Information Management Project – Fantasy Football Database

Section I - Brief Description of Database

For my project I decided to create a fantasy football database for a five-a-side league. The idea is that a person can create an account for the game and be given a manager ID, they can then create a team and be given a team ID, then they can pick players from a player database and add five players to their team. If the manager has played in previous years of the game they will have a manager history table which will include the year and the points they got that year. Similarly, if a player has been in the five-a-side league either for one or more seasons, their point history will be available in a player history table. In this database there is a managers table, for people who would actually be playing the game, a manager history table, a players table, for players the managers might pick, a player history table, a leagues table, for managers to compete against one another, and a team table for a manager's team.

Description of tables

Manager table: This table includes the manager's ID number, their points, the league they are in and their position in that league.

Manager history table: This table includes the manager's ID, the year and the points they got that year.

Player table: This table includes the player's ID, the player's names, the team they play for, their position, the number of goals they got, the number of assists they got and the number of clean sheets they got.

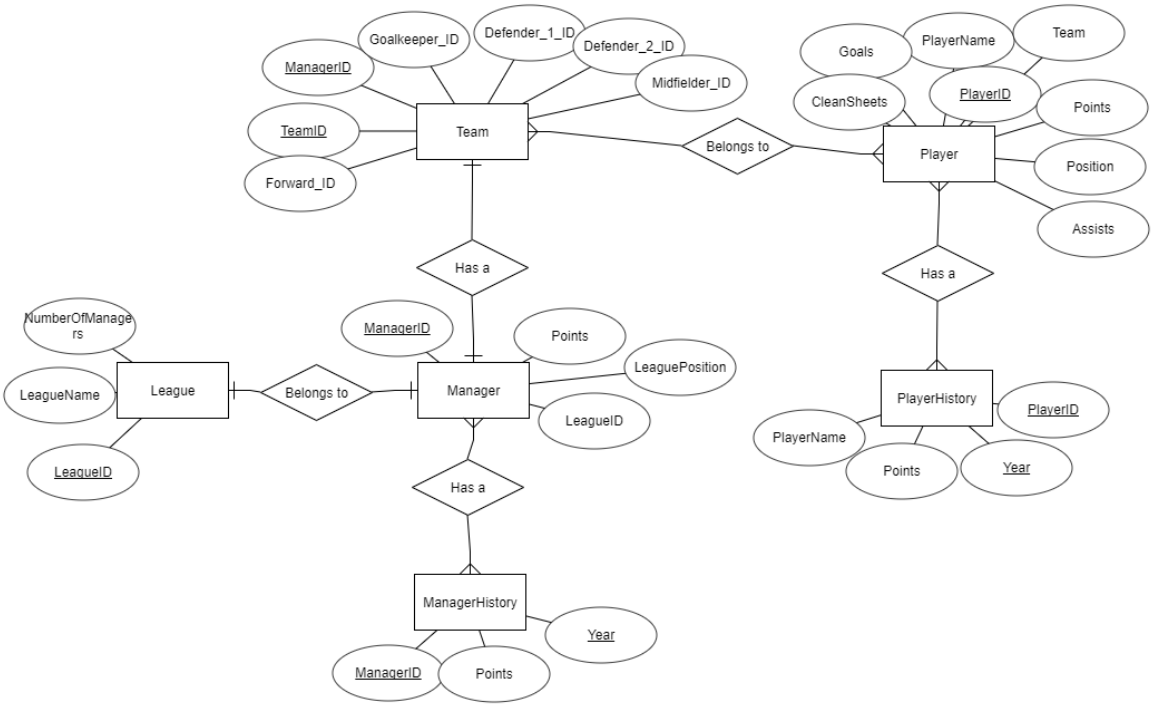
Player history table: This table includes the player' ID, the player's name, the year and the points they got that year.

Team table: This table includes the team ID, the manager ID associated with that team and the player ID of the players that the manager has picked.

League table: This table includes the league ID, the league name and the number of managers in that league.

Section II - Diagrams

Entity Relationship Diagram



One-to-one relationships are represented by a vertical line.

