH/GAME. So this relationship is necessary in EER diagram. The participation of PLAYER in RECEIVED relationship is partial because it is not necessary that every player has to receive the cards. The participation of CARD in received relationship is partial because it’s not necessary that every card is received by atleast one player. We do not consider CARD as a weak entity because CARDS can exist without soccer PLAYER and a Soccer MATCH. Cardinality ratio between a PLAYER and CARDS is 1 to 0..3 .because at a time Player can receive 0 cards or 1 card or 2

card or 3 card. There are chances that 3 cards are received by the same PLAYER at a time. Cardinality ratio between MATCH and CARDS is 1 to 0..3 .

We have a table CARDS in a database to store the card info and CARDS\_MATCH\_PLAYER to ensure which card is received by which player in which match and type of the card.

Entity type GOAL has a ternary relationship with a PLAYER and a MATCH/GAME. GOAL is a weak entity because it cannot exist without its owner entities PLAYER and a MATCH. So, we have Goal\_id set to be a partial key. We also have time\_of\_goal recorded.

GOAL is a weak entity to the owner hence its identifying relationship becomes weak. Cardinality ratio between MATCH:PLAYER: GOAL is 1:M: N because one match can have many players and many goals.

We also have an additional table GOAL\_MATCH\_PLAYER in our database to keep the track of which player scored goal in which match.

We have another table PLAYER\_MATCH which tells us total number of goals received by a player and total cards received by a player and in which match.

We also have a relationship between a PLAYER and MATCH. Cardinality ratio is M:N, it's because each player (Only the 11 players because we have already created a starting lineup) plays many matches. Each Match is played by many players. Participation of PLAYER in a 'PARTICIPATED' relationship is partial. It's because only 11 players out of 23 play the match. But participation of match in participated in relation is total.

### Relational Schema

## RELATIONAL SCHEMA

## TEAM

cId	CName	Continent	confederation	Population
-----	-------	-----------	---------------	------------

## PLAYER

Pno	CName	Fname	LName	Jname	Jno	BirthDate	Club	Height	Weight	Substitute-Pno	Substitution_Time	Ppno
-----	-------	-------	-------	-------	-----	-----------	------	--------	--------	----------------	-------------------	------

## PLAYING POSITION

Ppno	Playingposition
------	-----------------

## GROUPS

id	Group_name
----	------------

## GROUP-COUNTRY

Gno	Ename	Id
-----	-------	----

## STADIUM

Sid	Sname	City	Capacity
-----	-------	------	----------

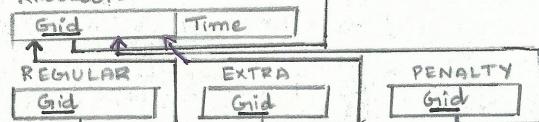
## GAME / MATCH

Gid	Gtype	Gidate	Stadium_id	Team1	Team2	Team1_Score	Team2_Score
-----	-------	--------	------------	-------	-------	-------------	-------------

## GROUP-GAME

Gid
-----

## KNOCKOUT-GAME



## MATCH\_RESULTS

M_id	Gid	Team1-points	Team2-points
------	-----	--------------	--------------

## CARDS\_R

C_id	CType
------	-------

## GOALS\_R

Goal_id	Goal_Type
---------	-----------

## PLAYER-MATCH

Player-match_id	Pno	Gid	Total_goals_Scored	Total_cards_Received
-----------------	-----	-----	--------------------	----------------------

## CARDS\_MATCH\_R

Card_id	Id	Time_of_card	Issued_By
---------	----	--------------	-----------

## CARD-MATCH-PLAYER

Id	Card_id	Gid	Pno
----	---------	-----	-----

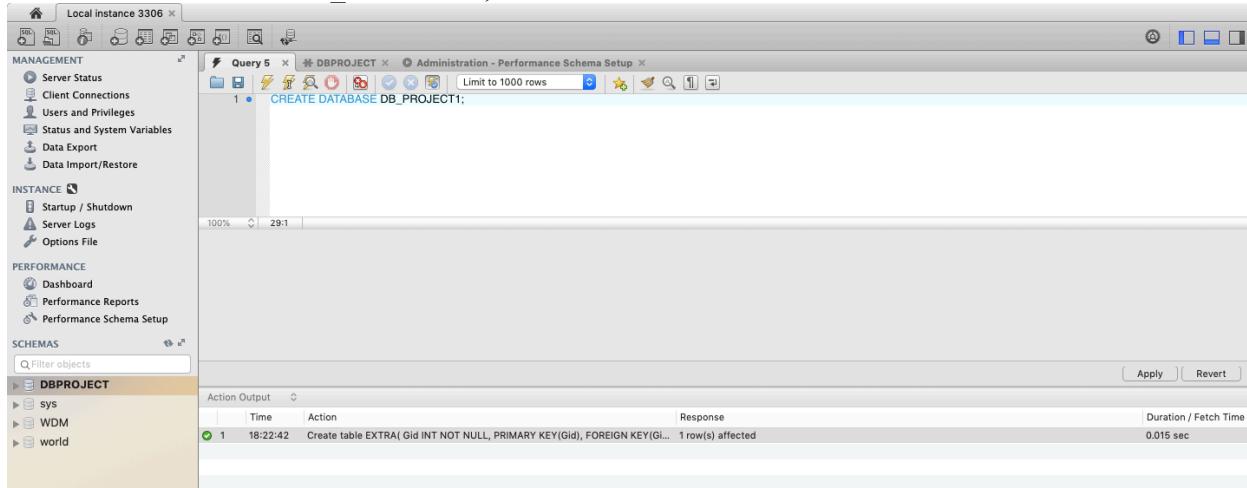
## GOAL-MATCH-PLAYER

ID	goal_ID	Pno	GID_at_goal	Time_of_goal
----	---------	-----	-------------	--------------

