# Description

The Player entity contains a **key attribute** of **Player_ID** with the attributes of First Name, Last Name, Shirt Number, Nationality, Position and Age. This will store the **Team_ID** as the **Foreign Key**.

The Team entity contains a **key attribute** of **Team_ID** with the attributes of Team_Abbreviation, Team_Name and Years_Since_Foundation. This will store the **Manager_ID** as the **foreign key.**

The Manager entity contains the **key attribute** of **Manager_ID** with the attributes of First_Name, and Last_Name.

A team must be managed by 1 manager and a manager can manage 1 team. A player can have 1 team and a Team must have at least 1 player.

# Normalization

The database is in BCNF.

**First Normal Form**

We surpass the first normal form as all our attributes are atomic.

**Second Normal Form**

Since none of our tables in the database contain a composite key. By default, our database is in 2nd Normal form. (Each non-key attribute is functionally dependant on the primary key of the table it is contained in.)

**Player**

Primary Key: Player_ID

**Manager**

Primary Key: Manager_ID

**Team**

Primary Key: Team_ID

### BCNF

There is no case that another non-prime attribute determines a non-prime attribute. Hence our database is in Third Normal Form. No transitive dependencies.

## DDL Statements

```
CREATE TABLE Manager (

Manager_ID INT NOT NULL,

First_Name VARCHAR(50),

Last_Name VARCHAR(50),

Date_Of_Birth DATE,

PRIMARY KEY (Manager_ID)

) ;


CREATE TABLE IF NOT EXISTS Team (

Team_ID INTEGER NOT NULL,

Team_Name VARCHAR(5),

Team_Abbreviation VARCHAR(5),

Years_Since_Foundation INTEGER,

Manager_ID INTEGER NOT NULL,

PRIMARY KEY (Team_ID),

FOREIGN KEY (Manager_ID) REFERENCES Manager(Manager_ID) );

UNIQUE (Manager_ID),
);

CREATE TABLE Player (
Player_ID INTEGER NOT NULL,
First_Name VARCHAR(50),
Last_Name VARCHAR(50),
Shirt_Number INT,
Nationality VARCHAR(50),
Position VARCHAR(50),
Age INTEGER,
Team_ID INTEGER NOT NULL,
PRIMARY KEY (Player_ID),
FOREIGN KEY (Team_ID) REFERENCES Team (Team_ID)
);
```

## Relational Algebra Formulations

## Deletions

Manager =Manager - $\sigma_{\text{Manager_ID = 2(}}$Manager)

Team =Team - $\sigma_{\text{Team_ID = 2(}}$Team)

## Group By

$\pi$ Position, Average (Age) ($\sigma_{\text{Team\_ID = 2}}$ (Player))

$\pi$Nationality, Count(Nationality) ($\sigma_{\text{Team\_ID = 1}}$(Team))

# DML Statements

**Deletions**

```
DELETE * FROM MANAGER
WHERE Manager_ID = 2;

DELETE * FROM Team
WHERE Team_ID = 2;
```

**Group By**

```
SELECT Nationality, Count(Nationality)
FROM Player
WHERE Team_ID = 1
GROUP BY Nationality;

SELECT Position, AVG(Age) AS Average_Age
FROM Player
WHERE Team_ID = 2
GROUP BY Position;
```