Lab 3 COE848 - Database Design

Hamza Iqbal

500973673

The following is the ER diagram of the database to be designed:

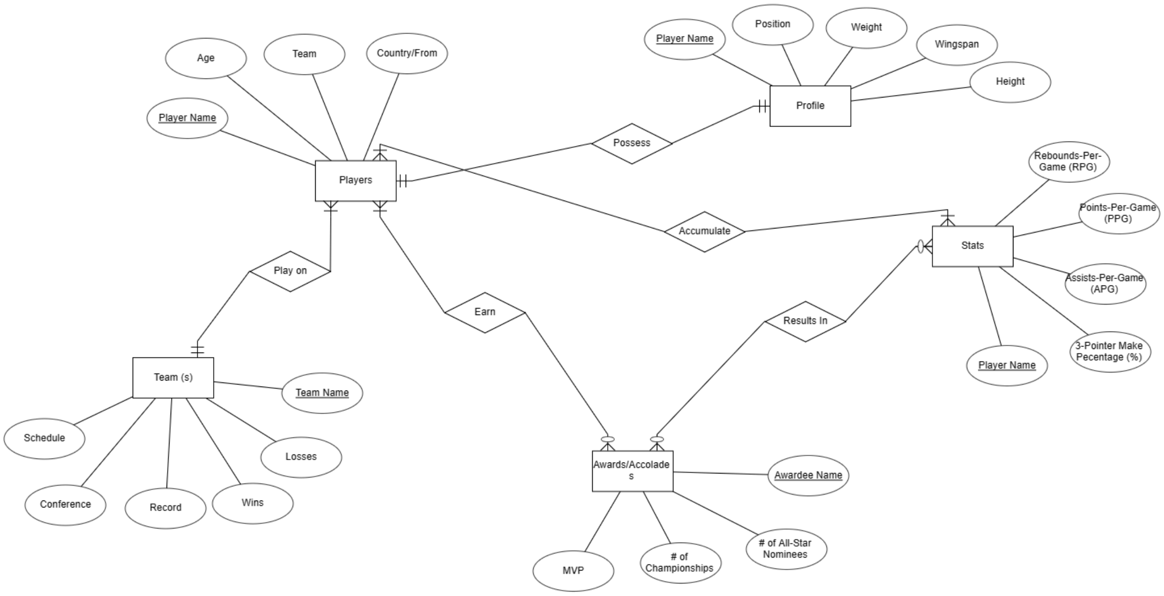
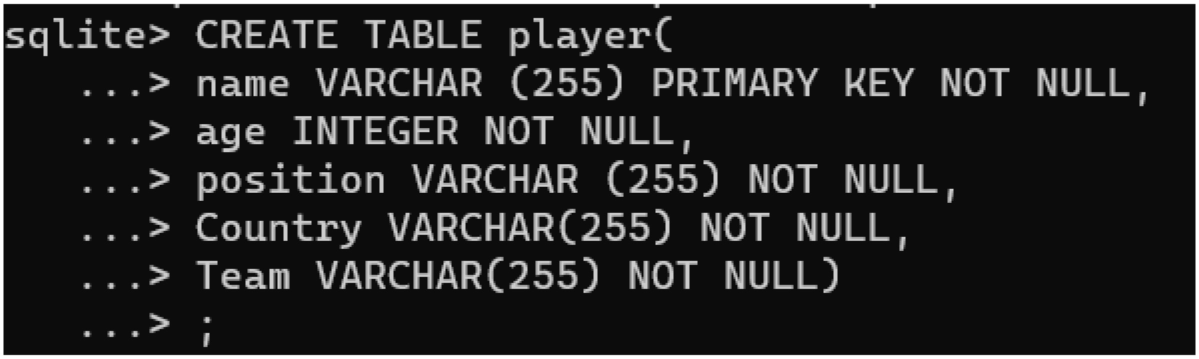


Figure 1: ER diagram of Basketball Database

Evidently, we can see from our ER model, we need 5 distinct tables corresponding to our design in order for the program to effectively work. Thus, we have created 5 tables with various relationships within our database to account for this.
We have a player table which will give basic information regarding the player. Likewise, the player has a page for awards, which the respective player has accumulated throughout his career thus far. We have a table relating to the team they play on, andvarious data in the current season of play of the respective team.

Additionally, a statistics table consisting of their various stats that they have accumulated. Finally, we also have a profile table, which gives the physical dimensions of a
Player.