SQL Queries: -

## Create a Database

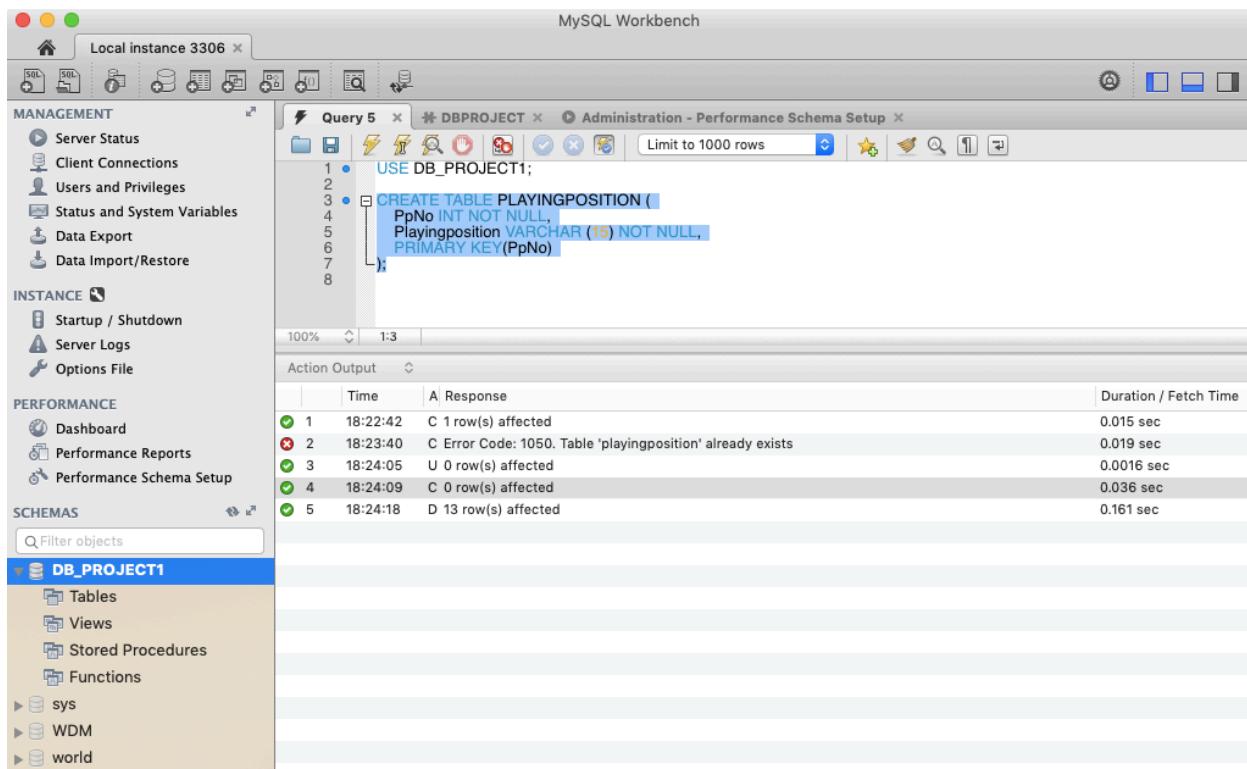
```
CREATE DATABASE DB_PROJECT1;
```



1. Create a table ‘PLAYINGPOSITION’ (We created this table first because this is used as a referencing table)

Since playingposition is a multivalued attribute to an entity ‘PLAYER’ in our EER diagram. Created a table ‘PLAYINGPOSITION’, where ‘playingposition’ is a multivalued attribute.

```
CREATE TABLE PLAYINGPOSITION (
    PpNo INT NOT NULL,
    Playingposition VARCHAR (15) NOT NULL,
    PRIMARY KEY(PpNo)
);
```



## 2. Create a table ‘Groups’

This table was created to differentiate all the groups present in the match (groups belong to Groupgame and knockoutgame). This table is used as reference to ‘GROUP\_COUNTRY’ table in our database.

```

CREATE TABLE GROUPS (
id INT NOT NULL,
Group_name VARCHAR (15) NOT NULL,
PRIMARY KEY (id)
);

```

The screenshot shows the MySQL Workbench interface. On the left, the sidebar includes sections for MANAGEMENT (Server Status, Client Connections, Users and Privileges, etc.), INSTANCE (Startup / Shutdown, Server Logs, Options File), PERFORMANCE (Dashboard, Performance Reports, Performance Schema Setup), and SCHEMAS (DB\_PROJECT1, Tables, PLAYINGPOSITION, Views, Stored Procedures, Functions). The main area displays a query editor titled 'Query 5' with the following SQL code:

```
1 USE DB_PROJECT1;
2
3 CREATE TABLE GROUPS (
4     id INT NOT NULL,
5     Group_name VARCHAR(15) NOT NULL,
6     PRIMARY KEY (id)
7 );
```

The status bar at the bottom indicates 'Action Output' with a timeline from 100% to 1:3. Below the timeline is a table showing the execution history:

	Time	Action	Response	Duration / Fetch Time
1	18:22:42	C	1 row(s) affected	0.015 sec
2	18:23:40	C	Error Code: 1050. Table 'playingposition' already exists	0.019 sec
3	18:24:05	U	0 row(s) affected	0.0016 sec
4	18:24:09	C	0 row(s) affected	0.036 sec
5	18:24:18	D	13 row(s) affected	0.161 sec
6	18:25:10	C	0 row(s) affected	0.022 sec

### 3. Create a Table 'STADIUM'

Create table STADIUM (  
Sid INT NOT NULL,  
SName VARCHAR (15),  
City VARCHAR (15),  
Capacity int,  
PRIMARY KEY(Sid)  
);

The screenshot shows the MySQL Workbench interface. On the left, the sidebar displays the DB\_PROJECT1 schema, which contains tables like GROUPS, PLAYINGPOSITION, and STADIUM. The main area shows a query editor with the following SQL code:

```

USE DB_PROJECT1;
Create table STADIUM (
  Sid INT NOT NULL,
  SName VARCHAR(15),
  City VARCHAR(15),
  Capacity int,
  PRIMARY KEY(Sid)
);
    
```

Below the query editor, the Action Output pane shows the results of the execution:

	Time	A Response	Duration / Fetch Time
1	18:22:42	C 1 row(s) affected	0.015 sec
2	18:23:40	C Error Code: 1050. Table 'playingposition' already exists	0.019 sec
3	18:24:05	U 0 row(s) affected	0.0016 sec
4	18:24:09	C 0 row(s) affected	0.036 sec
5	18:24:18	D 13 row(s) affected	0.161 sec
6	18:25:10	C 0 row(s) affected	0.022 sec
7	18:26:01	C 0 row(s) affected	0.026 sec

