March 2023

FIFA World Cup Database Project

Group Stage Relational database, SQL and Python queries

Created by

Shanice Williams

Creating the ER Model

To design our database in the most optimum way, we need to design an enitity relationship model. This will graphically show all of the tables that we will need to create within out database, all of the atrributes or column names for each table and how each table relates to each other.

To build this model we need to follow 5 steps:

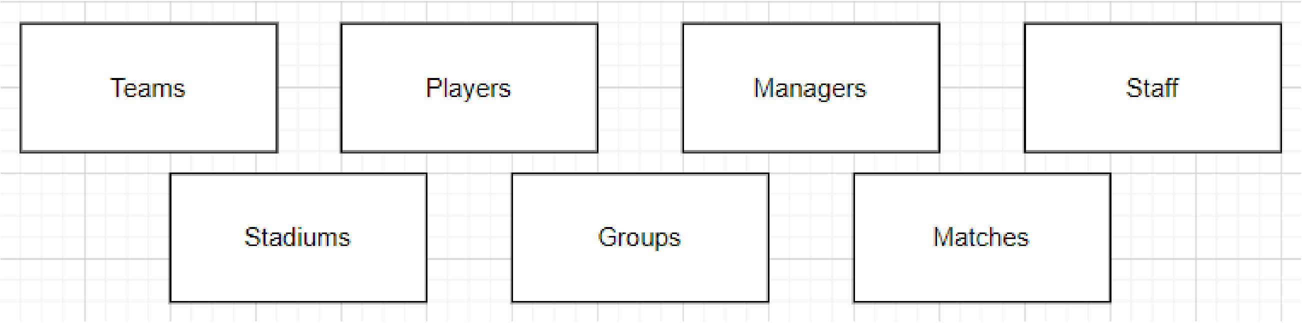
1. Entity identification
2. Relationship identification
3. Cardinality identification
4. Attribute identification
5. Create the final ER diagram

Step 1: Entity identification

To identify the entities of this database we must refer to the use case:

> The FIFA World Cup 2022 t is scheduled to take place in Qatar from 20 November to 18 December 2022! You will need to design and develop a database to store data in a relational format for national football teams, football players, head coach managers, staff members, stadiums, groups and football games.

Here we can see there are 7 possible entities listed:



Step 2: Relationship Identificaiton

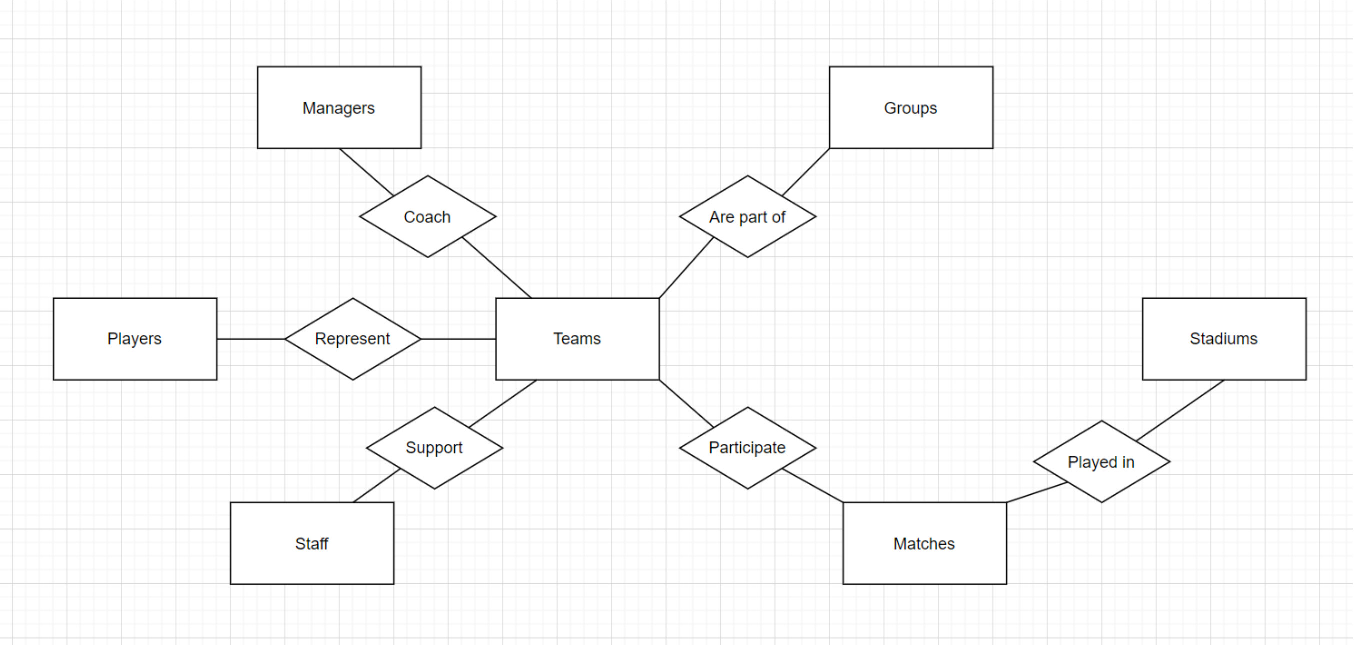
As the team is integral to a tournament, I decided to start with the team entity as the lynchpin for establishing relations between each entity.

From the use case we can see that each team has several players so there must be a relation here. Each team also has multiple staff and a manager so there must be relations here also between the manager and the team and the staff and the team.

During the tournament each football stadium can host multiple games so we will require a relation between stadiums and matches.

Each group will have multiple teams so there must be a relation between teams and groups, and as each team will be involved in multiple matches, there must be a relation between teams and matches also.

This all culminates in the below model:



One crucial question that remain following the review of the logical arguments that this model is based on, is whether there was an error when it comes to the following statement:

> During the tournament, each football stadium can host multiple games. However, each game is already planned for a specific football stadium.

Due to how this was worded I was not sure whether this was ain implication that games were at certain stadiums or the more complicated theory that there was also a relation between groups and stadiums (that groups played an specific stadiums). If this second assumption was the case this would add another relation to the model.

I decided to go with the first assumption and continue with the create model.

Step 3: Cardinality Identification

When identifying the cardinality of the relationships we were asked to consider the following requirements:

- Each national team might have several players.
- Each national team has only one head coach manager, but it can have multiple staff members.
- Players, head coach manager and staff are associated only with their national team.
- During the tournament, each football stadium can host multiple games. However, each game is already planned for a specific football stadium.
- Each group can have multiple teams, and each team participates only in one group.
- Each team can play many football games.

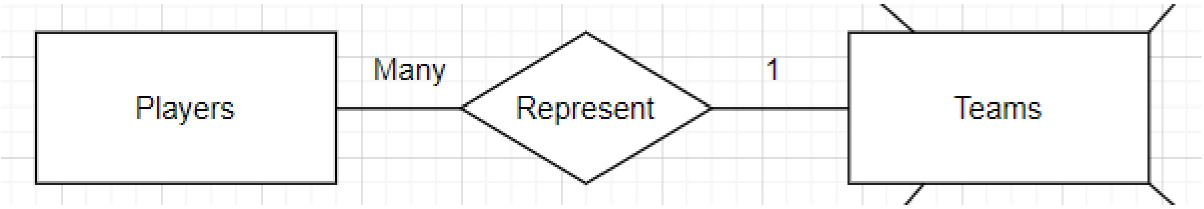
I will take each relation step by step and consider each requirement as I go.

Player and Team Relationship

For this relation we must consider the following requirements:

- Each national team might have several players.
- Players, head coach manager and staff are associated only with their national team.

Based on these two statements we can ascertain that the cardinality between these two entities is many to one. This is because we have many players representing exactly one team.

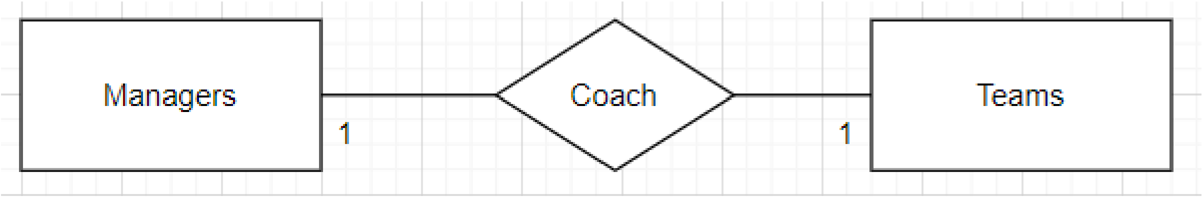


Manager and Team Relationship

We are told to consider the following requirements:

- Each national team has only one head coach manager, but it can have multiple staff members.
- Players, head coach manager and staff are associated only with their national team.

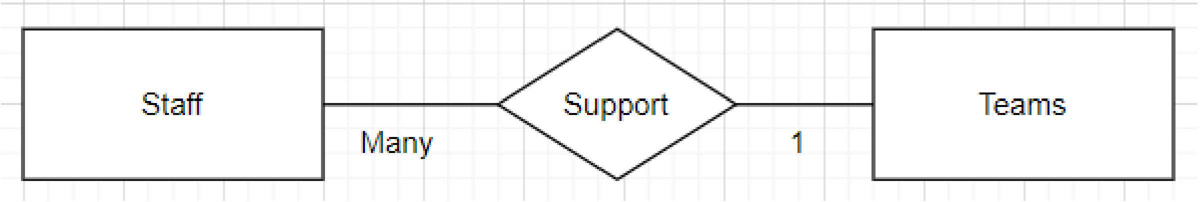
As every team manager is only related to one team and the relation is one to one is one to one I did contemplate combining these entities to save computational effort, however based on some of the queries we will need to run on the database, I have decided against this.



Staff and Team Relationship

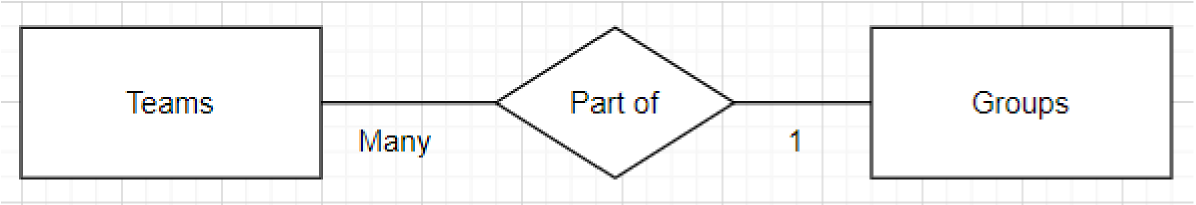
From the requirements provided we can see that there are multiple staff members that support the team, but that these staff members will only support one team.

That being said we can determine that there is a many to one relation between the staff and the team.



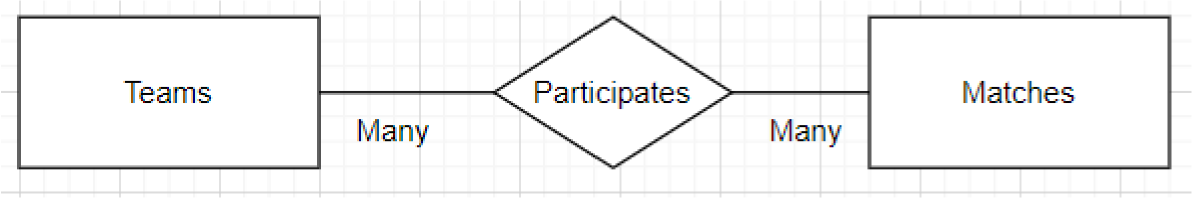
Team and Group Relationship

From our specification we are told that each group has multiple teams (4 to be exact), but that each team will only participate in one group. We can therefore establish a many to one relationship between the team and the group entity.