Many-to-many relationships are represented by triangles.

Relational Schema

Manager

ManagerID
Points
LeagueID
LeaguePosition

Manager_History

ManagerID
Year
Points

Team

TeamID
ManagerID
Goalkeeper_ID
Defender_1_ID
Defender_2_ID
Midfielder_ID
Forward_ID

League

LeagueID
LeagueName
NumberOfPlayers

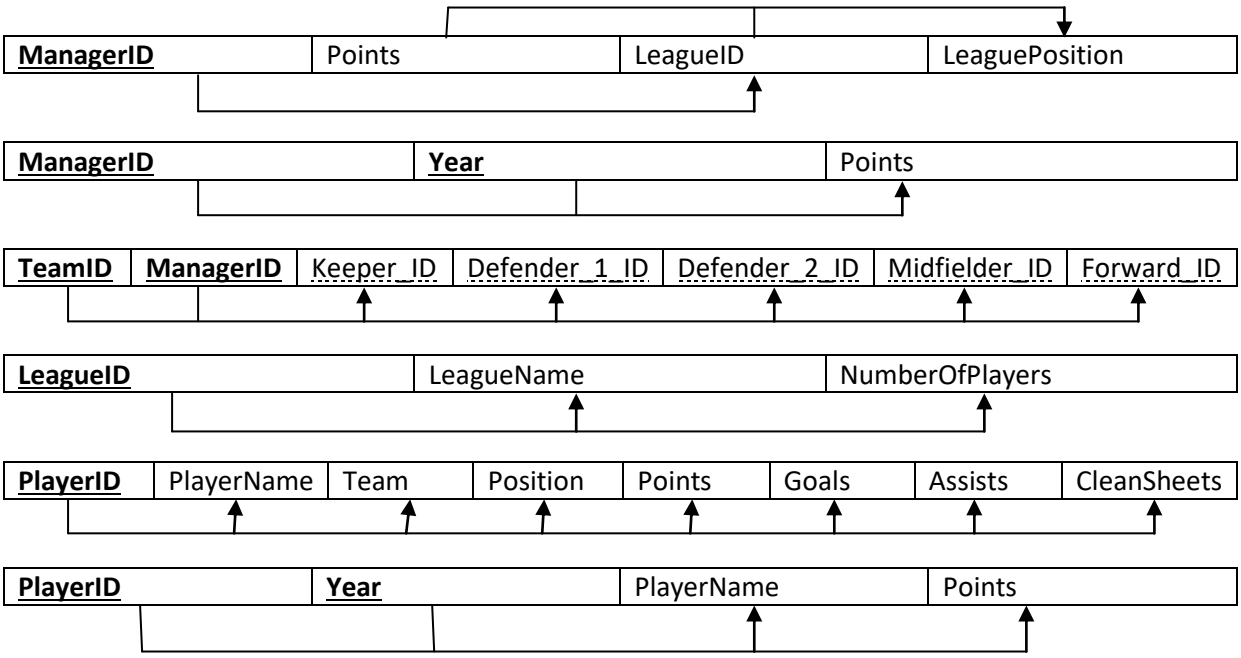
Player

PlayerID
PlayerName
Team
Position
Points
Goals
Assists
CleanSheets

Player_History

PlayerID
Year
PlayerName
Points

Functional Dependency Diagram

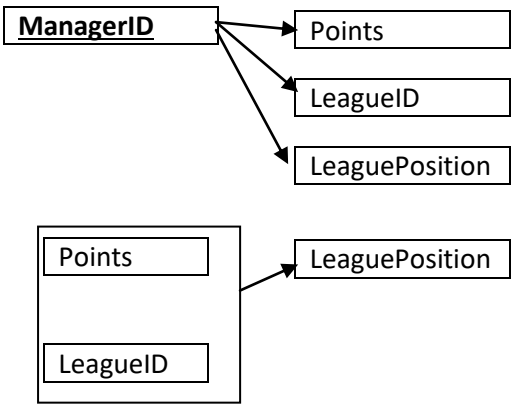


Primary keys are in bold and underlined

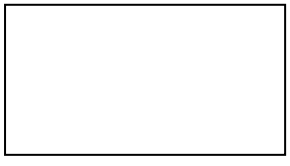
Foreign keys are underlined with a dotted line.