#### 4.Create a table GAME/MATCH

This table has an attribute ‘Gid’ (surrogate key) as a primary key and attribute ‘stadium\_id’ is a foreign key which is references table ‘STADIUM’.

```

Create table GAME (
Gid INT NOT NULL,
GType VARCHAR (15),
Gdate INT,
Stadium_id INT NOT NULL,
Team1 VARCHAR (15),
Team2 VARCHAR (15),
Team1_Score INT,
Team2_Score INT,
PRIMARY KEY(Gid),
FOREIGN KEY(Stadium_id) REFERENCES STADIUM(Sid)
);
    
```

The screenshot shows the MySQL Workbench interface. The left sidebar contains navigation panels for MANAGEMENT, INSTANCE, PERFORMANCE, and SCHEMAS. The SCHEMAS panel shows a database named 'DB\_PROJECT1' with tables 'GAME', 'GROUPS', 'PLAYINGPOSITION', 'STADIUM', 'Views', 'Stored Procedures', and 'Functions'. The right side has a 'Query 5' editor with the following SQL code:

```

1 USE DB_PROJECT1;
2
3 Create table GAME (
4     Gid INT NOT NULL,
5     GType VARCHAR(15),
6     Gdate INT,
7     Stadium_id INT NOT NULL,
8     Team1 VARCHAR(15),
9     Team2 VARCHAR(15),
10    Team1_Score INT,
11    Team2_Score INT,
12    PRIMARY KEY(Gid),
13    FOREIGN KEY(Stadium_id) REFERENCES STADIUM(Sid)
14 );
15
16
17
18

```

The 'Action Output' pane below the query editor displays the execution log:

	Time	A Response	Duration / Fetch Time
1	18:22:42	C 1 row(s) affected	0.015 sec
2	18:23:40	C Error Code: 1050. Table 'playingposition' already exists	0.019 sec
3	18:24:05	U 0 row(s) affected	0.0016 sec
4	18:24:09	C 0 row(s) affected	0.036 sec
5	18:24:18	D 13 row(s) affected	0.161 sec
6	18:25:10	C 0 row(s) affected	0.022 sec
7	18:26:01	C 0 row(s) affected	0.026 sec
8	18:27:00	C 0 row(s) affected	0.034 sec

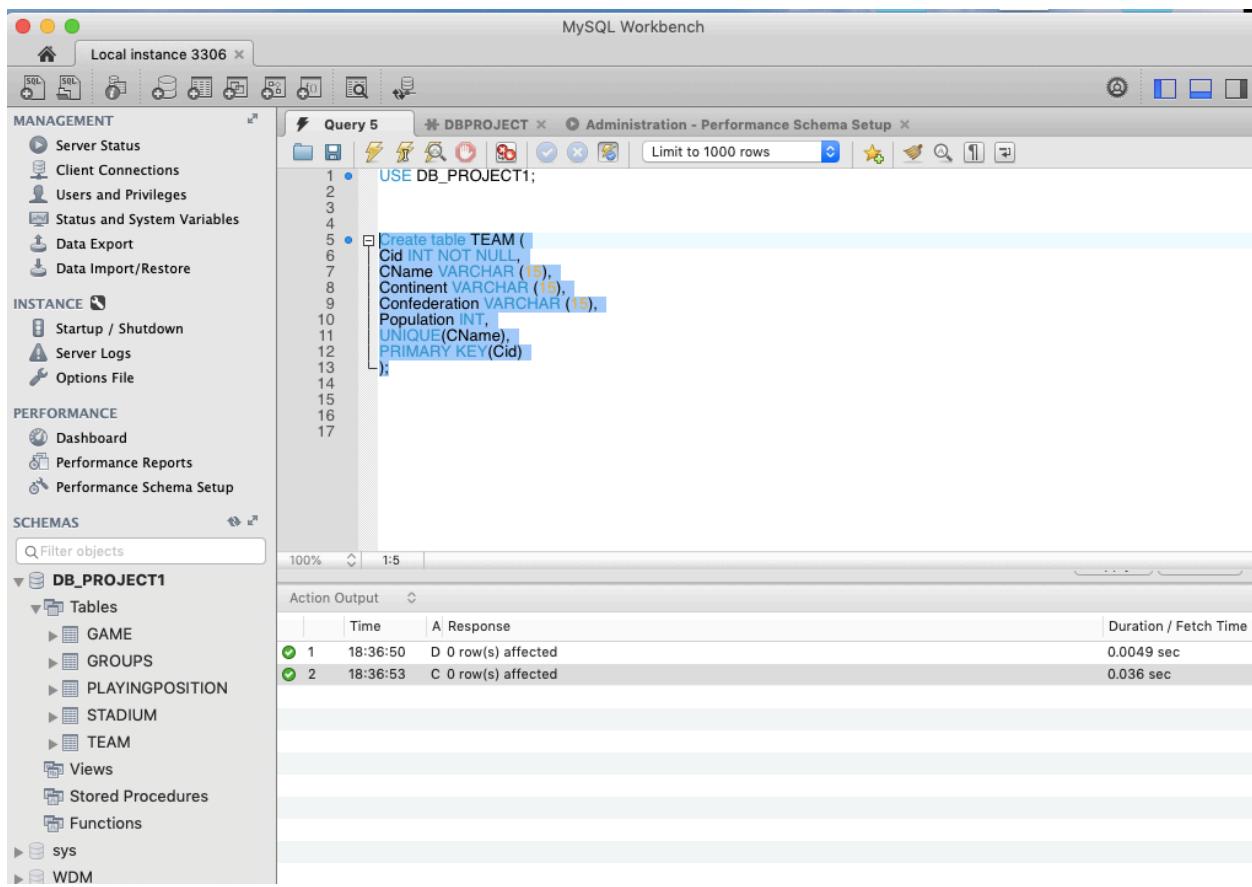
## 5. CREATE TABLE ‘TEAM’

This table has an attribute ‘cid’(surrogate key) as a primary key and attribute ‘CName’ is set as unique

```

Create table TEAM (
    Cid INT NOT NULL,
    CName VARCHAR(15),
    Continent VARCHAR(15),
    Confederation VARCHAR(15),
    Population INT,
    Gid INT NOT NULL,
    UNIQUE(CName),
    PRIMARY KEY(Cid),
);

