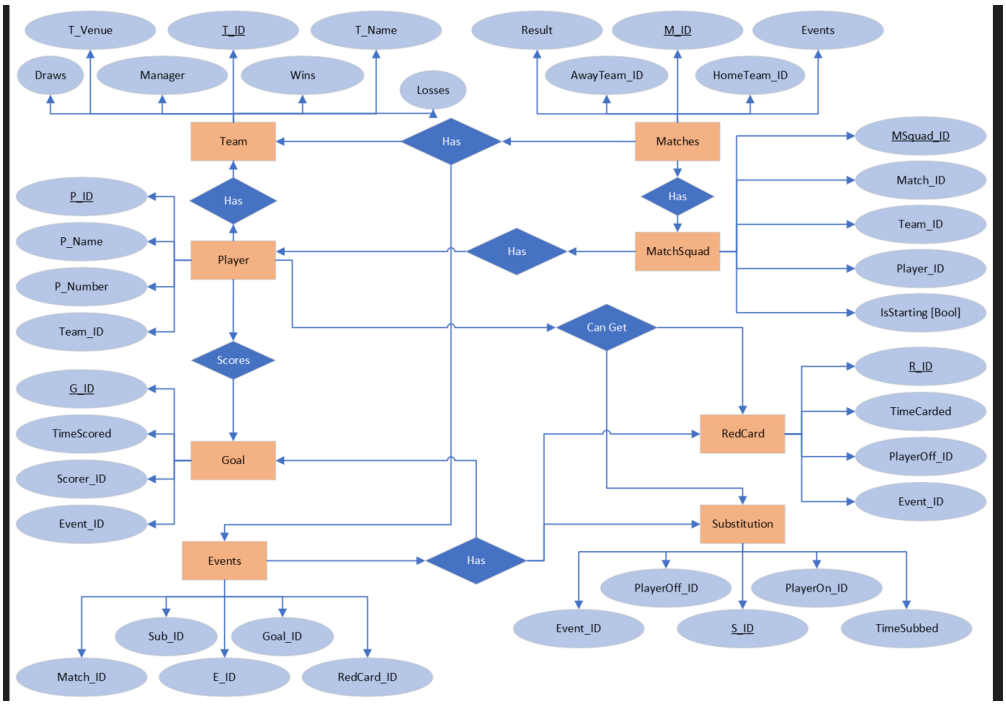
Database Assignment 1

Brian Moyles - 21333461

DESIGN

The database I designed consists of 8 Tables:



- The team table keeps track each team by storing their name, a unique id as a primary key, a venue and a manager. It also stores how many wins, draws and losses they have, which is used to calculate their points.

```sql
-- Team Table
CREATE TABLE team (
    id INT PRIMARY KEY,
    t_name VARCHAR(255) NOT NULL,
    venue VARCHAR(255),
    manager varchar(255),
    wins INT,
    draws INT,
    losses INT
);
```

id	t_name	venue	manager	wins	draws	losses
1	Liverpool	Anfield	Jurgen Klopp	38	0	0
2	Ireland	Aviva Stadium	Stephen Kenny	21	7	10
3	Galway United	Eamonn Deacy Park	John Caulfield	3	17	18
4	Ballina Town FC	Belleek Park	Alan Reilly	35	1	2

- The player table is used to store player details. This includes a unique id as a primary key, the player's name, player number and team_id. The team_id is a foreign key that links to the team table as a player belongs to a team.

```
-- Player Table
CREATE TABLE player (
    id INT PRIMARY KEY,
    player_name VARCHAR(255) UNIQUE NOT NULL ,
    player_number INT,
    team_id INT,
    FOREIGN KEY (team_id) REFERENCES team (id)
);
```

id	player_name	player_number	team_id
101	Steven Gerrard	8	1
102	Mohamed Salah	11	1
103	Virgil van Dijk	4	1
201	Robbie Keane	7	2
202	James McClean	10	2
203	Shane Long	9	2
301	David Cawley	2	3
302	Grank Flavin	15	3
303	Colm TheBest	23	3
401	John Doe	1	4
403	Brian Moyles	67	4
404	Joe Bloggs	12	4