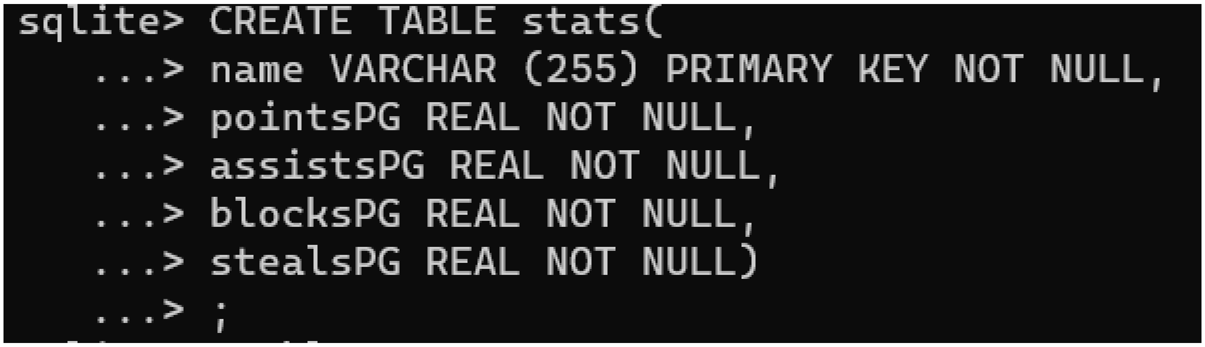
The following are the 5 entities translated into SQLite tables:



```
sqlite> CREATE TABLE player(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> age INTEGER NOT NULL,
   ...> position VARCHAR (255) NOT NULL,
   ...> Country VARCHAR(255) NOT NULL,
   ...> Team VARCHAR(255) NOT NULL)
   ...> ;
```

Figure 1: Player Table

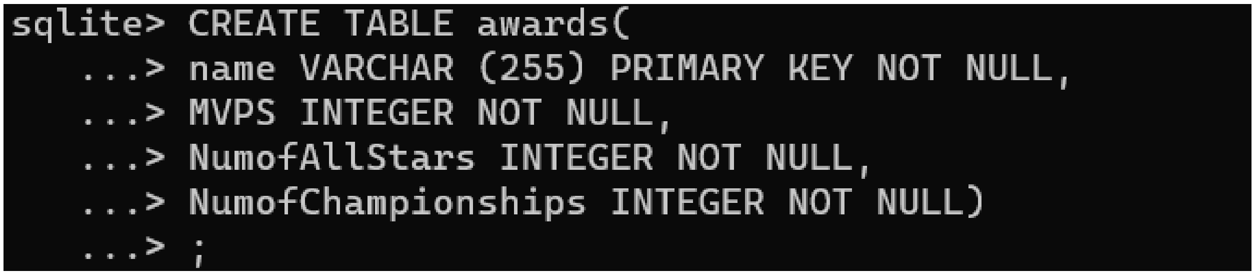
Within the player table, the name of the player will act as the primary key as a varchar. All other attributes are derived from the ER diagram player entity itself, and must be not null (contain a value). Attributes are described as varchar for values which may take characters (age,position, country, team) and int for those that only need numbers (age).



```
sqlite> CREATE TABLE stats(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> pointsPG REAL NOT NULL,
   ...> assistsPG REAL NOT NULL,
   ...> blocksPG REAL NOT NULL,
   ...> stealsPG REAL NOT NULL)
   ...> ;
```

Figure 2: Stats Table

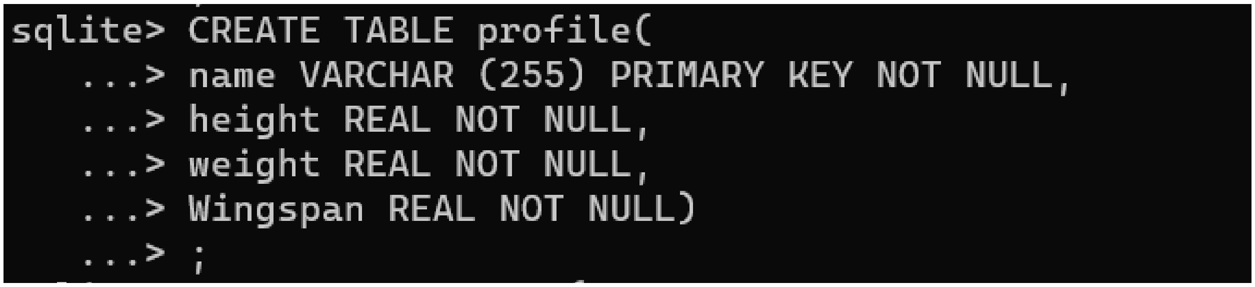
Within the stats table, the name of the player will act as the primary key once again as a varchar. All other attributes are derived from the ER diagram Table entity itself. Sinec stats are accumulated throughout the season regardless, all attributes must be not null (contain a value). Attributes are described as real numbers since they could be decimal values.



```
sqlite> CREATE TABLE awards(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> MVPS INTEGER NOT NULL,
   ...> NumofAllStars INTEGER NOT NULL,
   ...> NumofChampionships INTEGER NOT NULL)
   ...> ;
```

