National Hockey League Database

By: Jake Litts

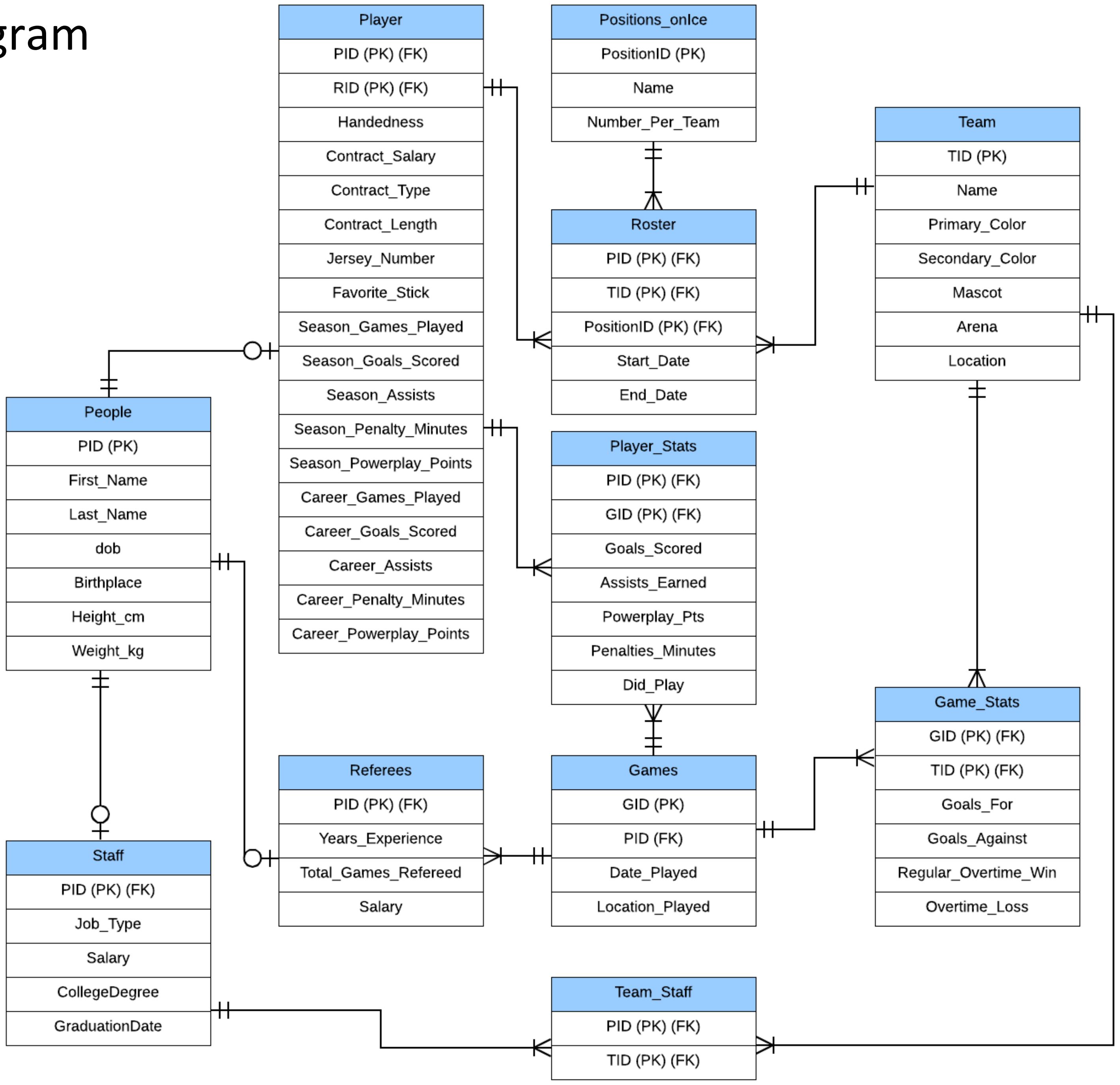
Table of Contents

Executive Summary

This database was created with the sole purpose of keeping track of every team, player, referee, and team staff member in the National Hockey League.

The following paper provides information regarding the database built, as well as the practical use of said database. Firstly, the E/R diagram is displayed in order to show the functional dependencies for the database. Next, create statements for the tables and sample data will be presented. Following these are the uses of the database, including creating queries, views, stored procedures, and triggers are designed and tested. Finally, there is a list of roles, improvements that need to be made to the database, and current problems it faces.