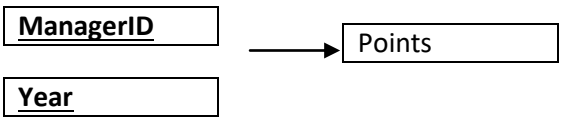
List of normalised relations:

Manager table:

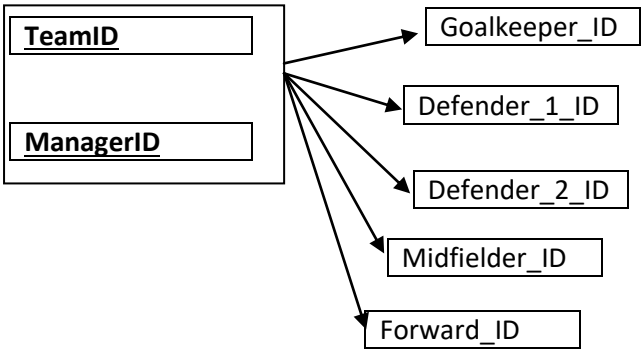


Manager history table:

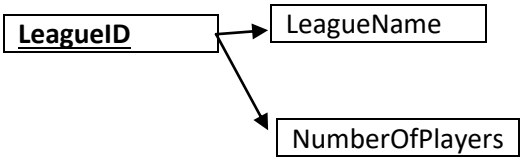




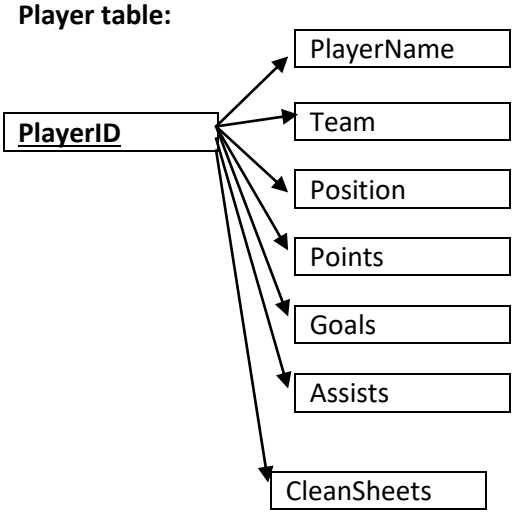
Team table:



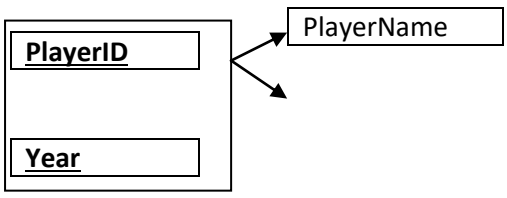
League table:



Player table:



Player history table:



Points

Section III - Examples of database security commands for access and brief explanation of security policy intended

Database Security Commands:

The admins of the database would have all privileges to all of the tables.

```
GRANT read_privilege, modification_privilege, reference_privilege ON
manager_table, manager_history, player_table, player_history, league, team
TO admin;
```

The users of the database (the managers) would only have access to read privileges so that they could select data from the database only.

```
GRANT read_privilege ON manager_table, manager_history, player_table,
player_history, league, team TO user;
```

Database Security Policy:

For the database it should be possible for the manager of a team to look at their history and the history of other managers, their league, their team and other manager's teams. They should not, however, be able to change the values in these tables. In these cases it would be important to create views of these tables so that managers could look at them but not change any values. This would mean managers would have a read privilege on tables.

As for modifying the tables, only an admin or automated system which updates the points after a game should be able to change the values in a given table. Therefore, modification priveleges should be given to admins of the database.

Only admins should be able to grant and revoke privileges to other admins and no one else. Since privileges can be given to other people once a person has a privilege it is important that only admins are allowed to have privileges.