Figure 3: Awards Table

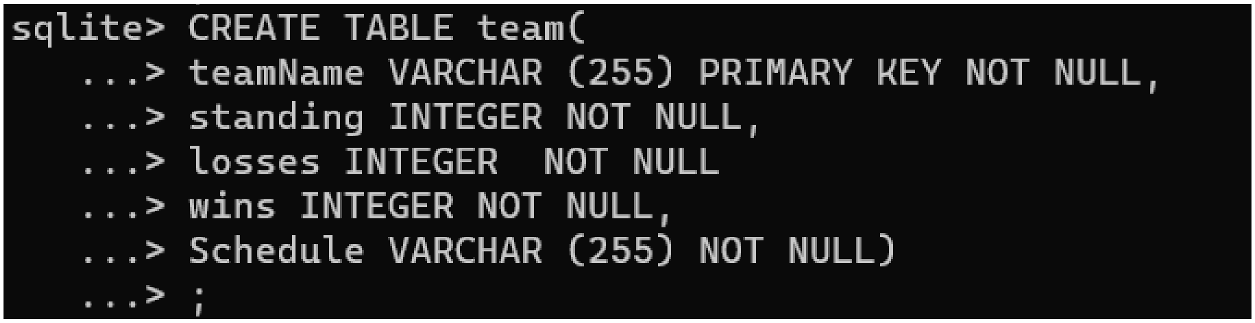
Within the Awards table, the name of the player will again act as the primary key as a varchar. All other attributes are derived from the ER diagram Table entity itself. Even though a player may have no MVPs/Allstars/Championships, values can be written as 0 & must be not null. Attributes are described as integers since they cannot be character or decimal values.



```
sqlite> CREATE TABLE profile(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> height REAL NOT NULL,
   ...> weight REAL NOT NULL,
   ...> Wingspan REAL NOT NULL)
   ...> ;
```

Figure 4: Profile Table

Within the profile table, the name of the player will act as the primary key once again as a varchar. All other attributes are derived from the ER diagram profile entity itself, and must be not null (contain a value), since every human has a physical profile. Attributes are described as REAL since decimal numbers may exist.

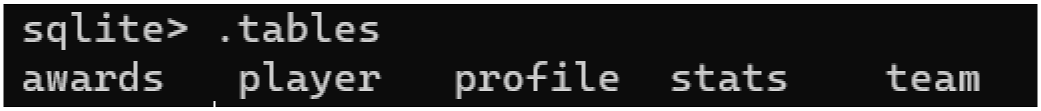


```
sqlite> CREATE TABLE team(
   ...> teamName VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> standing INTEGER NOT NULL,
   ...> losses INTEGER  NOT NULL
   ...> wins INTEGER NOT NULL,
   ...> Schedule VARCHAR (255) NOT NULL)
   ...> ;
```

Figure 5: Team Table

Within the team table, the name of the team will act as the primary key. All other attributes are derived from the ER diagram player entity itself, and must be not null (contain a value). Attributes are described as varchar for the ‘Schedule’ & ‘teamName’ attribute since that attribute will detail which teams will be played against and the name of ones own team, which would use varchar. While the rest of the attribute will take ints for those that only need numbers.