```



The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Management):** MANAGEMENT, INSTANCE, PERFORMANCE, SCHEMAS.
- SCHEMAS:** DB\_PROJECT1 selected, showing Tables (GAME, GROUPS, PLAYINGPOSITION, STADIUM, TEAM), Views, Stored Procedures, Functions, sys, WDM.
- Central Panel (Query Editor):** Query 5, DB\_PROJECT1 selected, Administration - Performance Schema Setup tab. The query is:
 

```

1 USE DB_PROJECT1;
2
3
4
5 Create table TEAM (
6     Cid INT NOT NULL,
7     CName VARCHAR (15),
8     Continent VARCHAR (15),
9     Confederation VARCHAR (15),
10    Population INT,
11    UNIQUE(CName),
12    PRIMARY KEY(Cid)
13
14
15
16
17
      
```
- Action Output:** Shows two actions:
 

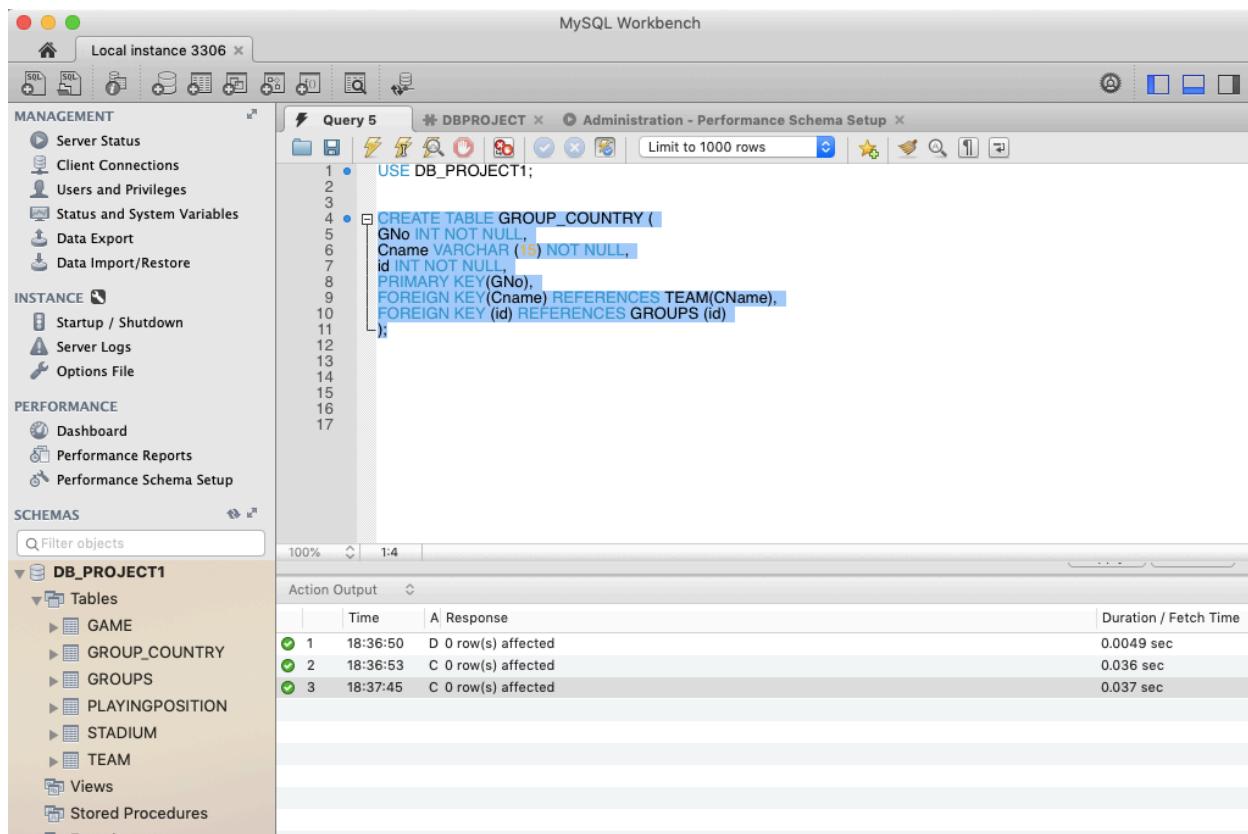
	Time	A Response	Duration / Fetch Time
1	18:36:50	D 0 row(s) affected	0.0049 sec
2	18:36:53	C 0 row(s) affected	0.036 sec

## 6. CREATE TABLE ‘GROUP\_COUNTRY’

This table is used to map country and groups. This table helps identify which country belongs to which group. This table has an attribute ‘GNo’ as a primary key and attribute ‘id’ as a foreign key which provides the reference to ‘GROUPS’ table. attribute ‘Cname’ provides reference to ‘TEAM’ Table.

```

CREATE TABLE GROUP_COUNTRY (
GNo INT NOT NULL,
Cname VARCHAR (15) NOT NULL,
id INT NOT NULL,
PRIMARY KEY(GNo),
FOREIGN KEY(Cname) REFERENCES TEAM(CName),
FOREIGN KEY (id) REFERENCES GROUPS (id)
);
      
```



The screenshot shows the MySQL Workbench interface with the following details:

- Left Sidebar (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Sidebar (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Sidebar (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Schemas:** DB\_PROJECT1 (selected), containing Tables (GAME, GROUP\_COUNTRY, GROUPS, PLAYINGPOSITION, STADIUM, TEAM) and Views, Stored Procedures.
- Central Area (Query Editor):** Title: Query 5, Tab: DBPROJECT, Sub-tab: Administration - Performance Schema Setup. The query is:
 