Section IV – Examples of view, table join, select, update and trigger commands

View Creation and Table Joins:

Views were created in this database for the manager table and between the manager and league tables using the table join operation. The data selected for the manager view was the ManagerID and their Points. For the manager and league view, the data selected was the ManagerID, their Points and their LeaguePosition from the manager table which was then joined with the data LeagueName and NumberOfManagers from the league table.

```
select `a`.`ManagerID` AS `ManagerID`,`a`.`Points` AS
`Points`,`a`.`LeaguePosition` AS `LeaguePosition`,`b`.`LeagueID` AS
`LeagueID`,`b`.`LeagueName` AS `LeagueName`,`b`.`NumberOfManagers` AS
`NumberOfManagers` from (`fantasy football database`.`manager_table` `a` join
`fantasy football database`.`league` `b` on((`a`.`LeagueID` = `b`.`LeagueID`)))
```

Relational Select:

A relational select in the database would select some or all of the columns from a particular table. One such select statement could select eh PlayerID, PlayerName, Team and Points from the player_table.

```
SELECT ‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Points’ FROM player_table;
```

Update operations:

An update operation on the database could be performed on one of the players. After a game if they got a goal, assist or clean sheet that piece of data would get updated and their tally for that piece of data would also go up.

UPDATE ‘player_table’ SET ‘Points’ = ‘160’, ‘Goals’ = ‘9’ WHERE ‘player_table’.’PlayerID’ = 214783647

Trigger command:

A trigger command was also created in the database for whenever a new football player is added to the database. When a new player is added they are also inserted into the player history table with their points and the year ‘2018’.

```
BEGIN
INSERT INTO player_history VALUES(NEW.PlayerID, NEW.PlayerName, 2018, NEW.Points);
END
```

Section V - Appendix

List of data definition commands used to create database:

```
create database ‘fantasy football database’;

create table ‘fantasy football database’.’manager_table’ ( ‘ManagerID’ INT(10) NOT NULL, ‘Points’ INT(10) NOT NULL, ‘LeagueID’ INT(10) NOT NULL, ‘LeaguePosition’ INT(10) NOT NULL, PRIMARY KEY (‘ManagerID’)) ENGINE = InoDB;

create table ‘fantasy football database’.’manager_history’ ( ‘ManagerID’ INT(10) NOT NULL, ‘Year’ INT(10) NOT NULL, ‘Points’ INT(10) NOT NULL, PRIMARY KEY (‘ManagerID’, ‘Year’)) ENGINE = InoDB;

create table ‘fantasy football database’.’player_table’ ( ‘PlayerID’ INT(10) NOT NULL, ‘PlayerName’ TEXT NOT NULL, ‘Team’ TEXT NOT NULL, ‘Position’ TEXT NOT NULL, ‘Points’ INT(10) NOT NULL, ‘Goals’ INT(10) NOT NULL, ‘Assists’ INT(10) NOT NULL, ‘CleanSheets’ INT(10) NOT NULL, PRIMARY KEY (‘PlayerID’)) ENGINE = InoDB;

create table ‘fantasy football database’.’player_history’ ( ‘PlayerID’ INT(10) NOT NULL, ‘PlayerName’ TEXT NOT NULL, ‘Year’ INT(10) NOT NULL, ‘Points’ INT(10) NOT NULL, PRIMARY KEY (‘PlayerID’, ‘Year’)) ENGINE = InoDB;

create table ‘fantasy football database’.’league’ ( ‘LeagueID’ INT(10) NOT NULL, ‘LeagueName’ TEXT NOT NULL, ‘NumberOfManagers’ INT(10) NOT NULL, PRIMARY KEY (‘LeagueID’)) ENGINE = InoDB;
```

```
create table ‘fantasy football database’.’team’ ( ‘TeamID’ INT(10) NOT NULL, ‘ManagerID’ INT(10) NOT NULL, ‘Goalkeeper_ID’ INT(10) NOT NULL, ‘Defender_1_ID’ INT(10) NOT NULL, ‘Defender_2_ID’ INT(10) NOT NULL, ‘Midfielder_ID’ INT(10) NOT NULL, ‘Forward_ID’ INT(10) NOT NULL, PRIMARY KEY (‘PlayerID’)) ENGINE = InoDB;

select `a`.`ManagerID` AS `ManagerID`,`a`.`Points` AS `Points`,`a`.`LeaguePosition` AS `LeaguePosition`,`b`.`LeagueID` AS `LeagueID`,`b`.`LeagueName` AS `LeagueName`,`b`.`NumberOfManagers` AS `NumberOfManagers` from (`fantasy football database`.`manager_table` `a` join `fantasy football database`.`league` `b` on((`a`.`LeagueID` = `b`.`LeagueID`)))

select `fantasy football database`.`manager_table`.`ManagerID` AS `ManagerID`,`fantasy football database`.`manager_table`.`Points` AS `Points` from `fantasy football database`.`manager_table`


BEGIN

INSERT INTO player_history VALUES(NEW.PlayerID, NEW.PlayerName, 2018, NEW.Points);

END
```