- The matches table stores details about the matches played. Each match has its own unique id as primary key. The home team has its own id and away team has its own id. These are foreign keys which reference the team table to keep track of the teams that are playing in each match. The result scores the match score and the event id is used in reference to the events table which stores more in-depth information about the game

```
-- Match Table
CREATE TABLE Matches (
    id INT PRIMARY KEY,
    hometeamid INT,
    awayteamid INT,
    result VARCHAR(10),
    event_id INT,
    FOREIGN KEY (hometeamid) REFERENCES team (id),
    FOREIGN KEY (awayteamid) REFERENCES team (id),
);
```

id	hometeamid	awayteamid	result	event_id
1	1	2	1-0	101
2	3	4	3-2	102
3	1	3	1-1	103

- The events table keeps track of further information about each match. It uses a match_id as a foreign key to link to the match it refers to. It has its own id as a primary key. goal_id, sub_id and redCard_id is used for those events.

```
-- Events Table
CREATE TABLE events (
    id INT PRIMARY KEY,
    match_id INT,
    goal_id int,
    sub_id int,
    redcard_id int,
    FOREIGN KEY (match_id) REFERENCES Match (id)
);
```

id	match_id	goal_id	sub_id	redcard_id
101	1	201	103	202
102	1	202	404	203
103	2	101	301	303

- The goal table is used to track every goal scored in matches by using a foreign key to the event table (which is linked to the specific match) it also uses a scored id as a foreign key to the player table to track who scored the goal. The time scored is stored in a time column. Every goal has its own unique id as a primary key.

```sql
-- Goal Table
CREATE TABLE goal (
    id INT PRIMARY KEY,
    event_id INT,
    scorer_id INT,
    timescored TIME,
    FOREIGN KEY (event_id) REFERENCES events (id),
    FOREIGN KEY (scorer_id) REFERENCES player (id)
);
```

id	event_id	scorer_id	timescored
202	102	403	30:00:00
203	103	302	47:00:00
204	102	201	90:00:00
403	101	102	12:34:00

- The substitution table is used to track substitutions in a match. Like the goal table, it uses a foreign key to the event table to keep track of which match it is in. and id for the player on and player off is also used as foreign keys to the player table. This allows the tracking of which player is involved in the substitution. The time is also stored and each substitution has a unique id as a primary key.

```sql
-- Substitution Table
CREATE TABLE Substitution (
    id INT PRIMARY KEY,
    event_id INT,
    playeron_id INT,
    playeroff_id INT,
    timesub TIME,
    FOREIGN KEY (event_id) REFERENCES events (id),
    FOREIGN KEY (playeron_id) REFERENCES Player (id),
    FOREIGN KEY (playeroff_id) REFERENCES Player (id)
);
```

playeron_id	timesub	id	playeroff_id	event_id
302	55:00:00	301	301	103
303	65:00:00	302	302	103
203	75:00:00	303	202	102

- The red card table works the same as the substitution and goal tables. It has its own unique id as primary key. It has a foreign key to the events table and player table to track which match and which player it refers to. It also tracks the time at which the red card happens.

```sql
-- RedCard Table
CREATE TABLE RedCard (
    id INT PRIMARY KEY,
    event_id INT,
    playeroff_id INT,
    timecarded TIME,
    FOREIGN KEY (event_id) REFERENCES events (id),
    FOREIGN KEY (playeroff_id) REFERENCES Player (id)
);
```

id	event_id	playeroff_id	timecarded
401	101	202	10:00:00
402	102	202	25:00:00
403	103	301	60:00:00