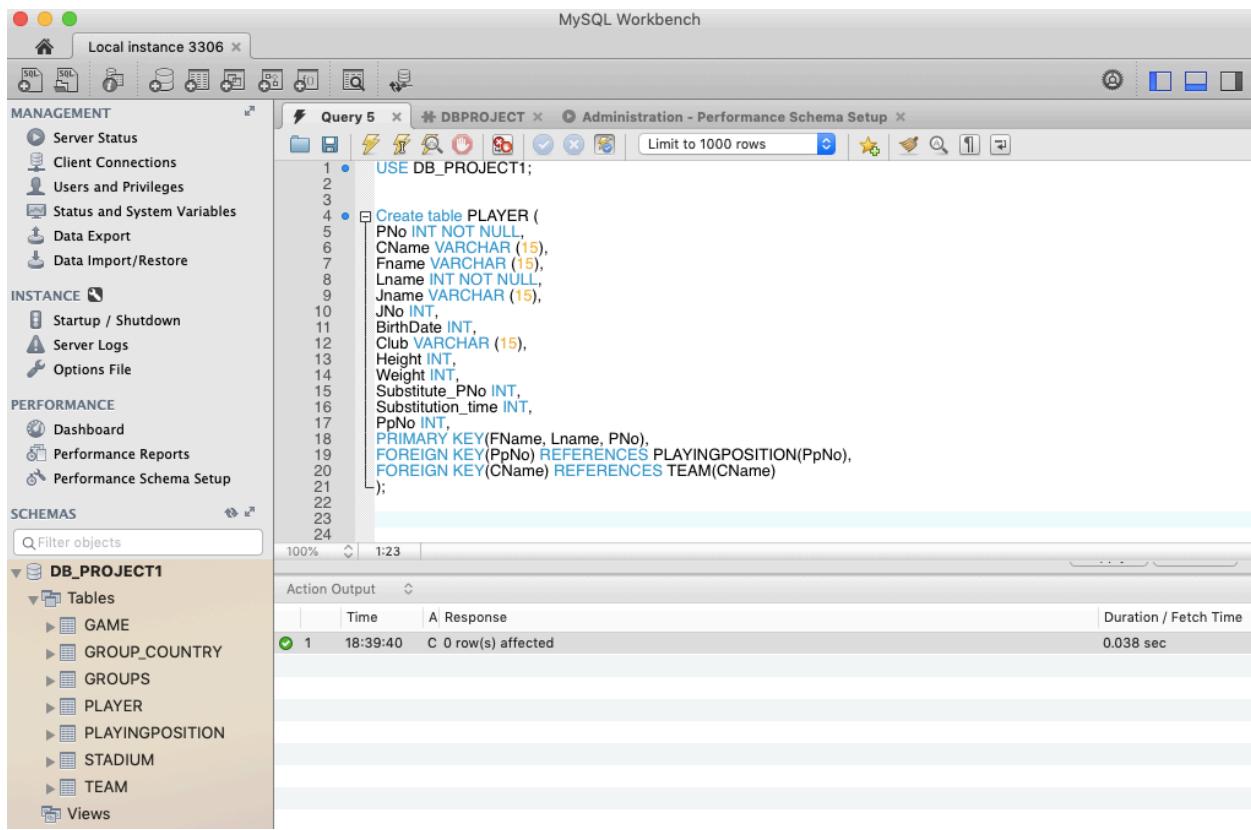
```
1 USE DB_PROJECT1;
2
3
4 CREATE TABLE GROUP_COUNTRY (
5   GNo INT NOT NULL,
6   Cname VARCHAR (15) NOT NULL,
7   id INT NOT NULL,
8   PRIMARY KEY(GNo),
9   FOREIGN KEY(Cname) REFERENCES TEAM(CName),
10  FOREIGN KEY(id) REFERENCES GROUPS (id)
11 );
12
13
14
15
16
17 
```
- Action Output:** Shows three log entries:
 

	Time	A Response	Duration / Fetch Time
1	18:36:50	D 0 row(s) affected	0.0049 sec
2	18:36:53	C 0 row(s) affected	0.036 sec
3	18:37:45	C 0 row(s) affected	0.037 sec

## 7. CREATE TABLE 'PLAYER'

This table has an PName (Fname and Lname) as a partial key has 'PpNo' as a foreign key referencing the 'PLAYINGPOSITION' table, attribute CName as a foreign key references TEAM.

```
Create table PLAYER (
PNo INT NOT NULL,
CName VARCHAR (15),
Fname VARCHAR (15),
Lname INT NOT NULL,
Jname VARCHAR (15),
JNo INT,
BirthDate INT,
Club VARCHAR (15),
Height INT,
Weight INT,
Substitute_PNo INT,
Substitution_time INT,
PpNo INT,
PRIMARY KEY(FName, Lname, PNo),
FOREIGN KEY(PpNo) REFERENCES PLAYINGPOSITION(PpNo),
FOREIGN KEY(CName) REFERENCES TEAM(CName)
);
```



The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Panel (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Panel (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Left Panel (Schemas):** DB\_PROJECT1 (selected), Tables, Views.
- Central Area (Query Editor):**

```

1 USE DB_PROJECT1;
2
3
4 Create table PLAYER (
5     PNo INT NOT NULL,
6     CName VARCHAR (15),
7     Fname VARCHAR (15),
8     Lname INT NOT NULL,
9     Jname VARCHAR (15),
10    JNo INT,
11    BirthDate INT,
12    Club VARCHAR (15),
13    Height INT,
14    Weight INT,
15    Substitute_PNo INT,
16    Substitution_time INT,
17    PpNo INT,
18    PRIMARY KEY(FName, Lname, PNo),
19    FOREIGN KEY(PpNo) REFERENCES PLAYINGPOSITION(PpNo),
20    FOREIGN KEY(CName) REFERENCES TEAM(CName)
21 );
22
23
24