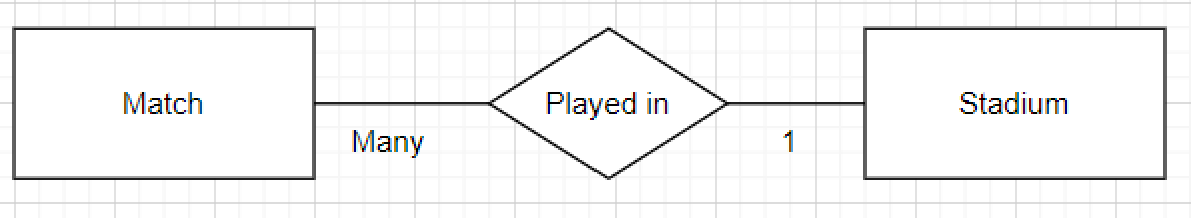
Team and Match relationship

According to the specification, each team will participate in multiple matches and likewise there will be many matches in which more than one team will participate. Therefore we can establish the cardinality of this relationship to be many to many.



Match and Stadium Relationship

There were 8 stadiums used for the world cup each hosting a number of matches. Every match will take place in a single stadium, and therefore we can assume this is a many to one relation.



Step 4: Attribute Identification

When identifying the attributes to use for this database it was important to review the queries that were to be completed once the database is created.

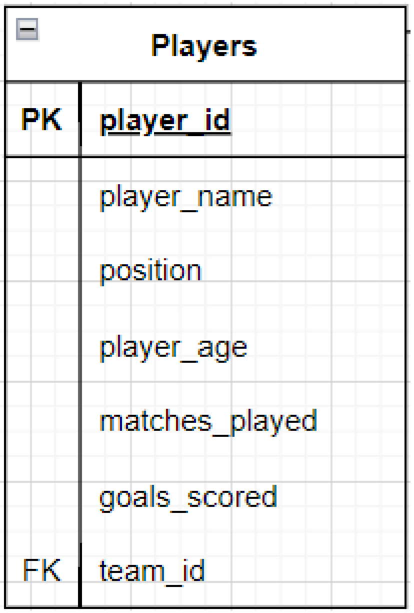
4.1 Export data about football players.
4.2 Export data about head coach managers and their associated teams.
4.3 Export data to show the group of each team.
4.4 Export data to show the football stadium for each team game.
4.5 Export data to count how many staff members are in the database per team.
4.6 Export data to find the teams that start with "Eng".
4.7 Export data to find the staff members per head coach.
4.8 Export data to find the youngest manager.
4.9 Export data to find players between 22-30 years old (including 22 and 30).
4.10 Export data to find the average of age per football team.

It was also important to consider the cardinality of the relations between the tables when creating the attributes as, when establishing 1 to many relations it is important to have foreign keys within the entity table to create that relation, however with many to many relations, as a composite table will be required, therefore identifying which keys would be required to create my composite keys were the main consideration.

For every enitity I created a respective id field to act as the primary key.

Player Attributes

I have chosen the following attributes for the player entity:



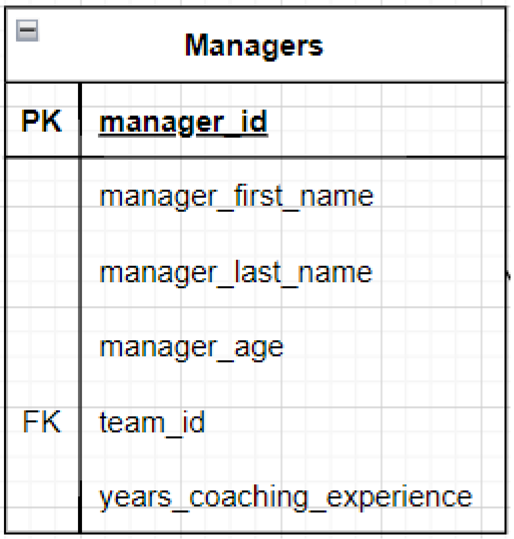
	Players
PK	**player_id**
	player_name
	position
	player_age
	matches_played
	goals_scored
FK	team_id

These attributes will provide general statistics for the players from the tournament, with the team_id included to for the relation with the team entity.

Team Attributes

For the team entity, we have several attributes to show information about each team in the competition, that could be interesting to know depending on what insights you wish to gain from your data. Thegroup id is also present to relate the groups and teams. I have added the group id to the team table because the each team is assigned to one group, but to assign the four teams to one group within the group table would not be prefered.

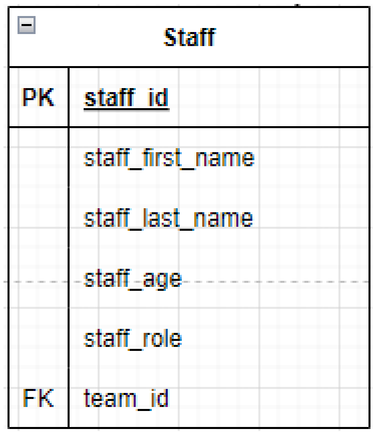
Manager Attributes



	Managers
PK	**manager_id**
	manager_first_name
	manager_last_name
	manager_age
FK	team_id
	years_coaching_experience

When considering the manager attributes, we had to consider some of the requirements of the queries we will have to run once the database is in place.
Namely:
4.8 Export data to find the youngest manager.
This meant that it was essential to have an attribute to either ascertain age (date of birth), or with an actual age field. In this case I have opted for an age attribute. I have also used the `team_id` as a foreign key to link to the team they are responsible for.

Staff Attributes

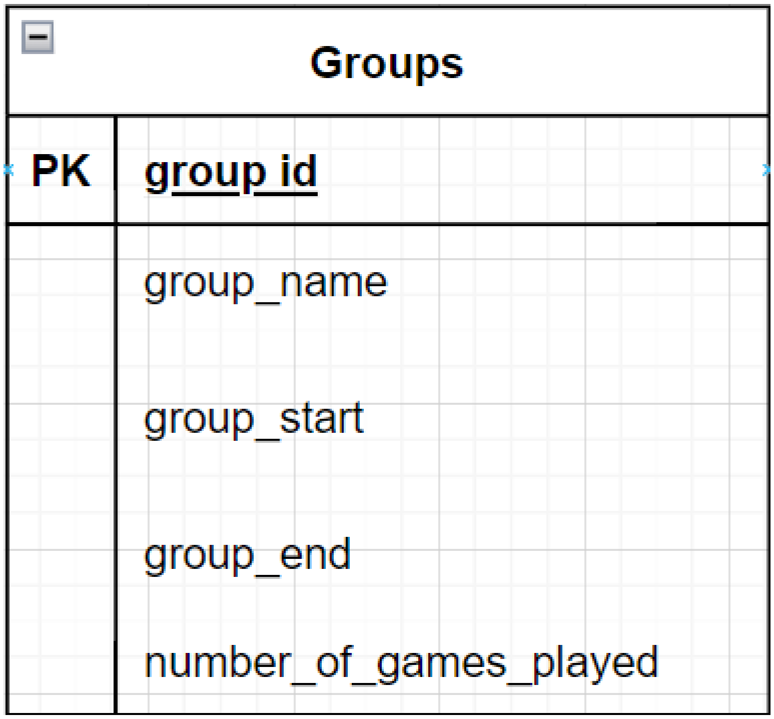


	Staff
PK	staff_id
	staff_first_name
	staff_last_name
	staff_age
	staff_role
FK	team_id

The staff attributes are just general information about the staff, including the team id to associate with the team table. This will be important when querying for 4.7 Export data to find the staff members per head coach.

Group Attributes

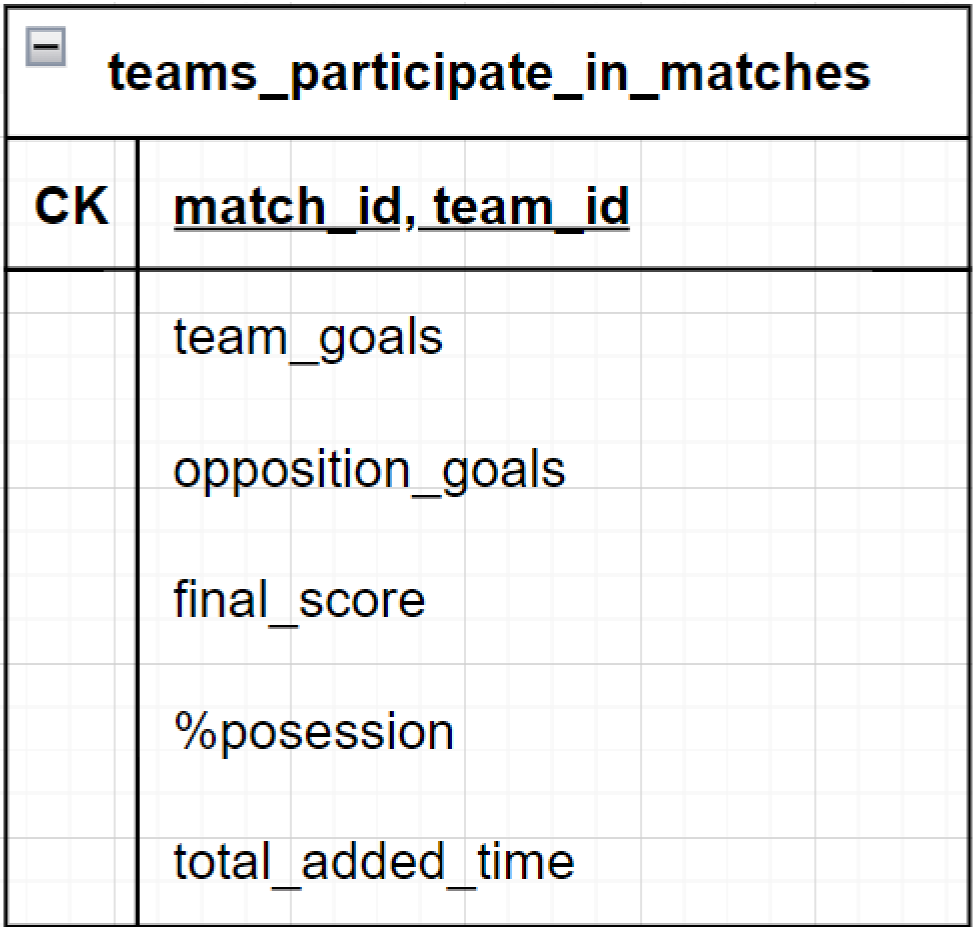
The main condsideration when creating attributes for this table was ensuring there was a group id present. Although there is a need to be able to ascertain the group of each team in the specification, this only impacts the attributes of the team as that is where that information will be obtained from. I included group start and end dates, as it could be some thing to consider when reflecting on how close together group matches were and time btween the group stages and later rounds.