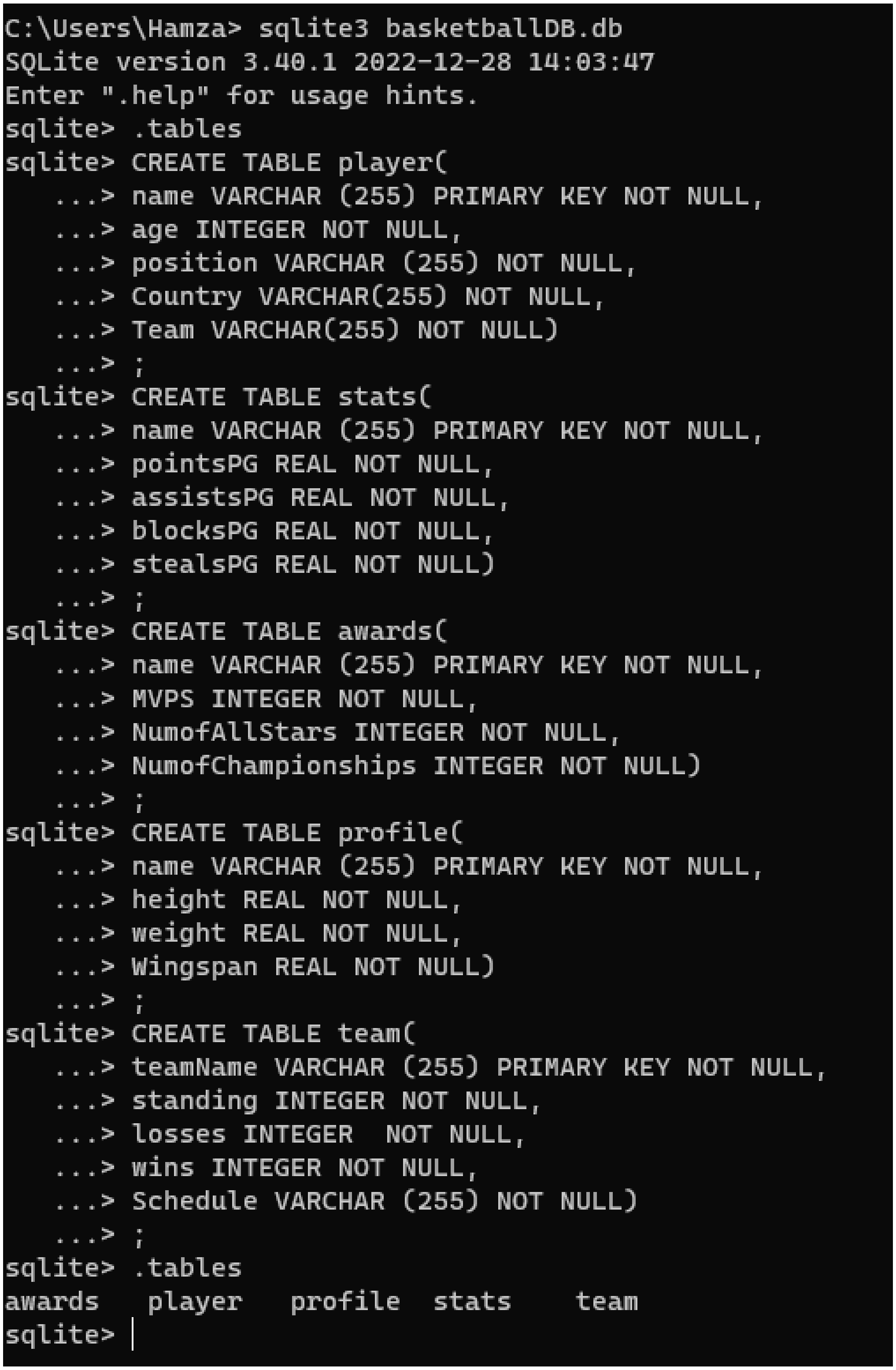
The following is the ‘.tables’ command showcasing all the tables implemented within sqlite:



```
sqlite> .tables
awards   player   profile  stats    team
```

Figure 6: List of all tables in SQLite

The following is the complete overview of the implementation within SQLite:



```
C:\Users\Hamza> sqlite3 basketballDB.db
SQLite version 3.40.1 2022-12-28 14:03:47
Enter ".help" for usage hints.
sqlite> .tables
sqlite> CREATE TABLE player(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> age INTEGER NOT NULL,
   ...> position VARCHAR (255) NOT NULL,
   ...> Country VARCHAR(255) NOT NULL,
   ...> Team VARCHAR(255) NOT NULL)
   ...> ;
sqlite> CREATE TABLE stats(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> pointsPG REAL NOT NULL,
   ...> assistsPG REAL NOT NULL,
   ...> blocksPG REAL NOT NULL,
   ...> stealsPG REAL NOT NULL)
   ...> ;
sqlite> CREATE TABLE awards(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> MVPS INTEGER NOT NULL,
   ...> NumofAllStars INTEGER NOT NULL,
   ...> NumofChampionships INTEGER NOT NULL)
   ...> ;
sqlite> CREATE TABLE profile(
   ...> name VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> height REAL NOT NULL,
   ...> weight REAL NOT NULL,
   ...> Wingspan REAL NOT NULL)
   ...> ;
sqlite> CREATE TABLE team(
   ...> teamName VARCHAR (255) PRIMARY KEY NOT NULL,
   ...> standing INTEGER NOT NULL,
   ...> losses INTEGER  NOT NULL,
   ...> wins INTEGER NOT NULL,
   ...> Schedule VARCHAR (255) NOT NULL)
   ...> ;
sqlite> .tables
awards   player   profile  stats    team
sqlite> 
```

Figure 7: Full overview of Implementation