```
- Bottom Area (Action Output):**

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec

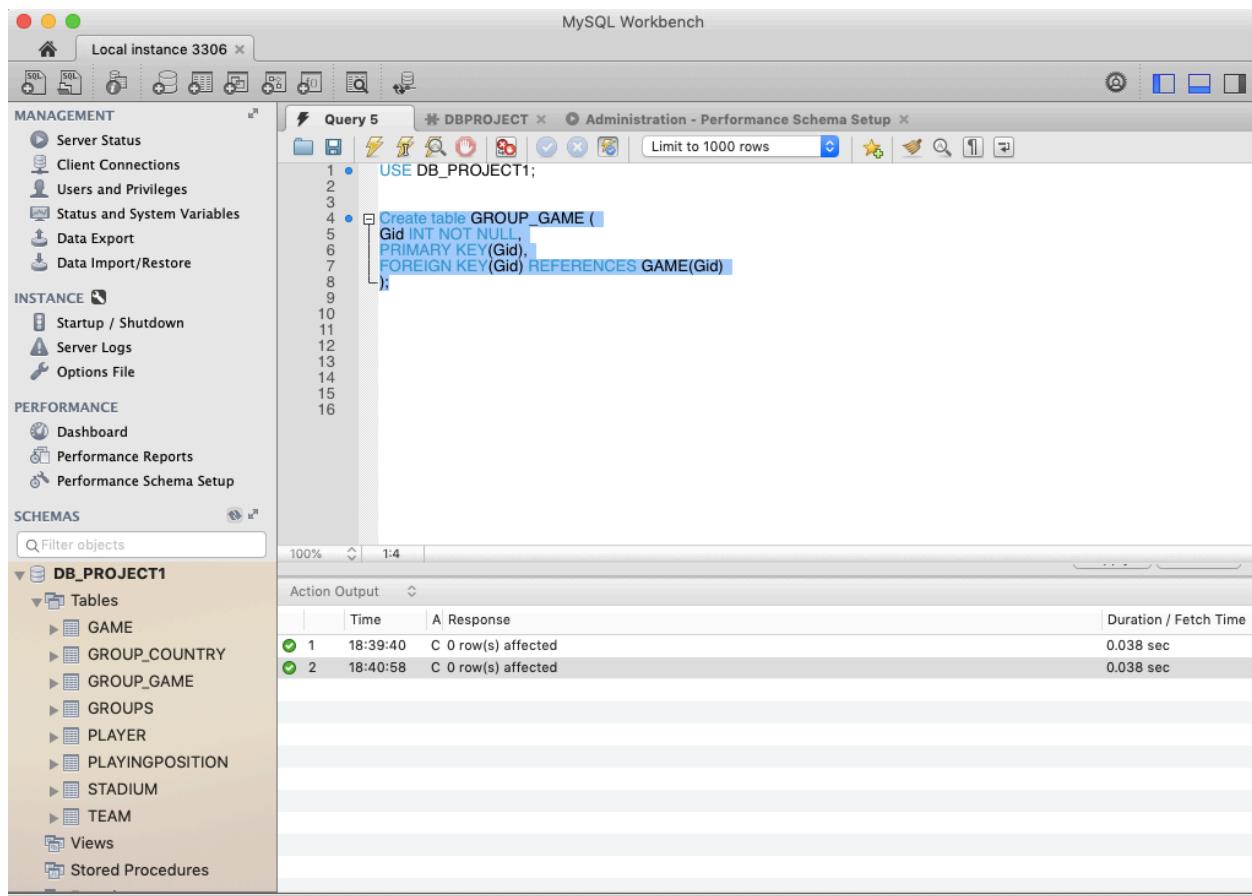
### 8.CREATE TABLE ‘GROUP\_GAME’

This table has an attribute ‘Gid’ set as a foreign key referencing the ‘GAME’ table. This Entity type is a subclass of a ‘GAME’ Entity type.

```

Create table GROUP_GAME (
Gid INT NOT NULL,
PRIMARY KEY(Gid),
FOREIGN KEY(Gid) REFERENCES GAME(Gid)
);

```



## 9.CREATE TABLE ‘KNOCKOUT\_GAME’

This table has an attribute ‘Gid’ set as a foreign key referencing the ‘GAME’ table. This entity type is a subclass of a ‘GAME’ Entity type.

```

Create table KNOCK_OUT (
Gid INT NOT NULL,
Knockout_time INT,
PRIMARY KEY(Gid),
FOREIGN KEY(Gid) REFERENCES GAME(Gid)
);

```

The screenshot shows the MySQL Workbench interface. On the left is the 'Management' sidebar with options like Server Status, Client Connections, and Data Export. Below it is the 'Schemas' sidebar showing the 'DB\_PROJECT1' schema with tables such as GAME, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, PLAYER, PLAYINGPOSITION, STADIUM, TEAM, and Views. The main area is titled 'Query 5' and contains the following SQL code:

```

1 USE DB_PROJECT1;
2
3
4 Create table KNOCK_OUT (
5   Gid INT NOT NULL,
6   Knockout_time INT,
7   PRIMARY KEY(Gid),
8   FOREIGN KEY(Gid) REFERENCES GAME(Gid)
9 );
10
11
12
13
14
15
16
17
18
19

```

Below the code is an 'Action Output' table showing three rows of execution results:

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec

## 10.CREATE TABLE ‘REGULAR’

This table has an attribute ‘Gid’ set as a foreign key referencing the ‘KNOCK\_OUT’ table. This entity type is a subclass of a ‘KNOCK\_OUT’ Entity type.

```

Create table REGULAR (
Gid INT NOT NULL,
PRIMARY KEY(Gid),
FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
);

```

The screenshot shows the MySQL Workbench interface with the following details:

- Top Bar:** Local instance 3306, MySQL Workbench.
- Left Sidebar (Management):**
  - Server Status
  - Client Connections
  - Users and Privileges
  - Status and System Variables
  - Data Export
  - Data Import/Restore
- Left Sidebar (INSTANCE):**
  - Startup / Shutdown
  - Server Logs
  - Options File
- Left Sidebar (PERFORMANCE):**
  - Dashboard
  - Performance Reports
  - Performance Schema Setup
- Schemas:** DB\_PROJECT1 is selected.
- Tables:** A list of tables including GAME, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, PLAYER, PLAYINGPOSITION, REGULAR, STADIUM, and TEAM.
- Query Editor (Query 5):**

```

1 USE DB_PROJECT1;
2
3
4 E Create table REGULAR (
5   Gid INT NOT NULL,
6   PRIMARY KEY(Gid),
7   FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
8 );
9
10
11
12
13
14
15
16
  
```
- Action Output:**

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec

### 11.CREATE TABLE ‘EXTRA’

This table has a attribute ‘Gid’ set as a foreign key referencing the ‘KNOCK\_OUT’ table. This entity type is a subclass of a ‘KNOCK\_OUT’ Entity type.

```
Create table EXTRA (
  Gid INT NOT NULL,
  PRIMARY KEY(Gid),
  FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- Left Sidebar (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Sidebar (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Sidebar (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Schemas:** DB\_PROJECT1 is selected, showing its tables: EXTRA, GAME, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, PLAYER, PLAYINGPOSITION, REGULAR, STADIUM.
- Query Editor:** A query named "Query 5" is running under the schema "#DBPROJECT". The query is:

```

1 USE DB_PROJECT1;
2
3 Create table EXTRA (
4     Gid INT NOT NULL,
5     PRIMARY KEY(Gid),
6     FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
7 );
8
9
10
11
12
13
14
15
16
    
```
- Action Output:** Shows the execution log with 5 rows of data, all successful (green checkmarks), with execution times and durations:

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec
5	18:42:58	C 0 row(s) affected	0.027 sec

## 12.CREATE TABLE ‘PENALTY’

This table has an attribute ‘Gid’ set as a foreign key referencing the ‘KNOCK\_OUT’ table. This entity type is a subclass of a ‘KNOCK\_OUT’ Entity type.

```
Create table PENALTY (
Gid INT NOT NULL,
PRIMARY KEY(Gid),
FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
);
```

```

USE DB_PROJECT1;
Create table PENALTY (
Gid INT NOT NULL,
PRIMARY KEY(Gid),
FOREIGN KEY(Gid) REFERENCES KNOCK_OUT(Gid)
);

```

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec
5	18:42:58	C 0 row(s) affected	0.027 sec
6	18:43:31	C 0 row(s) affected	0.046 sec

### 13.CREATE TABLE ‘MATCH\_RESULTS’

This table has information on winner team of the tournament, along with the points scored by Team1 and Team2. We used Attribute ‘Team1\_Points’ and ‘Team2\_Points’. As mentioned in the requirement ‘GROUP\_GAME’ has a draw match. If in any match, winning team gets 2 points and a losing team doesn’t get any point. But if it’s a draw match then both the teams get 1 point each. So, it’s easier to determine a winner with points specially in these cases.

```

Create table MATCH_RESULTS (
M_id INT NOT NULL,
Gid INT NOT NULL,
Team1_Points INT NOT NULL,
Team2_Points INT NOT NULL,
PRIMARY KEY(M_id),
FOREIGN KEY(Gid) REFERENCES GAME(Gid)
);

```

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Management):** Contains sections for Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore.
- Left Panel (INSTANCE):** Contains sections for Startup / Shutdown, Server Logs, and Options File.
- Left Panel (PERFORMANCE):** Contains sections for Dashboard, Performance Reports, and Performance Schema Setup.
- Left Panel (SCHEMAS):** Shows the current schema structure:
  - GROUP\_COUNTRY
  - GROUP\_GAME
  - GROUPS
  - KNOCK\_OUT
  - MATCH\_RESULTS
  - PENALTY
  - PLAYER
  - PLAYINGPOSITION
  - REGULAR
  - STADIUM
  - TEAM
  - Views
- Central Area (Query Editor):** A query window titled "Query 5" is open, showing the SQL code for creating the `MATCH_RESULTS` table. The code includes `USE DB_PROJECT1;`, `Create table MATCH_RESULTS ( M_id INT NOT NULL, Gid INT NOT NULL, Team1_Points INT NOT NULL, Team2_Points INT NOT NULL, PRIMARY KEY(M_id), FOREIGN KEY(Gid) REFERENCES GAME(Gid) );`.
- Bottom Area (Action Output):** A table showing the execution results:
 

	Time	A. Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec
5	18:42:58	C 0 row(s) affected	0.027 sec
6	18:43:31	C 0 row(s) affected	0.046 sec
7	18:44:16	U 0 row(s) affected	0.00024 sec
8	18:44:16	C 0 row(s) affected	0.035 sec

#### 14.CREATE TABLE ‘CARDS\_R’

This table stores three types of cards.

```
CREATE TABLE CARDS_R (
Card_id INT NOT NULL,
Ctype VARCHAR (15) NOT NULL,
PRIMARY KEY (Card_id)
);
```

The screenshot shows the MySQL Workbench interface with the following details:

- Left Sidebar (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Sidebar (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Sidebar (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Schemas:** DB\_PROJECT1 (selected), Tables: CARDS\_R, EXTRA, GAME, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, MATCH\_RESULTS, PENALTY, PLAYER.
- Query Editor (Query 5):**

```

1 USE DB_PROJECT1;
2
3 CREATE TABLE CARDS_R (
4     Card_id INT NOT NULL,
5     Ctype VARCHAR(15) NOT NULL,
6     PRIMARY KEY (Card_id)
7 );
8
9
10
11
12
13
14
15
16
    
```
- Action Output:**

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec
5	18:42:58	C 0 row(s) affected	0.027 sec
6	18:43:31	C 0 row(s) affected	0.046 sec
7	18:44:16	U 0 row(s) affected	0.0024 sec
8	18:44:16	C 0 row(s) affected	0.035 sec
9	18:44:54	C 0 row(s) affected	0.026 sec

## 15. CREATE TABLE 'GOALS'

This table stores three types of goals

```

CREATE TABLE GOALS (
Goal_id INT NOT NULL,
Goal_Type VARCHAR(15) NOT NULL,
PRIMARY KEY(Goal_id)
);
    
```

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Management):** Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore.
- Left Panel (Instance):** Startup / Shutdown, Server Logs, Options File.
- Left Panel (Performance):** Dashboard, Performance Reports, Performance Schema Setup.
- Schemas:** CARDs\_R, EXTRA, GAME, GOALS, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, MATCH\_RESULTS, PENALTY, PLAYER, PLAYINGPOSITION.
- Query Editor (Query 5):**

```

1 USE DB_PROJECT1;
2
3 CREATE TABLE GOALS (
4     Goal_id INT NOT NULL,
5     Goal_Type VARCHAR(15) NOT NULL,
6     PRIMARY KEY(Goal_id)
7
8
9
10
11
12
13
14
15
16
    