Groups	
PK	group id
	group_name group_start group_end number_of_games_played

Match Attributes

I have included the stadium id as a foreign key in this table to relate each match to the stadium it was played in. As this table has a many to many relationship with the team entity, I will need a composite table to reflect this relation. Although it will not be reflected in the final ER diagram I did consider it, and it's fields at this stage:



teams_participate_in_matches

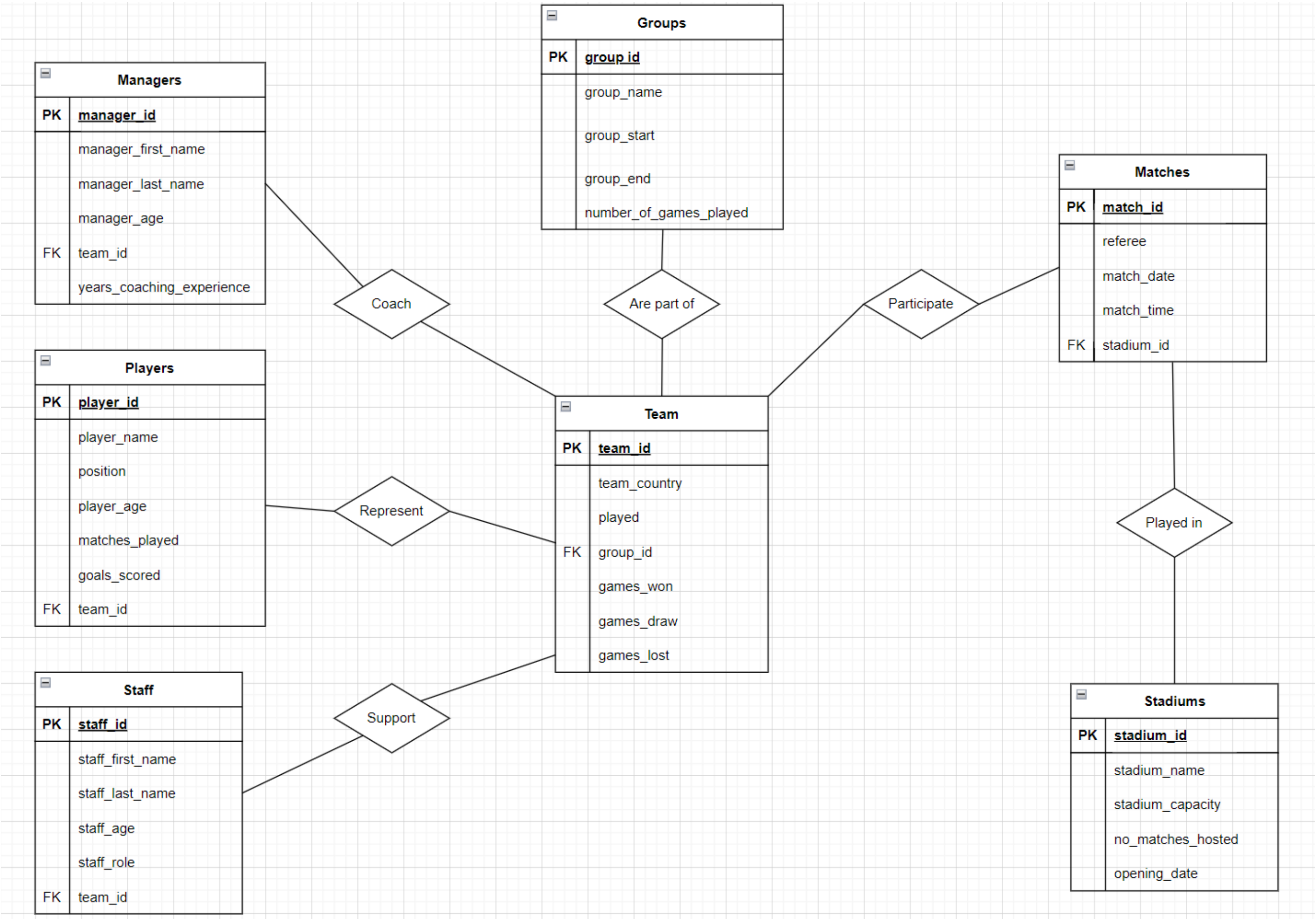
CK	match_id, team_id
	team_goals
	opposition_goals
	final_score
	%posession
	total_added_time

Stadium Attributes

As with creating the group attributes, there was only one requirement which involved the stadium data, and so there is a primary key of stadium_id which is used in the match entity table to establish that relation. Otherwise the other attributes are general information about a stadium that could be interesting to know.

Step 5: Creating the final ER diagram

To complete my final ER digram I converted the lines to reflect the cardinality and this was the final result:


Groups
PK group_id
group_name
group_start
group_end
number_of_games_played
Managers
PK manager_id
manager_first_name
manager_last_name
manager_age
FK team_id
years_coaching_experience
Matches
PK match_id
referee
match_date
match_time
FK stadium_id
Coach
Are part of
Participate
Players
PK player_id
player_name
position
player_age
matches_played
goals_scored
FK team_id
Team
PK team_id
team_country
played
FK group_id
games_won
games_draw
games_lost
Represent
Played in
Staff
PK staff_id
staff_first_name
staff_last_name
staff_age
staff_role
FK team_id
Support
Stadiums
PK stadium_id
stadium_name
stadium_capacity
no_matches_hosted
opening_date

Creating the database

Creation of the database within mySQL is performed using the `CREATE DATABASE` statement as shown below. I then used the `USE` statement to ensure I would be building my data tables within the correct database.

```
CREATE DATABASE worldcup2022;

USE worldcup2022;
```

Creating the tables

When creating the tables there were several considerations. Firstly I had to consider the order in which I created each table, as some would require the primary key of other tables to be foreign keys to create the required relations. For instance, the first table I created was the groups table, because the `group_id` primary key would then be used in the `teams` table as a foreign key.

Choosing data types and requirements

Something else to consider when building my tables was the data types used and what other requirements may be required. I chose for all of the table ids to be `VARCHAR(8)`, allowing for flexibility for adding to this data if it were used for other tournaments. The largest number of rows of data would be the players table, as there were 32 teams in the tournament and there could be up to 26 players in each squad.

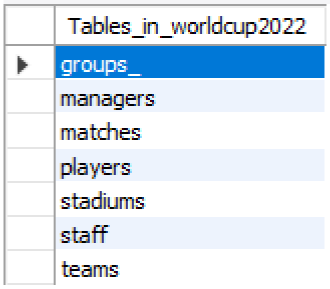
Although each group name is small, I kept the maximum character length to 16 for flexiblity for a new group naming convention, as it is looking more likely there will be a greater number of teams, and so more groups in future competitions, so there can be flexibility of the name of the groups for the future.

The `DATE` datatype and `TIME` dataype were used for all appropriate attributes.

Another consideration I made when creating these tables was that my database will only represent the group stage of the tournament and house data from all of the group stage matches once completed.

```
/*Create groups table*/
CREATE TABLE groups_ (
group_id VARCHAR(8),
group_name VARCHAR(16) NOT NULL,
```

```sql
group_start DATE NOT NULL, /*Representing the date the group started (YYYY-MM-DD)*/
group_end DATE NOT NULL, /*Representing the date the group ended (YYYY-MM-DD)*/
number_of_games_played INT(3),
PRIMARY KEY (group_id)
);

/*Create teams table*/
CREATE TABLE teams (
team_id VARCHAR(8),
team_country VARCHAR(32) NOT NULL,
played INT(3) NOT NULL,
matches_won INT(3) NOT NULL, /*Number of matches won in the group stage*/
matches_drawn INT(3) NOT NULL, /*Number of matches drawn in the group stage*/
matches_lost INT(3) NOT NULL, /*Number of matches lost in the group stage*/
group_id VARCHAR(8),
PRIMARY KEY (team_id),
FOREIGN KEY (group_id) REFERENCES groups(group_id)
);

/*Create players table*/
CREATE TABLE players (
player_id VARCHAR(8),
player_name VARCHAR(32) NOT NULL,
position VARCHAR(2) NOT NULL, /*this will be a position code such as GK or AM*/
player_age INT(2) NOT NULL,
matches_played INT(3) NOT NULL, /*matches within the group stage, starting or from the bench*/
goals_scored INT(3) NOT NULL,
team_id VARCHAR(8),
PRIMARY KEY (player_id),
FOREIGN KEY (team_id) REFERENCES teams(team_id)
);

/*Create managers table*/
CREATE TABLE managers (
manager_id VARCHAR(8),
manager_name VARCHAR(30) NOT NULL,
manager_age INT(3) NOT NULL,
years_coaching_experience INT(3),
team_id VARCHAR(8),
PRIMARY KEY (manager_id),
FOREIGN KEY (team_id) REFERENCES teams(team_id)
);

/*Create staff table*/
CREATE TABLE staff (
staff_id VARCHAR(8),
staff_name VARCHAR(30) NOT NULL,
staff_age INT(8) NOT NULL,
staff_role VARCHAR(30) NOT NULL,
team_id VARCHAR(8),
PRIMARY KEY (staff_id),
FOREIGN KEY (team_id) REFERENCES teams(team_id)
);

/*Create stadiums table*/
CREATE TABLE stadiums (
stadium_id VARCHAR(8),
stadium_name VARCHAR(30) NOT NULL,
stadium_capacity INT(8) NOT NULL,
no_matches_hosted INT(3) NOT NULL,
opening_date DATE NOT NULL,
PRIMARY KEY (stadium_id)
);

/*Create matches table*/
CREATE TABLE matches (
match_id VARCHAR(8),
referee VARCHAR(30) NOT NULL,
match_date DATE NOT NULL,
match_time TIME NOT NULL, /*All times in local time for Qatar*/
attendance INT(8) NOT NULL, /*Number of people who attended the match*/
stadium_id VARCHAR(8) NOT NULL,
PRIMARY KEY (match_id),
FOREIGN KEY (stadium_id) REFERENCES stadiums(stadium_id)
);
```



Tables_in_worldcup2022
groups_
managers
matches
players
stadiums
staff
teams

Establishing the Composite table

To create the composite table required the management of constraints to ensure that the composite key was created correctly and maintained referential integrity.

```
/*Create teams_participate_in_matches table*/
CREATE TABLE teams_participate_in_matches (
match_id VARCHAR(8),
team_id VARCHAR(8),
team_goals INT(3),
opposition_goals INT(3) NOT NULL,
final_score VARCHAR(5)  NOT NULL,
percent_posession INT(3) NOT NULL,
total_added_time INT(3) NOT NULL,
PRIMARY KEY (match_id, team_id),
CONSTRAINT FK1 FOREIGN KEY (match_id) REFERENCES matches(match_id),
CONSTRAINT FK2 FOREIGN KEY (team_id) REFERENCES teams(team_id)
);
```