List of database population commands:

```
INSER INTO ‘manager_table’ (‘ManagerID’, ‘Points’, ‘LeagueID’, ‘LeaguePosition’) VALUES (‘52908501’, ‘970’, ‘1422919111’, ‘9’)
INSER INTO ‘manager_table’ (‘ManagerID’, ‘Points’, ‘LeagueID’, ‘LeaguePosition’) VALUES (‘222111334’, ‘1204’, ‘1234567890’, ‘7’)

INSER INTO ‘manager_table’ (‘ManagerID’, ‘Points’, ‘LeagueID’, ‘LeaguePosition’) VALUES (‘437918722’, ‘1333’, ‘1422919111’, ‘4’)

INSER INTO ‘manager_table’ (‘ManagerID’, ‘Points’, ‘LeagueID’, ‘LeaguePosition’) VALUES (‘1234561231’, ‘1531’, ‘1111111111’, ‘3’)

INSER INTO ‘manager_table’ (‘ManagerID’, ‘Points’, ‘LeagueID’, ‘LeaguePosition’) VALUES (‘2147483647’, ‘1690’, ‘1111111111’, ‘1’)

INSER INTO ‘manager_history’ (‘ManagerID’, ‘Year’, ‘Points’) VALUES (‘52908501’, ‘2016’, ‘2130’)

INSER INTO ‘manager_history’ (‘ManagerID’, ‘Year’, ‘Points’) VALUES (‘222111334’, ‘2016’, ‘1890’)
```

```
INSER INTO ‘manager_history’ (‘ManagerID’, ‘Year’, ‘Points’) VALUES
(‘222111334’, ‘2017’, ‘2150’)

INSER INTO ‘manager_history’ (‘ManagerID’, ‘Year’, ‘Points’) VALUES
(‘1234561231’, ‘2017’, ‘2248’)

INSER INTO ‘manager_history’ (‘ManagerID’, ‘Year’, ‘Points’) VALUES
(‘2147483647’, ‘2015’, ‘1907’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘112937012’, ‘Christian
Eriksen’, ‘Tottenham Hotspur’, ‘Midfielder’, ‘199’, ‘11’, ‘10’, ‘17’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘312636092’, ‘Matt
Doherty’, ‘Wolverhampton Wanderers’, ‘Defender’, ‘95’, ‘5’, ‘1’, ‘6’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘343923223’, ‘Callum
Paterson’, ‘Cardiff City FC’, ‘Midfielder’, ‘41’, ‘3’, ‘1’, ‘1’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘1122334455’, ‘Eden
Hazard’, ‘Chelsea’, ‘Forward’, ‘211’, ‘13’, ‘10’, ‘5’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘1231231234’, ‘Mo Salah’,
‘Liverpool’, ‘Forward’, ‘132’, ‘7’, ‘4’, ‘8’)

INSER INTO ‘player_table’ (‘PlayerID’, ‘PlayerName’, ‘Team’, ‘Position’,
‘Points’, ‘Goals’, ‘Assists’, ‘CleanSheets’) VALUES (‘2147483647’, ‘Sergio
Aguero’, ‘Manchester City’, ‘Forward’, ‘156’, ‘8’, ‘6’, ‘10’)

INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘112937012’, ‘Christian Eriksen’, ‘2017’ ‘218’)

INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘312636092’, ‘Matt Doherty’, ‘2015’ ‘124’)

INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘1122334455’, ‘Eden Hazard’, ‘2017’ ‘173’)

INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘1231231234’, ‘Mo Salah’, ‘2017’ ‘303’)

INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘2147483647’, ‘Sergio Aguero’, ‘2017’ ‘269’)
```

```
INSER INTO ‘player_history’ (‘PlayerID’, ‘PlayerName’, ‘Year’, ‘Points’)
VALUES (‘2147483647’, ‘Sergio Aguero’, ‘2017’ ‘218’)

INSER INTO ‘team’ (‘TeamID’, ‘ManagerID’, ‘Goalkeeper_ID’, ‘Defender_1_ID’,
‘Defender_2_ID’, ‘Midfielder_ID’, ‘Forward_ID’) VALUES (‘13’, ‘52908501’,
‘10001’, ‘20001’, ‘20002’, ‘30001’, ‘1122334455’)

INSER INTO ‘team’ (‘TeamID’, ‘ManagerID’, ‘Goalkeeper_ID’, ‘Defender_1_ID’,
‘Defender_2_ID’, ‘Midfielder_ID’, ‘Forward_ID’) VALUES (‘37’, ‘222111334’,
‘10002’, ‘312636092’, ‘20003’, ‘343923223’, ‘40003’)

INSER INTO ‘team’ (‘TeamID’, ‘ManagerID’, ‘Goalkeeper_ID’, ‘Defender_1_ID’,
‘Defender_2_ID’, ‘Midfielder_ID’, ‘Forward_ID’) VALUES (‘82’, ‘1234561231’,
‘10003’, ‘20004’, ‘20005’, ‘112937012’, ‘1122334455’)

INSER INTO ‘team’ (‘TeamID’, ‘ManagerID’, ‘Goalkeeper_ID’, ‘Defender_1_ID’,
‘Defender_2_ID’, ‘Midfielder_ID’, ‘Forward_ID’) VALUES (‘90’, ‘437918722’,
‘10004’, ‘312636092’, ‘20004’, ‘112937012’, ‘2147483647’)

INSER INTO ‘team’ (‘TeamID’, ‘ManagerID’, ‘Goalkeeper_ID’, ‘Defender_1_ID’,
‘Defender_2_ID’, ‘Midfielder_ID’, ‘Forward_ID’) VALUES (‘132’, ‘2147483647’,
‘10005’, ‘20003’, ‘20001’, ‘112937012’, ‘40001’)

INSER INTO ‘league’ (‘LeagueID’, ‘LeagueName’, ‘NumberOfManagers’) VALUES
(‘1111111111’, ‘League2’, ‘5’)

INSER INTO ‘league’ (‘LeagueID’, ‘LeagueName’, ‘NumberOfManagers’) VALUES
(‘1234567890’, ‘League1’, ‘10’)

INSER INTO ‘league’ (‘LeagueID’, ‘LeagueName’, ‘NumberOfManagers’) VALUES
(‘1422919111’, ‘League5’, ‘18’)
INSER INTO ‘league’ (‘LeagueID’, ‘LeagueName’, ‘NumberOfManagers’) VALUES (‘
1817161514’, ‘League4’, ‘27’)

INSER INTO ‘league’ (‘LeagueID’, ‘LeagueName’, ‘NumberOfManagers’) VALUES
(‘2147483647’, ‘League3’, ‘2’)
```

```
INSERT INTO `player_table` (`PlayerID`, `PlayerName`, `Team`, `Position`, `Points`,`Goals`, `Assists`, `CleanSheets`) VALUES ('1', 'x', 'x', 'y', '1', '1', '1', '1')
```