- The matchsquad table is used to track the players in the squad for a particular game. Each matchsquad has its own unique id as a primary key. It uses foreign keys for reference to which match the squad is used in, which team the squad belongs to and what players are in the squad. It also uses a Boolean to determine if the player is in the starting 11 as not every player in the squad is in the starting 11.

```sql
-- MatchSquad Table
CREATE TABLE MatchSquad (
    id INT PRIMARY KEY,
    match_id INT,
    team_id INT,
    player_id INT,
    IsStarting BOOLEAN,
    FOREIGN KEY (match_id) REFERENCES Matches (id),
    FOREIGN KEY (team_id) REFERENCES Team (id),
    FOREIGN KEY (player_id) REFERENCES Player (id)
);
```

id	match_id	team_id	player_id	IsStarting
501	1	1	101	1
502	1	1	102	1
503	2	1	103	1
504	2	2	201	1
505	2	2	202	1
506	2	2	203	1
507	3	1	101	1
508	3	1	103	1
509	3	3	302	1
510	3	3	303	1

Relationships

One to one (1:1)

- Team with Matchsquad – 1 team has 1 squad and a squad is for 1 team

One to many (1: n)

- Player with team – a player has 1 team and 1 team has many players

Many to One (n: 1)

- Events with matches - Many events can happen in one match but each event is associated with 1 match
- Goal with Events – many events can have 1 goal
- Goal with Players – many players can have goals but each goal is associated with 1 player
- Substitution with Events and players - many events can have 1 sub and many players can have substitutions but each sub is associated with 1 player on and off
- Red Cards with Events and players - many events can have 1 red card and many players can have red cards but each red card is associated with 1 player off
- Match Squad with matches, team and player - Each squad entry is associated with one match, one team, and one player. But a player can be in many squads, a team has many squads and matches have many squads.

Many to Many (n: n)

- Matches with team - each match involves two teams, and each team can be part of multiple matches.

Issues, redundancies and observations

- Having a player id instead of using player name in the match squad maintains normalization and reduces redundancy
- Event table was added to reduce redundancy and simplify queries. Initially all events were stored in the matches table, but adding the event table allows for centralization and males it easier to manage the data without the need to duplicate the information across multiple tables.
- Initially, I had a foreign key in goal to relate to the id of the team of the player that scored. This was redundant as it can be tracked via the scorer to se what team he plays for.
- The goal, substitution and red card tables allow for normalization by separating the events into their own tables. There is now dedicated storage for each event type. This improves data integrity and reduces redundancy

Assumptions

- Result in matches table is a VARCHAR(10) but will store the match score result, e.g., 3-1.
- The time values are stored in terms of the time scored in the match
- isStarting Boolean is true if the player is in the starting 11 and false if not
- A player can only be in one team
- Wins are worth 3 points, draws worth 1, losses worth 0 when calculating points

Queries

List all players playing for a given team.

```
SELECT *
FROM player
WHERE team_id = (SELECT id FROM team WHERE t_name = 'Liverpool')
```

id	player_name	player_number	team_id
101	Steven Gerrard	8	1
102	Mohamed Salah	11	1
103	Virgil van Dijk	4	1

List all players who have scored in a given game

```
SELECT player.player_name
FROM player
INNER JOIN goal ON player.id = goal.scorer_id
WHERE goal.event_id = 102
```

player_name
Brian Moyles
Robbie Keane

List the top 5 goal scorers in the league

```
SELECT player.player_name, COUNT(goal.id) AS goals_scored
FROM player
LEFT JOIN goal ON player.id = goal.scorer_id
GROUP BY player.id
ORDER BY goals_scored DESC
LIMIT 5
```

player_name	goals_scored
Colm TheBest	4
Brian Moyles	2
Robbie Keane	1
Mohamed Salah	1
James McClean	0

List all teams and the number of points they have so far

```
SELECT t_name AS team_name, SUM((wins*3) + draws) AS points
FROM team
GROUP BY t_name
ORDER BY points DESC
```

team_name	points
Liverpool	114
Ballina Town FC	106
Ireland	70
Galway United	26