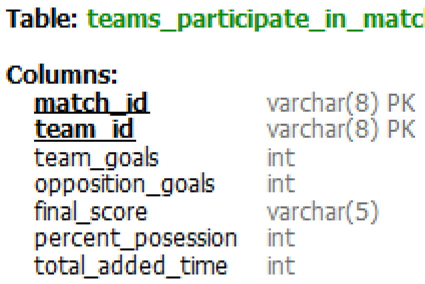


Table: teams_participate_in_matches

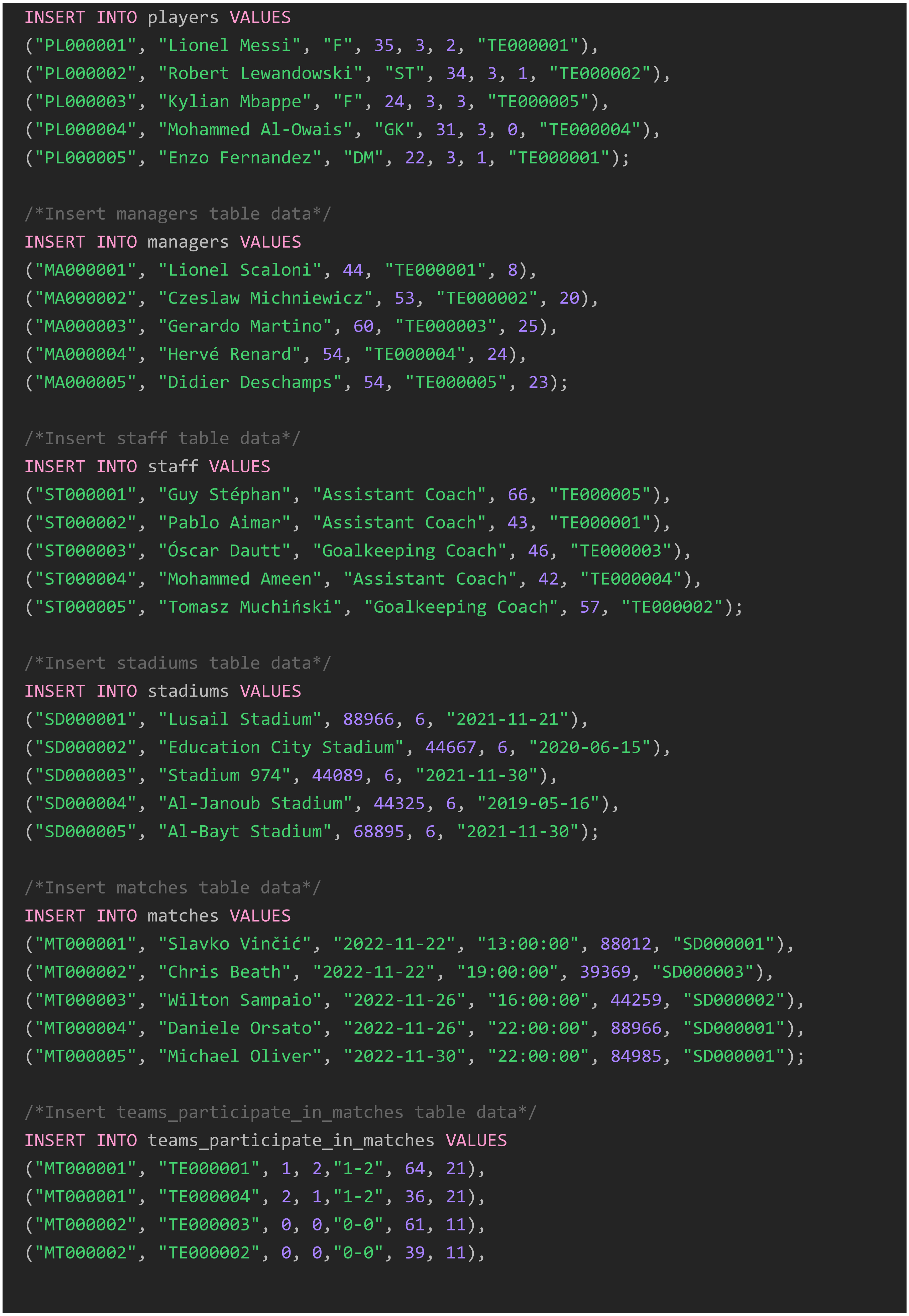
Columns:

match_id	varchar(8) PK
team_id	varchar(8) PK
team_goals	int
opposition_goals	int
final_score	varchar(5)
percent_posession	int
total_added_time	int

Inserting Data into the tables

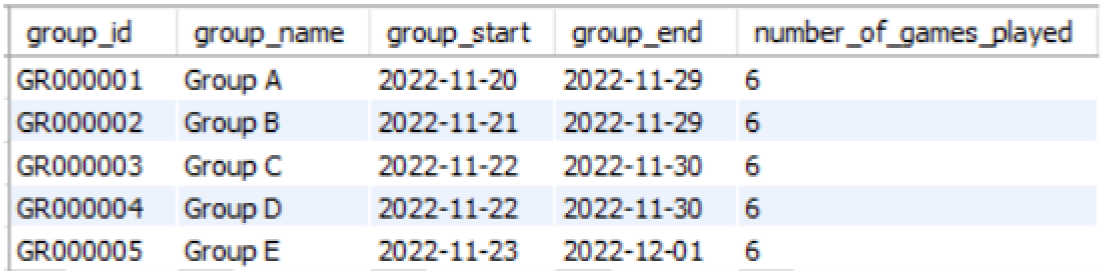
The final part of this stage was ensuring there was data in the database to be able to query and ensuring that each time nore data was added, that it complied with the reults applied when creating each table.

```
/*Insert groups table data*/
INSERT INTO groups VALUES
("GR000001", "Group A", "2022-11-20", "2022-11-29", 6),
("GR000002", "Group B", "2022-11-21", "2022-11-29", 6),
("GR000003", "Group C", "2022-11-22", "2022-11-30", 6),
("GR000004", "Group D", "2022-11-22", "2022-11-30", 6),
("GR000005", "Group E", "2022-11-23", "2022-12-01", 6);

/*Insert teams table data*/
INSERT INTO teams VALUES
("TE000001", "Argentina", 3, "GR000003", 2, 0, 1),
("TE000002", "Poland", 3, "GR000003", 1, 1, 1),
("TE000003", "Mexico", 3, "GR000003", 1, 1, 1),
("TE000004", "Saudi Arabia", 3, "GR000003", 1, 0, 2),
("TE000005", "France", 3, "GR000004", 2, 0, 1);

/*Insert players table data*/
```



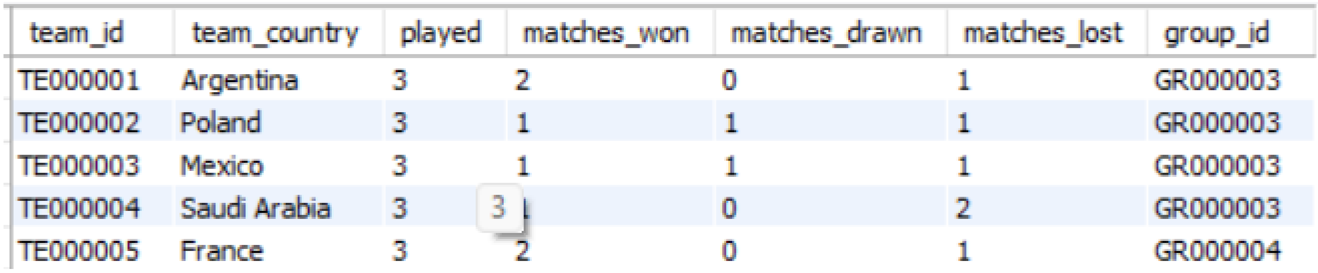
```
INSERT INTO players VALUES
("PL000001", "Lionel Messi", "F", 35, 3, 2, "TE000001"),
("PL000002", "Robert Lewandowski", "ST", 34, 3, 1, "TE000002"),
("PL000003", "Kylian Mbappe", "F", 24, 3, 3, "TE000005"),
("PL000004", "Mohammed Al-Owais", "GK", 31, 3, 0, "TE000004"),
("PL000005", "Enzo Fernandez", "DM", 22, 3, 1, "TE000001");

/*Insert managers table data*/
INSERT INTO managers VALUES
("MA000001", "Lionel Scaloni", 44, "TE000001", 8),
("MA000002", "Czeslaw Michniewicz", 53, "TE000002", 20),
("MA000003", "Gerardo Martino", 60, "TE000003", 25),
("MA000004", "Hervé Renard", 54, "TE000004", 24),
("MA000005", "Didier Deschamps", 54, "TE000005", 23);

/*Insert staff table data*/
INSERT INTO staff VALUES
("ST000001", "Guy Stéphan", "Assistant Coach", 66, "TE000005"),
("ST000002", "Pablo Aimar", "Assistant Coach", 43, "TE000001"),
("ST000003", "Óscar Dautt", "Goalkeeping Coach", 46, "TE000003"),
("ST000004", "Mohammed Ameen", "Assistant Coach", 42, "TE000004"),
("ST000005", "Tomasz Muchiński", "Goalkeeping Coach", 57, "TE000002");

/*Insert stadiums table data*/
INSERT INTO stadiums VALUES
("SD000001", "Lusail Stadium", 88966, 6, "2021-11-21"),
("SD000002", "Education City Stadium", 44667, 6, "2020-06-15"),
("SD000003", "Stadium 974", 44089, 6, "2021-11-30"),
("SD000004", "Al-Janoub Stadium", 44325, 6, "2019-05-16"),
("SD000005", "Al-Bayt Stadium", 68895, 6, "2021-11-30");

/*Insert matches table data*/
INSERT INTO matches VALUES
("MT000001", "Slavko Vinčić", "2022-11-22", "13:00:00", 88012, "SD000001"),
("MT000002", "Chris Beath", "2022-11-22", "19:00:00", 39369, "SD000003"),
("MT000003", "Wilton Sampaio", "2022-11-26", "16:00:00", 44259, "SD000002"),
("MT000004", "Daniele Orsato", "2022-11-26", "22:00:00", 88966, "SD000001"),
("MT000005", "Michael Oliver", "2022-11-30", "22:00:00", 84985, "SD000001");

/*Insert teams_participate_in_matches table data*/
INSERT INTO teams_participate_in_matches VALUES
("MT000001", "TE000001", 1, 2,"1-2", 64, 21),
("MT000001", "TE000004", 2, 1,"1-2", 36, 21),
("MT000002", "TE000003", 0, 0,"0-0", 61, 11),
("MT000002", "TE000002", 0, 0,"0-0", 39, 11),
("MT000003", "TE000002", 2, 0,"2-0", 36, 18),
("MT000003", "TE000004", 0, 2,"2-0", 64, 18);
```

Group table with data



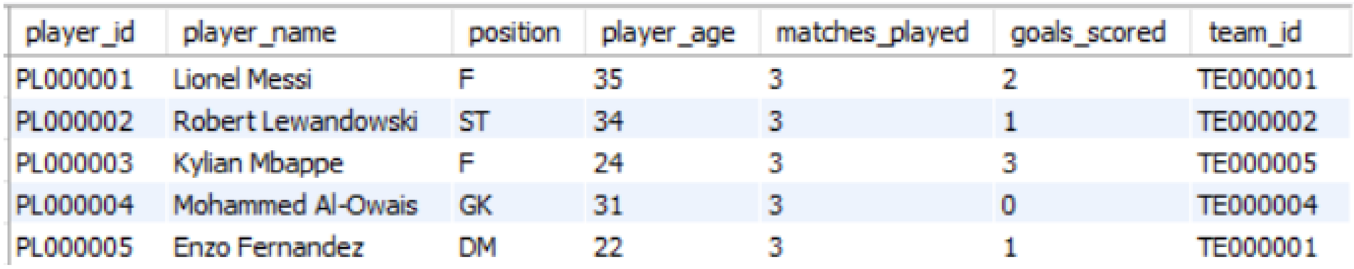
group_id	group_name	group_start	group_end	number_of_games_played
GR000001	Group A	2022-11-20	2022-11-29	6
GR000002	Group B	2022-11-21	2022-11-29	6
GR000003	Group C	2022-11-22	2022-11-30	6
GR000004	Group D	2022-11-22	2022-11-30	6
GR000005	Group E	2022-11-23	2022-12-01	6

Team table with data



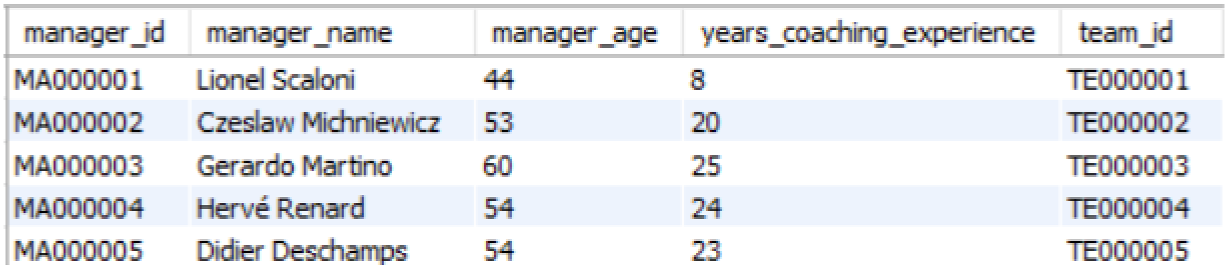
team_id	team_country	played	matches_won	matches_drawn	matches_lost	group_id
TE000001	Argentina	3	2	0	1	GR000003
TE000002	Poland	3	1	1	1	GR000003
TE000003	Mexico	3	1	1	1	GR000003
TE000004	Saudi Arabia	3	1	0	2	GR000003
TE000005	France	3	2	0	1	GR000004