```
- Action Output:**

	Time	A Response	Duration / Fetch Time
1	18:39:40	C 0 row(s) affected	0.038 sec
2	18:40:58	C 0 row(s) affected	0.038 sec
3	18:41:31	C 0 row(s) affected	0.031 sec
4	18:42:23	C 0 row(s) affected	0.029 sec
5	18:42:58	C 0 row(s) affected	0.027 sec
6	18:43:31	C 0 row(s) affected	0.046 sec
7	18:44:16	U 0 row(s) affected	0.0024 sec
8	18:44:16	C 0 row(s) affected	0.035 sec
9	18:44:54	C 0 row(s) affected	0.026 sec
10	18:45:32	C 0 row(s) affected	0.026 sec

## 16.CREATE TABLE 'PLAYER\_MATCH'

```

Create table PLAYER_MATCH (
Player_match_id INT NOT NULL,
PNo INT NOT NULL,
Gid INT NOT NULL,
Total_Goals_scored INT NOT NULL,
Total_cards_recieved INT NOT NULL,
PRIMARY KEY (Player_match_id),
FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
FOREIGN KEY(Gid) REFERENCES GAME(Gid)
);
    
```

Feb 19, 2019

The screenshot shows the MySQL Workbench interface with the following details:

- Left Panel (Management):** Includes Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore.
- Left Panel (Instance):** Includes Startup / Shutdown, Server Logs, and Options File.
- Left Panel (Performance):** Includes Dashboard, Performance Reports, and Performance Schema Setup.
- Schemas Panel:** Shows the DB\_PROJECT1 schema expanded, revealing tables such as CARD\_MATCH\_PLAYER, CARDS\_R, EXTRA, GAME, GOAL\_MATCH\_PLAYER, GOALS, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, and KNOCK\_OUT.
- Query Editor (Query 5):** Contains the SQL code for creating the PLAYER\_MATCH table:
 

```

1 USE DB_PROJECT1;
2
3 Create table PLAYER_MATCH (
4     Player_match_id INT NOT NULL,
5     PNo INT NOT NULL,
6     Gid INT NOT NULL,
7     Total_Goals_scored INT NOT NULL,
8     Total_cards_received INT NOT NULL,
9     PRIMARY KEY (Player_match_id),
10    FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
11    FOREIGN KEY(Gid) REFERENCES GAME(Gid)
12 );
      
```
- Action Output:** Displays the execution results with 5 rows:
 

	Time	A: Response	Duration / Fetch Time
1	19:00:26	C 0 row(s) affected	0.037 sec
2	19:01:33	C Error Code: 1215. Cannot add foreign key constraint	0.040 sec
3	19:01:46	C Error Code: 1215. Cannot add foreign key constraint	0.033 sec
4	19:02:17	C 0 row(s) affected	0.039 sec
5	19:03:26	C 0 row(s) affected	0.027 sec

### 17.CREATE TABLE 'CARD\_MATCH\_PLAYER'

```

Create table CARD_MATCH_PLAYER (
id INT NOT NULL,
Card_ID INT NOT NULL,
Gid INT NOT NULL,
PNo INT NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
FOREIGN KEY(Gid) REFERENCES GAME(Gid)
      
```

```

FOREIGN KEY(Card_ID) REFERENCES CARDS_R(Card_id)

);
      
```

The screenshot shows the MySQL Workbench interface. On the left, the 'MANAGEMENT' panel is open, displaying various server status and configuration options. The 'SCHEMAS' panel shows the 'DB\_PROJECT1' schema with its tables: CARD\_MATCH\_PLAYER, CARDS\_R, EXTRA, GAME, GOALS, GROUP\_COUNTRY, GROUP\_GAME, GROUPS, KNOCK\_OUT, and MATCH\_RESULTS. The main central area is a query editor titled 'Query 5' containing the SQL code for creating the 'CARD\_MATCH\_PLAYER' table. The code includes primary key and foreign key constraints referencing other tables like 'PLAYER', 'GAME', and 'GOALS'. Below the query editor, the 'Action Output' pane shows the execution results: a single row affected in 0.037 seconds. The top right of the interface shows standard window controls and a toolbar.

```

USE DB_PROJECT1;
CREATE table CARD_MATCH_PLAYER (
    id INT NOT NULL,
    Card_ID INT NOT NULL,
    Gid INT NOT NULL,
    PNo INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
    FOREIGN KEY(Gid) REFERENCES GAME(Gid),
    FOREIGN KEY(Card_ID) REFERENCES CARDS_R(Card_id)
);

```

### 18.CREATE TABLE ‘GOAL\_MATCH\_PLAYER’

```

Create table GOAL_MATCH_PLAYER (
    id INT NOT NULL,
    goal_ID INT NOT NULL,
    Gid INT NOT NULL,
    PNo INT NOT NULL,
    Time_of_goal INT,
    PRIMARY KEY (id),
    FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
    FOREIGN KEY(Gid) REFERENCES GAME(Gid),
    FOREIGN KEY (goal_ID) REFERENCES GOALS (Goal_id)
);

```

MySQL Workbench

Local instance 3306 ×

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

SCHEMAS

DB\_PROJECT1

- Tables
- CARD\_MATCH\_PLAYER
- CARDS\_R
- EXTRA
- GAME
- GOALS
- GROUP\_COUNTRY
- GROUP\_GAME
- GROUPS
- KNOCK\_OUT
- MATCH\_RESULTS

Query 5 × DBPROJECT × Administration - Performance Schema Setup ×

```
1 USE DB_PROJECT1;
2
3 Create table GOAL_MATCH_PLAYER (
4     id INT NOT NULL,
5     goal_ID INT NOT NULL,
6     Gid INT NOT NULL,
7     PNo INT NOT NULL,
8     Time_of_goal INT,
9     PRIMARY KEY (id),
10    FOREIGN KEY(PNo) REFERENCES PLAYER(PNo),
11    FOREIGN KEY(Gid) REFERENCES GAME(Gid),
12    FOREIGN KEY(goal_ID) REFERENCES GOALS (Goal_id)
13 );
14
15
16
17
18
```

100% 1:15

Action Output

	Time	A Response	Duration / Fetch Time
1	19:00:26	C 0 row(s) affected	0.037 sec
2	19:01:33	X Error Code: 1215. Cannot add foreign key constraint	0.040 sec
3	19:01:46	X Error Code: 1215. Cannot add foreign key constraint	0.033 sec
4	19:02:17	C 0 row(s) affected	0.039 sec

Query Completed