Players table with data



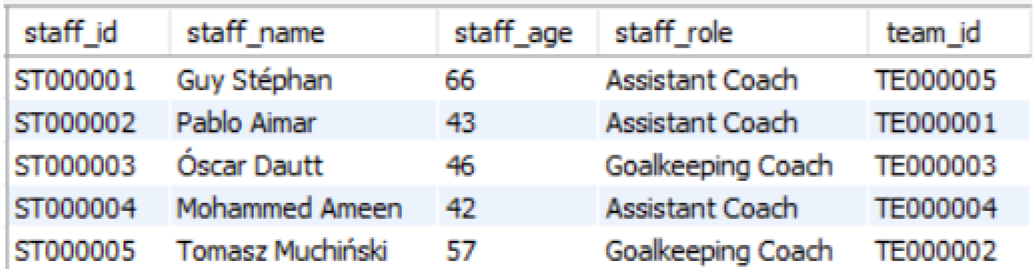
player_id	player_name	position	player_age	matches_played	goals_scored	team_id
PL000001	Lionel Messi	F	35	3	2	TE000001
PL000002	Robert Lewandowski	ST	34	3	1	TE000002
PL000003	Kylian Mbappe	F	24	3	3	TE000005
PL000004	Mohammed Al-Owais	GK	31	3	0	TE000004
PL000005	Enzo Fernandez	DM	22	3	1	TE000001

Managers table with data



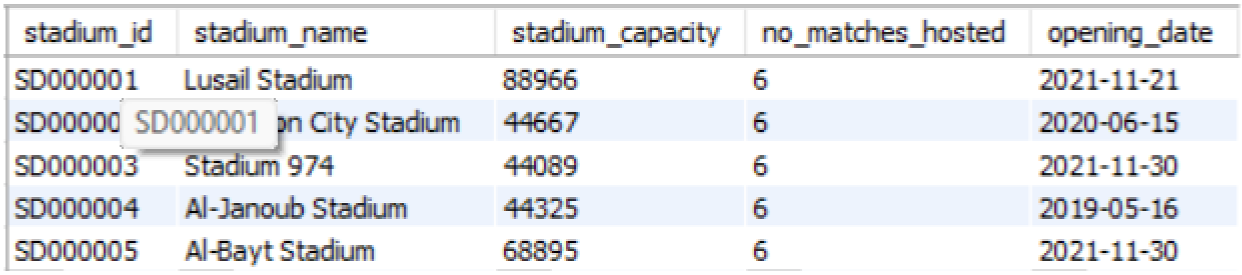
manager_id	manager_name	manager_age	years_coaching_experience	team_id
MA000001	Lionel Scaloni	44	8	TE000001
MA000002	Czeslaw Michniewicz	53	20	TE000002
MA000003	Gerardo Martino	60	25	TE000003
MA000004	Hervé Renard	54	24	TE000004
MA000005	Didier Deschamps	54	23	TE000005

Staff tables with data



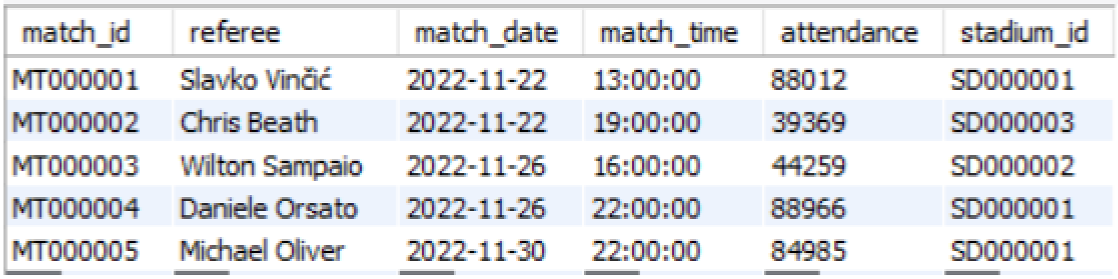
staff_id	staff_name	staff_age	staff_role	team_id
ST000001	Guy Stéphan	66	Assistant Coach	TE000005
ST000002	Pablo Aimar	43	Assistant Coach	TE000001
ST000003	Óscar Dautt	46	Goalkeeping Coach	TE000003
ST000004	Mohammed Ameen	42	Assistant Coach	TE000004
ST000005	Tomasz Muchiński	57	Goalkeeping Coach	TE000002

Stadiums table with data



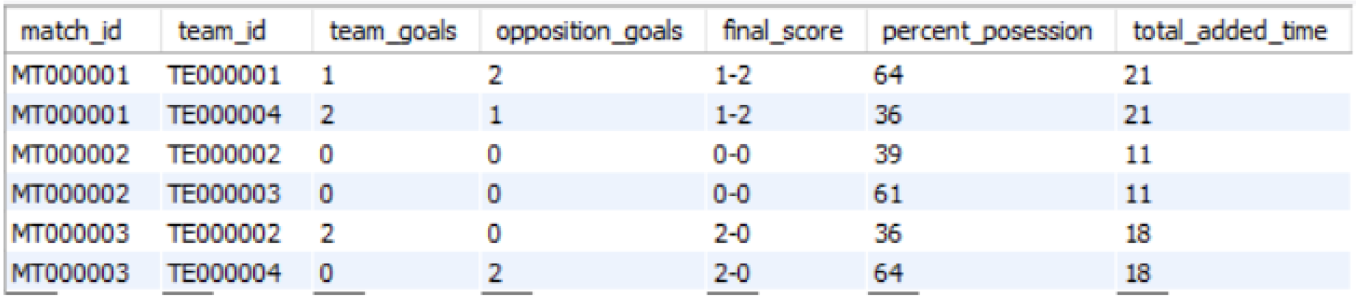
stadium_id	stadium_name	stadium_capacity	no_matches_hosted	opening_date
SD000001	Lusail Stadium	88966	6	2021-11-21
SD00000 SD000001	on City Stadium	44667	6	2020-06-15
SD000003	Stadium 974	44089	6	2021-11-30
SD000004	Al-Janoub Stadium	44325	6	2019-05-16
SD000005	Al-Bayt Stadium	68895	6	2021-11-30

Matches table with data



match_id	referee	match_date	match_time	attendance	stadium_id
MT000001	Slavko Vinčić	2022-11-22	13:00:00	88012	SD000001
MT000002	Chris Beath	2022-11-22	19:00:00	39369	SD000003
MT000003	Wilton Sampaio	2022-11-26	16:00:00	44259	SD000002
MT000004	Daniele Orsato	2022-11-26	22:00:00	88966	SD000001
MT000005	Michael Oliver	2022-11-30	22:00:00	84985	SD000001

Teams participate in matches table



match_id	team_id	team_goals	opposition_goals	final_score	percent_posession	total_added_time
MT000001	TE000001	1	2	1-2	64	21
MT000001	TE000004	2	1	1-2	36	21
MT000002	TE000002	0	0	0-0	39	11
MT000002	TE000003	0	0	0-0	61	11
MT000003	TE000002	2	0	2-0	36	18
MT000003	TE000004	0	2	2-0	64	18

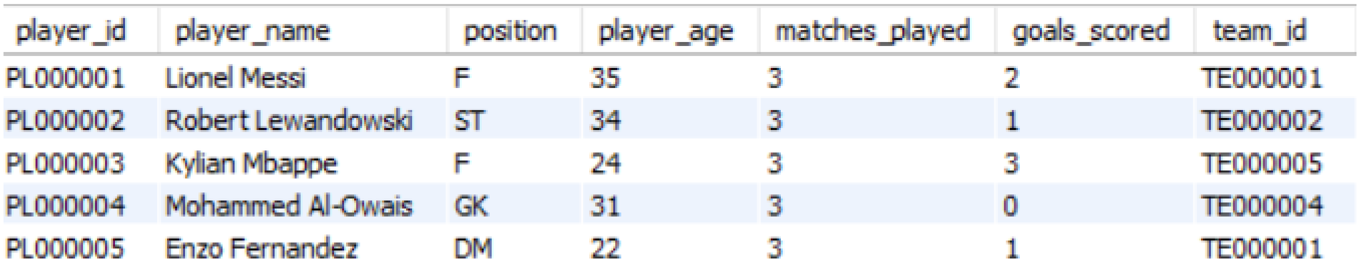
Querying the database using SQL

Once the database was created and had data in it, I coul start to build the queries to run so that we could get the data points being requested.

As can be seen below, I addressed each of the requirements in turn and created a select statement to be able to get the data requested. I have also provided the output of each query.

4.1 Export data about football players

```sql
/*Export data about football players.*/
SELECT * FROM players;
```



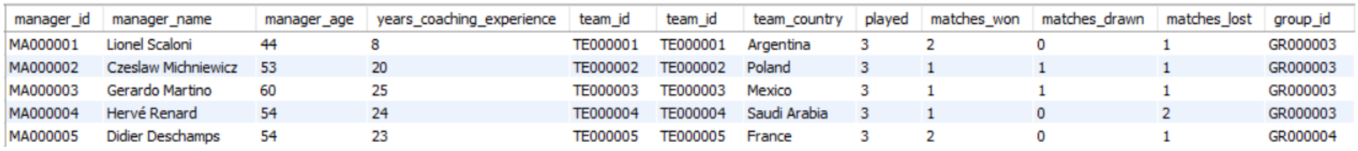
player_id	player_name	position	player_age	matches_played	goals_scored	team_id
PL000001	Lionel Messi	F	35	3	2	TE000001
PL000002	Robert Lewandowski	ST	34	3	1	TE000002
PL000003	Kylian Mbappe	F	24	3	3	TE000005
PL000004	Mohammed Al-Owais	GK	31	3	0	TE000004
PL000005	Enzo Fernandez	DM	22	3	1	TE000001

4.2 Export data about head coach managers and their associated teams

To export the data for each manager with only the team information for the team that they manage we need to utilise an inner join as can be seen below.

```sql
/* Data about head coach managers and their associated teams*/
SELECT * FROM managers, teams
WHERE managers.team_id=teams.team_id;
```

The result of which is this:

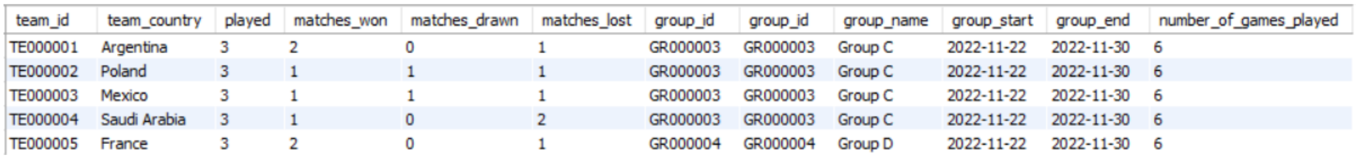


manager_id	manager_name	manager_age	years_coaching_experience	team_id	team_id	team_country	played	matches_won	matches_drawn	matches_lost	group_id
MA000001	Lionel Scaloni	44	8	TE000001	TE000001	Argentina	3	2	0	1	GR000003
MA000002	Czeslaw Michniewicz	53	20	TE000002	TE000002	Poland	3	1	1	1	GR000003
MA000003	Gerardo Martino	60	25	TE000003	TE000003	Mexico	3	1	1	1	GR000003
MA000004	Hervé Renard	54	24	TE000004	TE000004	Saudi Arabia	3	1	0	2	GR000003
MA000005	Didier Deschamps	54	23	TE000005	TE000005	France	3	2	0	1	GR000004

4.3 Export data to show the group of each team

```sql
/* Data to show the group of each team*/
SELECT * FROM teams, groups_
```

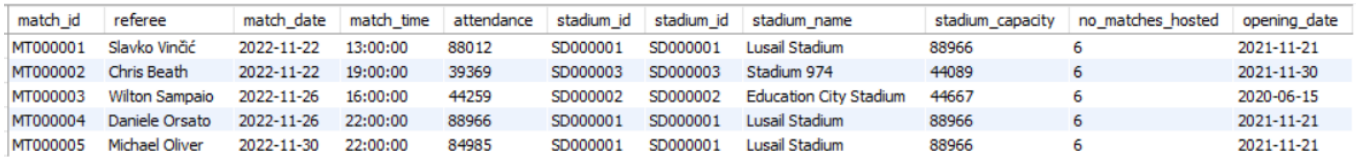
```
WHERE teams.group_id=groups_.group_id;
```



team_id	team_country	played	matches_won	matches_drawn	matches_lost	group_id	group_id	group_name	group_start	group_end	number_of_games_played
TE000001	Argentina	3	2	0	1	GR000003	GR000003	Group C	2022-11-22	2022-11-30	6
TE000002	Poland	3	1	1	1	GR000003	GR000003	Group C	2022-11-22	2022-11-30	6
TE000003	Mexico	3	1	1	1	GR000003	GR000003	Group C	2022-11-22	2022-11-30	6
TE000004	Saudi Arabia	3	1	0	2	GR000003	GR000003	Group C	2022-11-22	2022-11-30	6
TE000005	France	3	2	0	1	GR000004	GR000004	Group D	2022-11-22	2022-11-30	6

4.4 Export data to show the football stadium for each team game

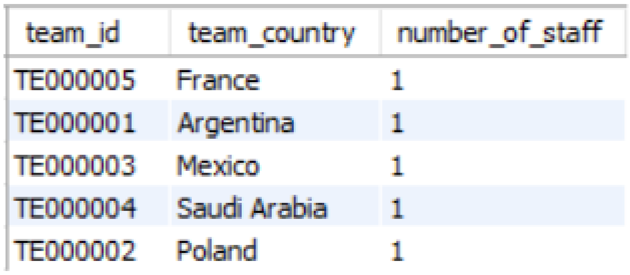
```
/* Data to show the football stadium for each team game*/
SELECT * FROM matches, stadiums;
```



match_id	referee	match_date	match_time	attendance	stadium_id	stadium_id	stadium_name	stadium_capacity	no_matches_hosted	opening_date
MT000001	Slavko Vinčić	2022-11-22	13:00:00	88012	SD000001	SD000001	Lusail Stadium	88966	6	2021-11-21
MT000002	Chris Beath	2022-11-22	19:00:00	39369	SD000003	SD000003	Stadium 974	44089	6	2021-11-30
MT000003	Wilton Sampaio	2022-11-26	16:00:00	44259	SD000002	SD000002	Education City Stadium	44667	6	2020-06-15
MT000004	Daniele Orsato	2022-11-26	22:00:00	88966	SD000001	SD000001	Lusail Stadium	88966	6	2021-11-21
MT000005	Michael Oliver	2022-11-30	22:00:00	84985	SD000001	SD000001	Lusail Stadium	88966	6	2021-11-21

4.5 Export data to count how many staff members are in the database per team

```
/* Data to count how many staff members are in the database per team*/
SELECT teams.team_id, teams.team_country, COUNT(staff.staff_id) AS number_of_staff
FROM staff, teams
WHERE staff.team_id = teams.team_id
GROUP BY staff.staff_id;
;
```



team_id	team_country	number_of_staff
TE000005	France	1
TE000001	Argentina	1
TE000003	Mexico	1
TE000004	Saudi Arabia	1
TE000002	Poland	1

4.6 Export data to find the teams that start with "Eng"

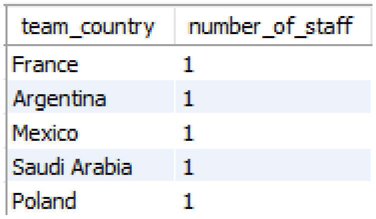
We are considering team_country as the team name.

```
/* Data to find the teams that start with "Eng"*/
SELECT * FROM teams WHERE team_country LIKE 'Eng%';
```

As there is no team starting with Eng in my database, these query yields no results.

4.7 Export data to find the staff members per head coach

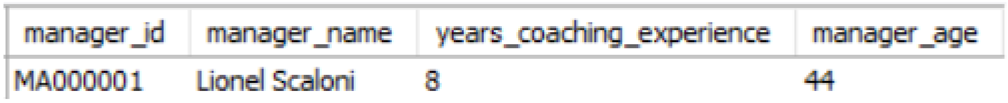
```sql
/* Data to find the staff members per head coach*/
SELECT M.manager_name, S.staff_name
FROM teams AS T, staff AS S, managers AS M
WHERE S.team_id = T.team_id AND T.team_id = M.team_id;
```



team_country	number_of_staff
France	1
Argentina	1
Mexico	1
Saudi Arabia	1
Poland	1

4.8 Export data to find the youngest manager

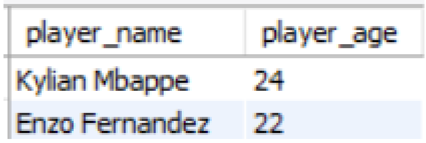
```sql
/* Data to find the youngest manager*/
SELECT M.manager_id, M.manager_name, M.years_coaching_experience, M.manager_age
FROM managers AS M
ORDER BY manager_age ASC LIMIT 1;
```



manager_id	manager_name	years_coaching_experience	manager_age
MA000001	Lionel Scaloni	8	44

4.9 Export data to find players between 22-30 years old (including 22 and 30)

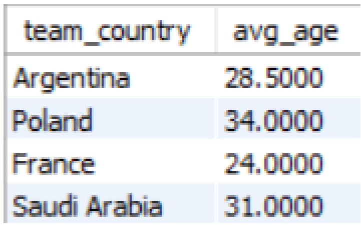
```sql
/* Data to find players between 22-30 years old (including 22 and 30)*/
SELECT p.player_name, p.player_age
FROM players AS p
WHERE p.player_age >= 22 AND p.player_age <= 30;
```



player_name	player_age
Kylian Mbappe	24
Enzo Fernandez	22

4.10 Export data to find the average of age per football team

```sql
/* Data to find the average of age per football team*/
SELECT t.team_country, AVG(p.player_age) AS avg_age
FROM players AS P,teams AS t
WHERE p.team_id = t.team_id
GROUP BY t.team_country;
```



team_country	avg_age
Argentina	28.5000
Poland	34.0000
France	24.0000
Saudi Arabia	31.0000

Querying the data using Python

When approaching querying the database using python, there were two main parts to this:

- Establishing the connection to the database
- using SQL in conjunction with python to gain the data requested by a user

Establishing database connection

To be able to establish a connection to the database I created locally, I installed and imported mySQL Connector for Python. This module allowed for communication between my pthon scripts and SQL Database. To make this simple and reproducible, I created this connection via a function as can be seen below.

```python
import mysql.connector

def create_db_connection(host_name, user_name, user_password, db_name):
    connection = None
    try:
        connection = mysql.connector.connect(
            host=host_name,
            user=user_name,
            passwd=user_password,
            database=db_name
        )
        print("MySQL Database connection successful")
    except Error as err:
        print(f"Error: '{err}'")
    return connection

host = "localhost"
user = "root"
password = "123456"
database = "worldcup2022"

connection = create_db_connection(host, user, password,database)
```

Querying the database

Once a connection was esatblished, I could use this connection along with the queries I had already created to write functions to pull data from the database for each use case.

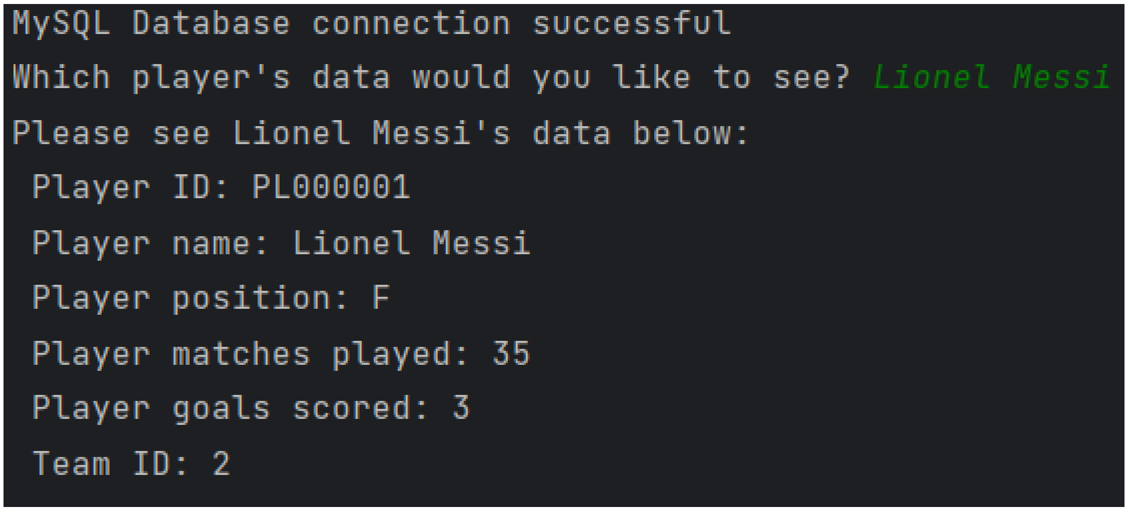
Modifications were made to each query to allow for a users search term to be queried for by adding %s where appropriate. For example in the first query, we are looking for player data, so the user is asked for a player name. Therefore in the query we see the statement `WHERE player_name=%s`. The `%s` is replaced by the search term entered by the user in the input statement.

THe result of the functions are tuples which we can then use the index to put out the exactly fields required and present them in a readable way to the user.

4.1 Export data about football players

```python
#Function to export the data of a specific player in the database
def showplayerinfo(player_name):
    query = """ SELECT * FROM players WHERE player_name=%s """
    cursor.execute(query, (player_name,))
    result = cursor.fetchall()
    return result


player_name = input("Which player's data would you like to see? ")  #asking the user for the player that they wish to search for

data = showplayerinfo(player_name)  #the result of this function is a tuple, containing the data under each column for that searched for player

print(f"""Please see {player_name}'s data below:\n
        Player ID: {data[0][0]}
        Player name: {data[0][1]}
        Player position: {data[0][2]}
        Player matches played: {data[0][3]}
        Player goals scored: {data[0][4]}
        Team ID: {data[0][5]}""")
```

Output:



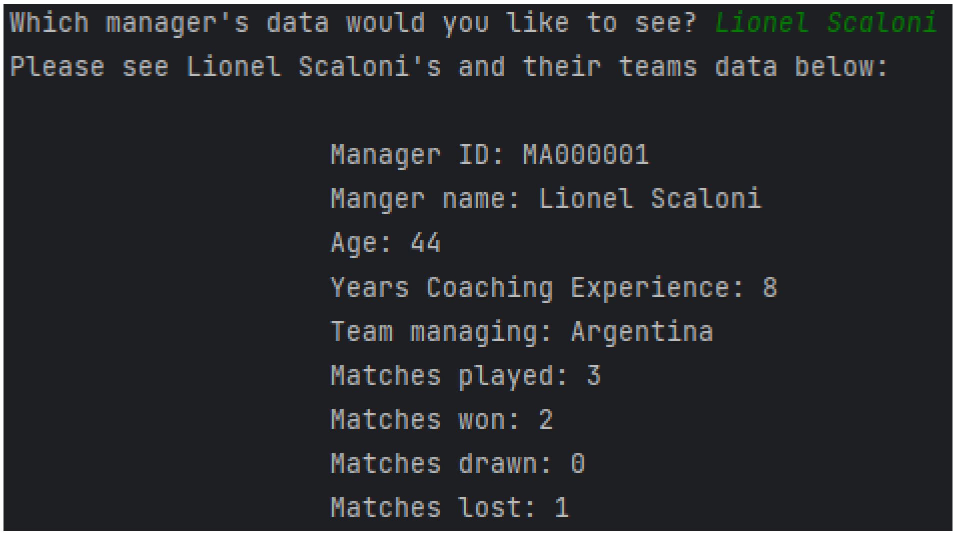
```
MySQL Database connection successful
Which player's data would you like to see? Lionel Messi
Please see Lionel Messi's data below:
 Player ID: PL000001
 Player name: Lionel Messi
 Player position: F
 Player matches played: 35
 Player goals scored: 3
 Team ID: 2
```

4.2 Export data about head coach managers and their associated teams

This function is querying the database for data about a specfic manager and the team they manage.

```
def managerandteaminfo(manager_name):
    query = """ SELECT * FROM managers, teams WHERE managers.team_id=teams.team_id
                AND manager_name=%s"""
    cursor.execute(query, (manager_name,))
    result = cursor.fetchall()
    return result

manager_name = input("Which manager's data would you like to see? ")

data = managerandteaminfo(manager_name)

print(f"""Please see {manager_name}'s and their teams data below:\n
            Manager ID: {data[0][0]}
            Manger name: {data[0][1]}
            Age: {data[0][2]}
            Years Coaching Experience: {data[0][3]}
            Team managing: {data[0][6]}
            _Group Stage Record_
            Matches played: {data[0][7]}
            Matches won: {data[0][8]}
            Matches drawn: {data[0][9]}
            Matches lost: {data[0][10]}
            """)
```

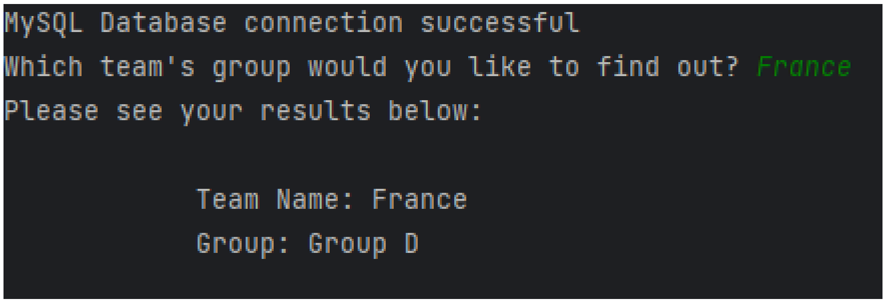
The result of which is this:



```
Which manager's data would you like to see? Lionel Scaloni
Please see Lionel Scaloni's and their teams data below:

                    Manager ID: MA000001
                    Manger name: Lionel Scaloni
                    Age: 44
                    Years Coaching Experience: 8
                    Team managing: Argentina
                    Matches played: 3
                    Matches won: 2
                    Matches drawn: 0
                    Matches lost: 1
```

4.3 Export data to show the group of each team

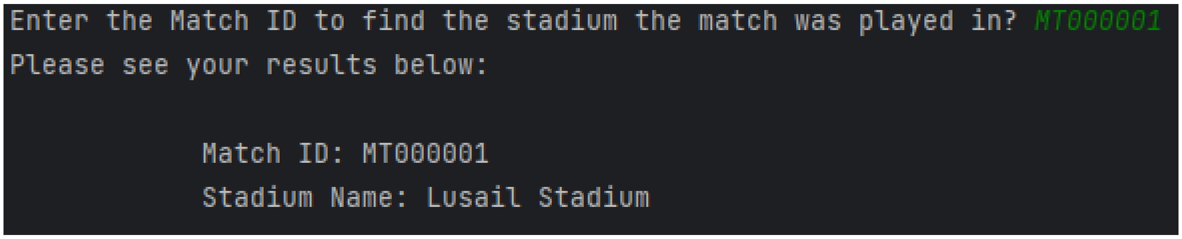
```
def find_group(team_name):
    query = """SELECT teams.team_country, groups_.group_name
               FROM teams, groups_
               WHERE teams.group_id=groups_.group_id
               AND teams.team_country=%s"""
    cursor.execute(query, (team_name,))
    result = cursor.fetchall()
    return result

team_name = input("Which team's group would you like to find out? ")

data = find_group(team_name)

print(f"""Please see your results below:\n
            Team Name: {data[0][0]}
            Group: {data[0][1]}
            """)
```



```
MySQL Database connection successful
Which team's group would you like to find out? France
Please see your results below:

            Team Name: France
            Group: Group D
```

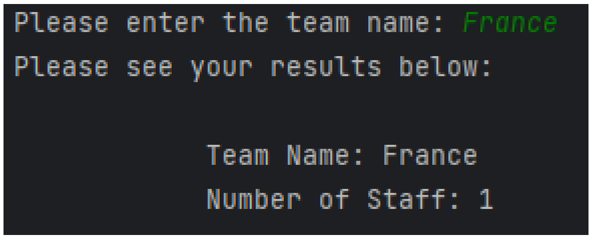
4.4 Export data to show the football stadium for each game

```
def find_match_stadium(match_id):
    query = """SELECT m.match_id, s.stadium_name
    FROM matches AS m, stadiums AS s
    WHERE m.stadium_id=s.stadium_id AND m.match_id=%s"""
    cursor.execute(query, (match_id,))
    result = cursor.fetchall()
    return result

match_id = input("Enter the Match ID to find the stadium the match was played in?
")

data = find_match_stadium(match_id)

print(f"""Please see your results below:\n
            Match ID: {data[0][0]}
            Stadium Name: {data[0][1]}
            """)
```



```
Enter the Match ID to find the stadium the match was played in? MT000001
Please see your results below:

            Match ID: MT000001
            Stadium Name: Lusail Stadium
```

4.5 Export data to count how many staff members are in the database per team

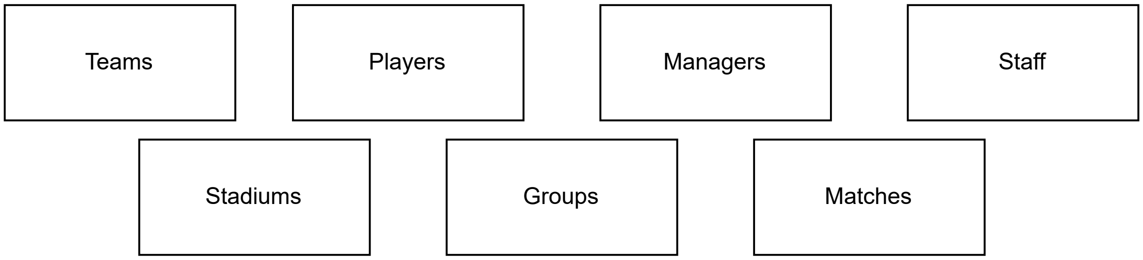


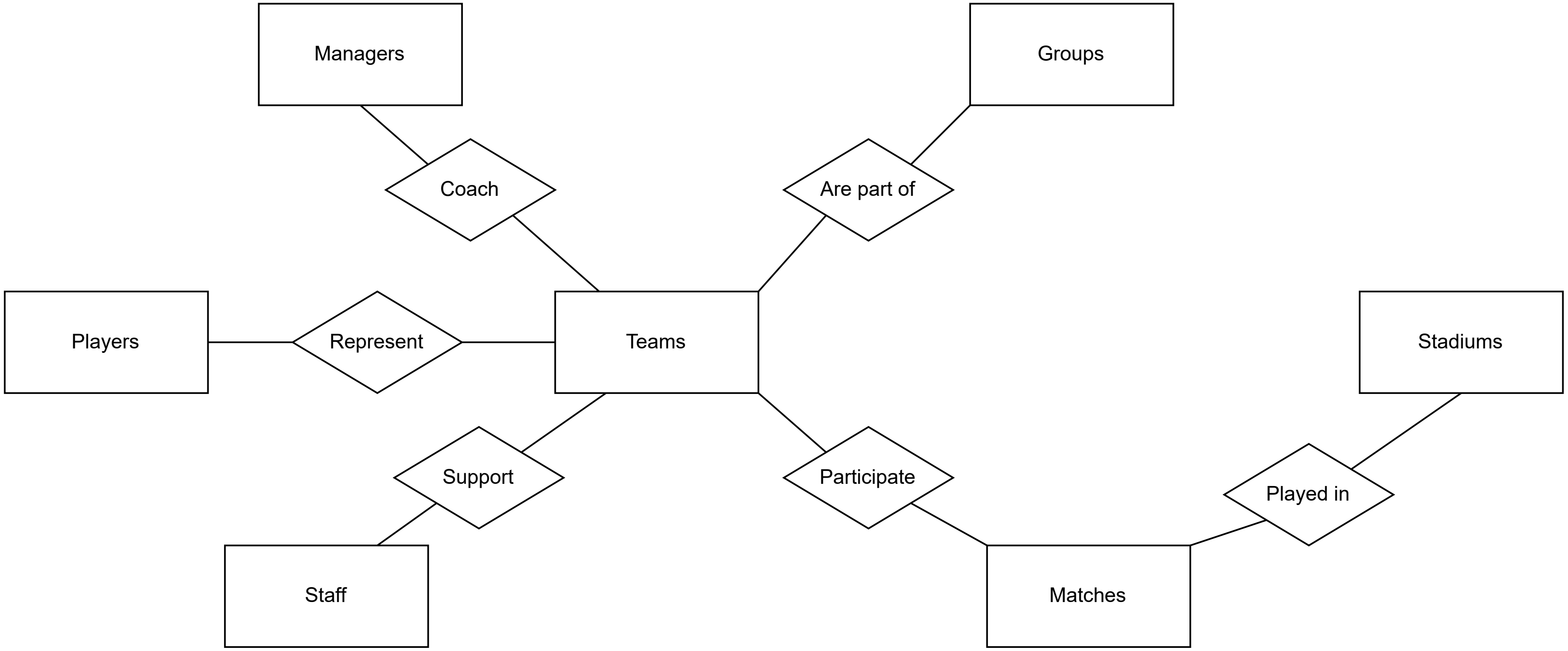
```
Please enter the team name: France
Please see your results below:

            Team Name: France
            Number of Staff: 1
```

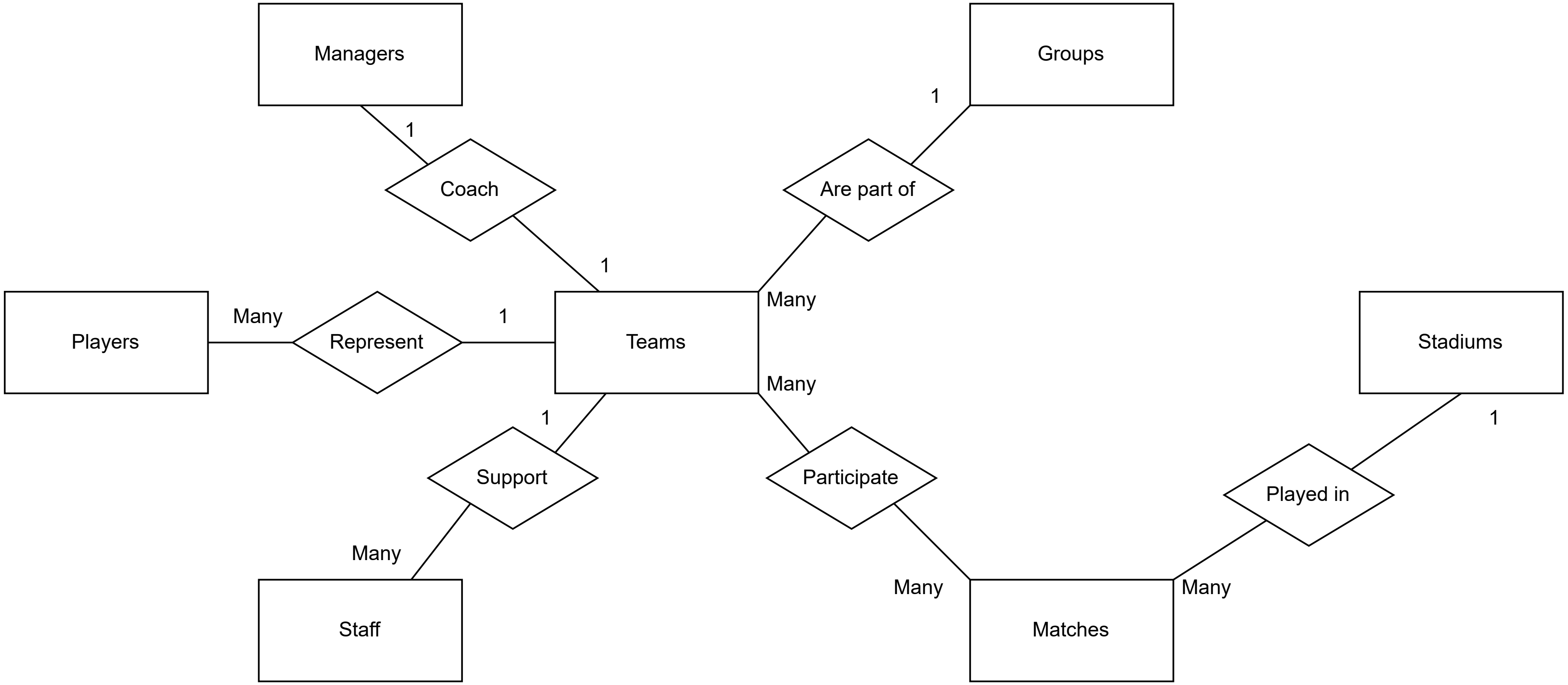
```
def number_of_staff(team_name):
    query = """SELECT t.team_country, COUNT(s.staff_id) AS number_of_staff
    FROM staff AS s, teams AS t    WHERE s.team_id = t.team_id AND
t.team_country=%s    GROUP BY s.staff_id;"""
    cursor.execute(query, (team_name,))
    result = cursor.fetchall()
    return result

team_name = input("Please enter the team name: ")
```

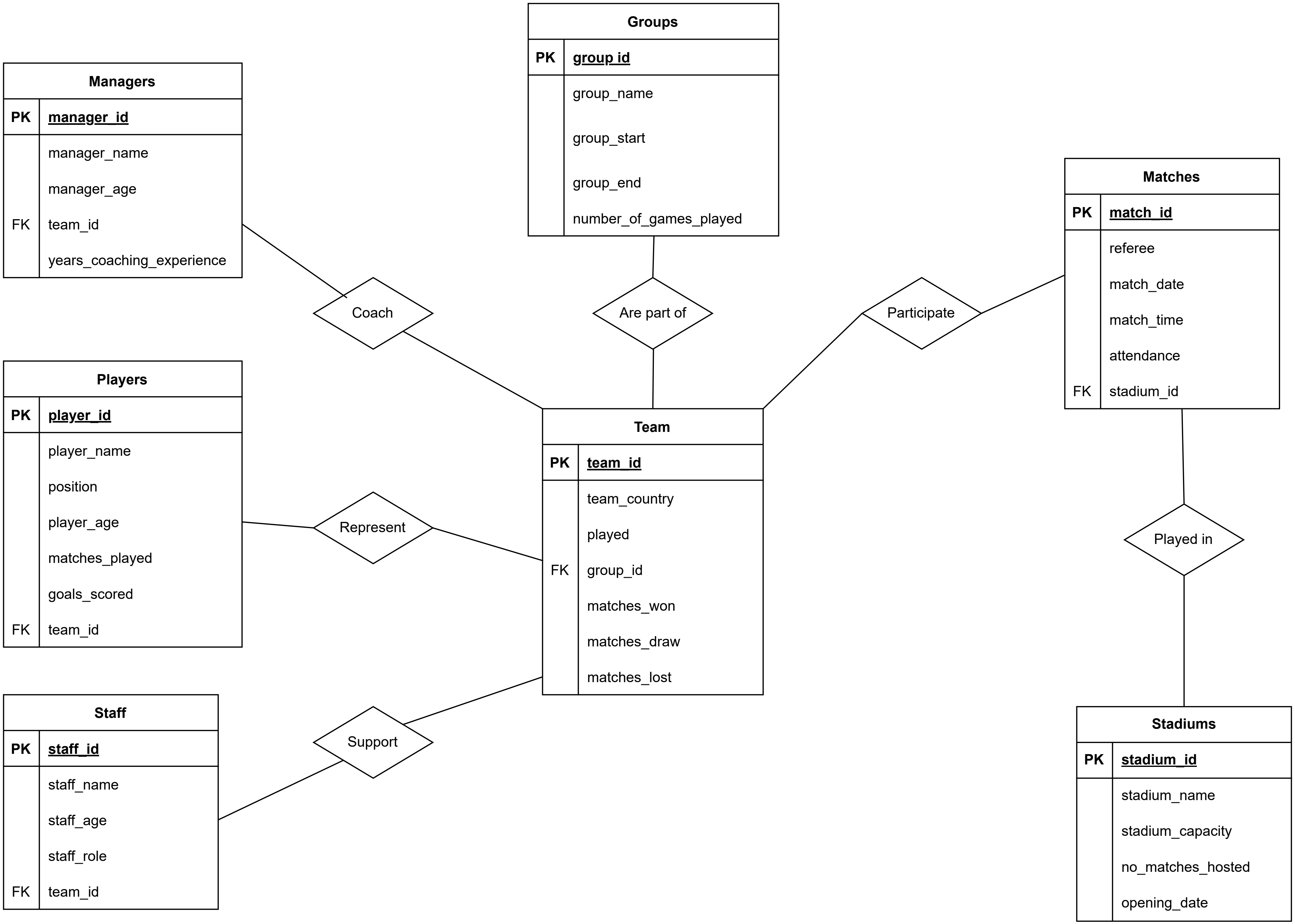
```
data = number_of_staff(team_name)

print(f"""Please see your results below:\n
            Team Name: {data[0][0]}
            Number of Staff: {data[0][1]}
            """)
```

Appendix: Full ER Model Diagrams




Managers
Groups
Coach
Are part of
Players
Represent
Teams
Stadiums
Support
Participate
Played in
Staff
Matches


Managers
Groups
1
1
Coach
Are part of
1
Players
Many
Represent
1
Teams
Many
Stadiums
Many
1
1
Support
Participate
Played in
Many
Many
Many
Staff
Matches



Managers

PK	**manager_id**
	manager_name
	manager_age
FK	team_id
	years_coaching_experience

Groups

PK	**group id**
	group_name
	group_start
	group_end
	number_of_games_played

Matches

PK	**match_id**
	referee
	match_date
	match_time
	attendance
FK	stadium_id

Players

PK	**player_id**
	player_name
	position
	player_age
	matches_played
	goals_scored
FK	team_id

Team

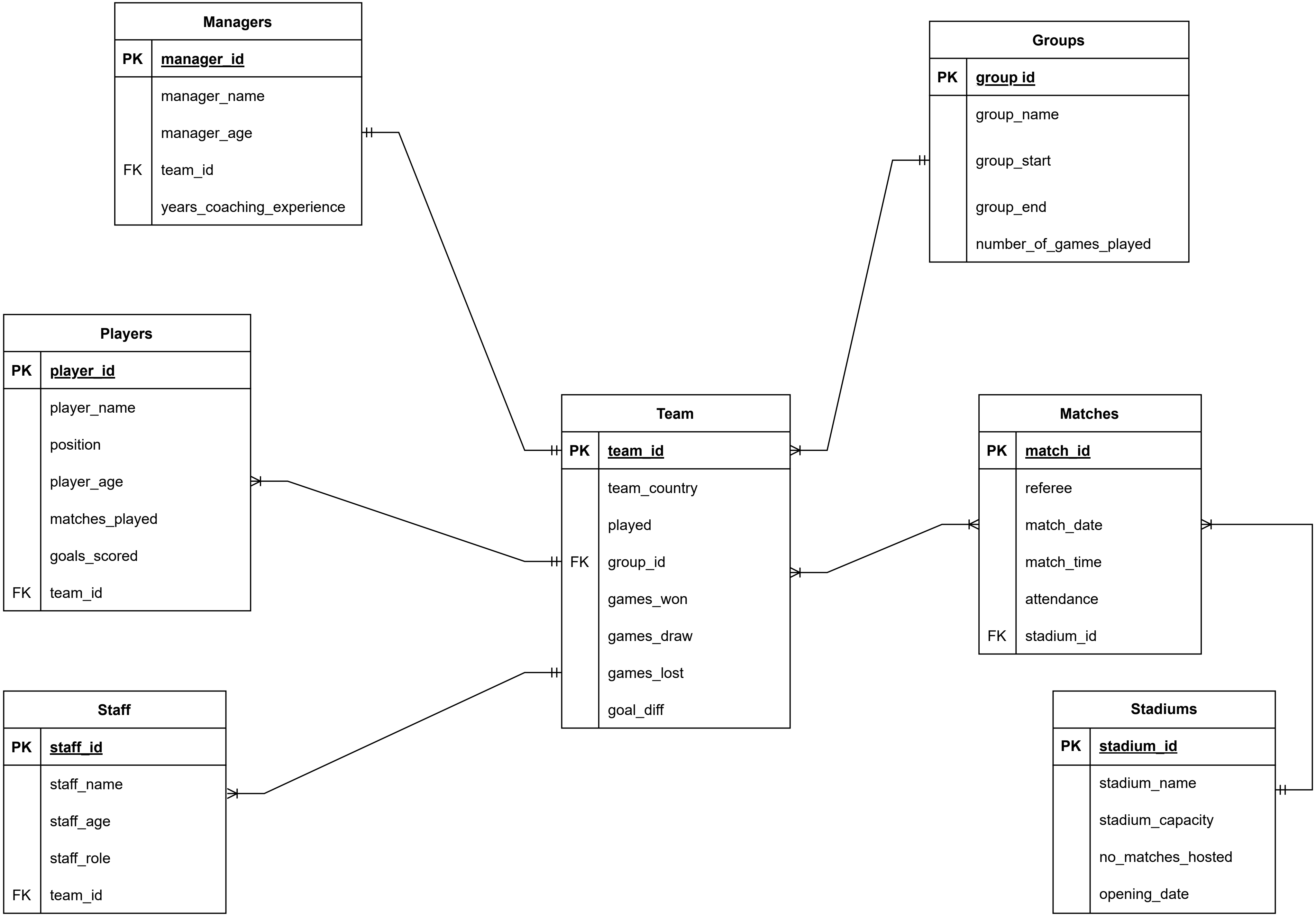
PK	**team_id**
	team_country
	played
FK	group_id
	matches_won
	matches_draw
	matches_lost

Staff

PK	**staff_id**
	staff_name
	staff_age
	staff_role
FK	team_id

Stadiums

PK	**stadium_id**
	stadium_name
	stadium_capacity
	no_matches_hosted
	opening_date



	Managers
PK	**manager_id**
	manager_name
	manager_age
FK	team_id
	years_coaching_experience

	Groups
PK	**group_id**
	group_name
	group_start
	group_end
	number_of_games_played

	Players
PK	**player_id**
	player_name
	position
	player_age
	matches_played
	goals_scored
FK	team_id

	Team
PK	**team_id**
	team_country
	played
FK	group_id
	games_won
	games_draw
	games_lost
	goal_diff

	Matches
PK	**match_id**
	referee
	match_date
	match_time
	attendance
FK	stadium_id

	Staff
PK	**staff_id**
	staff_name
	staff_age
	staff_role
FK	team_id

	Stadiums
PK	**stadium_id**
	stadium_name
	stadium_capacity
	no_matches_hosted
	opening_date