E/R Diagram


Player
PID (PK) (FK)
RID (PK) (FK)
Handedness
Contract_Salary
Contract_Type
Contract_Length
Jersey_Number
Favorite_Stick
Season_Games_Played
Season_Goals_Scored
Season_Assists
Season_Penalty_Minutes
Season_Powerplay_Points
Career_Games_Played
Career_Goals_Scored
Career_Assists
Career_Penalty_Minutes
Career_Powerplay_Points
Positions_onIce
PositionID (PK)
Name
Number_Per_Team
Team
TID (PK)
Name
Primary_Color
Secondary_Color
Mascot
Arena
Location
Roster
PID (PK) (FK)
TID (PK) (FK)
PositionID (PK) (FK)
Start_Date
End_Date
People
PID (PK)
First_Name
Last_Name
dob
Birthplace
Height_cm
Weight_kg
Player_Stats
PID (PK) (FK)
GID (PK) (FK)
Goals_Scored
Assists_Earned
Powerplay_Pts
Penalties_Minutes
Did_Play
Game_Stats
GID (PK) (FK)
TID (PK) (FK)
Goals_For
Goals_Against
Regular_Overtime_Win
Overtime_Loss
Referees
PID (PK) (FK)
Years_Experience
Total_Games_Refereed
Salary
Games
GID (PK)
PID (FK)
Date_Played
Location_Played
Staff
PID (PK) (FK)
Job_Type
Salary
CollegeDegree
GraduationDate
Team_Staff
PID (PK) (FK)
TID (PK) (FK)

Tables:

People Table – The People table keeps track of every person within the database with basic information. This information is shared between every player, referee, and team staff member.

```
CREATE TABLE People(
  PID int not null,
  First_Name text not null,
  Last_Name text not null,
  dob DATE not null,
  Height_cm decimal(8,3) not null,
  weight_kg decimal(7,3) not null,
  Birthplace text not null,
  primary key(PID)
);
```

Functional Dependencies:

PID -> First_Name, Last_Name, dob, Height_cm,

Weight_kg, Hair_color, Eye_color, Birthplace

Players Table - The Players table keeps track of any person within the database who is also a player. It keeps track of the handedness, jersey number, favorite stick, and the contract of each player. In addition, the table keeps stats of that player both for the season and for their career.

```
CREATE TABLE Players(
  PID int not null references People(PID),
  Handedness text not null,
  Contract_Salary int,
  Contract_Type text,
  Contract_Length text,
  Jersey_Number int not null,
  Favorite_Stick text not null,
  Season_Games_Played int,
  Season_Goals_Scored int,
  Season_Assists int,
  Season_Penalty_Minutes int,
  Season_Powerplay_Points int,
  Career_Games_Played int,
  Career_Goals_Scored int,
  Career_Assists int,
  Career_Penalty_Minutes int,
  Career_Powerplay_Points int,
  primary key(PID)
);
```

Functional Dependencies:

PID -> Handedness, Contract_Salary, Contract_Type, Contract_Length, Jersey_Number, Favorite_Stick, Season_Games_Played, Season_Goals_Scored, Season_Assists, Season_Penalty_Minutes, Season_Powerplay_Points, Career_Games_Played, Career_Goals_Scored, Career_Assists, Career_Penalty_Minutes, Career_Powerplay_Points,

Referee Table – The Referees table keeps track of any person within the database who is also a referee, as well as count how many games each referee has participated in, their salary, and their years of experience.

```
CREATE TABLE Referees(
 PID int not null references People(PID),
  Total_Games_Refereed int not null,
  Years_Experience int not null,
  Salary int not null,
  primary key(PID)
);
```

Functional Dependencies:

PID -> Total_Games_Refereed, Years_Experience, Salary

Positions_onIce Table – The Positions_onIce table keeps track of all positions a player can be in the game of Hockey, giving each position a name and stating the number of players on a team that can be that position at one time.

```
CREATE TABLE Positions_onIce(
  PositionID int not null,
  Name text not null,
  Number_Per_Team int not null,
  primary key(PositionID)
);
```

Functional Dependencies:

PositionID -> Name, Number_Per_Team

Teams Table – The Teams table keeps track of every team in the league, including their team name, team colors, team mascot, arena, and their location.

```
CREATE TABLE Teams(
  TID int not null,
  Name text not null,
  Primary_Color text not null,
  Secondary_Color text not null,
  Mascot text not null,
  Arena text not null,
  Location text not null,
  primary key(TID)
);
```

Functional Dependencies:

TID -> Name, Primary_Color, Secondary_Color, Mascot, Arena, Location

Roster Table – The Roster table intersects players, teams, and positions in order to keep track of every player in the league, what team they are on, and what position they have when they step on the ice.

```
CREATE TABLE Roster(
  PID int not null references People(PID),
  TID int not null references Teams(TID),
  PositionID int not null references Positions_onIce(PositionID),
  Start_Date DATE not null,
  End_Date DATE,
  primary key(PID, TID, PositionID)
);
```

Functional Dependencies:

PID, TID, PositionID -> Start_Date, End_Date

Games Table – The Games table contains data for each game played, including the date, people in the game, and the location of the game.

```
CREATE TABLE Games(
  GID int not null,
  PID int not null references People(PID),
  Date_Played DATE not null,
  Location_Played text not null,
  primary key(GID)
);
```

Functional Dependencies:

GID -> PID, Date_Played, Location_Played

Player_Stats Table – The Player_Stats table contains data about the players on the teams that played in each match, along with their respective goals, assists, powerplay points, and penalty minutes earned in the game and whether or not they actually played in the game for their team.

```
CREATE TABLE Player_Stats(
  PID int not null references
People(PID),
  GID int not null references
Games(GID),
  Goals_Scored int not null,
  Assists_Earned int not null,
  Powerplay_Pts int not null,
  Penalty_Minutes int not null,
  Did_Play boolean not null,
  primary key(PID, GID)
);
```

Functional Dependencies:

PID, GID-> Goals_Scored, Assists_Earned, Powerplay_Pts, Penalty_Minutes, Did_Play

Game_Stats Table – The Game_Stats table contains data about which teams played in which game, along with their respective goals for, and goals against for the match and whether or not the game went to overtime.

```
CREATE TABLE Game_Stats(
  GID int not null references Games(GID),
  TID int not null references Teams(TID),
  Goals_For int not null,
  Goals_Against int not null,
  Regulation_Overtime_Win boolean not null,
  Overtime_Loss boolean not null,
  primary key(GID, TID)
);
```

Functional Dependencies:

GID, TID -> Goals_For, Goals_Against, Regulation_Overtime_Win, Overtime_Loss

Staff Table – The Staff table keeps track of any person within the database who is a part of the staff of a team in the National Hockey League, as well as their salary, type of job they work, college degree, and when they graduated.

```
CREATE TABLE Staff(
  PID int not null references People(PID),
  Job_Type text not null,
  Salary int not null,
  CollegeDegree text,
  GradutationDate DATE,
  primary key(PID)
);
```

Functional Dependencies:

PID -> Job_Type, Salary, CollegeDegree, GraduationDate

Team_Staff Table – The Team_Staff table keeps track of which team each staff member works for.

```
CREATE TABLE Team_Staff(
  PID int not null references People(PID),
  TID int not null references Teams(TID),
  Primary key(PID, TID)
);
```

Functional Dependencies:

PID, TID ->

Views:

Current_Rosters - Lists the name, position, and team for every active player on every team.

```
CREATE VIEW Current_Rosters as
 SELECT First_Name as First_Name,Last_Name as Last_Name,Positions_onIce.Name as Position, Teams.Name as Team,
 FROM Players
  INNER JOIN People ON Players.PID=People.PID
  INNER JOIN Rosters ON Players.PID=Roster.PID
  INNER JOIN Positions_onIce ON Roster.PositionID=Positions_onIce.PositionID
  INNER JOIN Teams ON Roster.TID=Teams.TID
 WHERE Roster.End_Date IS NULL
 ORDER BY Teams ASC, People ASC;
```

Maurice_Richard_Trophy - Lists the player with the highest season goals scored, along with their team and position.

```
CREATE VIEW Maurice_Richard_Trophy as
 SELECT First_Name as First_Name,Last_Name as Last_Name, Season_Goals_Scored as Total_Goals,
        Positions_onIce.Name as Position, Teams.Name as Team
 FROM Players
  INNER JOIN People ON Players.PID=People.PID
  INNER JOIN Roster ON Players.PID=Roster.PID
  INNER JOIN Positions_onIce ON Roster.PositionID=Positions_onIce.PositionID
  INNER JOIN Teams ON Roster.TID=Teams.TID
WHERE Players.Season_Goals_Scored = (SELECT Players.Season_Goals_Scored
                                     FROM Players
                                     WHERE Season_Goals_Scored IS NOT NULL
                                     ORDER BY Season_Goals_Scored DESC
                                     LIMIT 1)
ORDER BY Season_Goals_Scored DESC, teams ASC;
```

Stored Procedures:

match_to_ref - This function returns a trigger and increments the games refereed row in referees depending on which ref was working which match.

```
CREATE OR REPLACE FUNCTION match_to_ref() RETURNS trigger AS $m2r$
  BEGIN
    UPDATE Referees SET Total_Games_Refereed=Total_Games_Refereed+1 WHERE PID=NEW.PID;
  RETURN NEW;
  END;
$m2r$ LANGUAGE plpgsql;
```

update_player_stats - This function returns a trigger and updates / adds to each player's stats(in the players table) according to the stats they received in the last game.

```
CREATE OR REPLACE FUNCTION update_player_stats() RETURNS trigger AS $update_player_stats$
  BEGIN
    IF (select Career_Goals_Scored from Players where PID=NEW.PID) IS NULL THEN
      IF NEW.Did_Play THEN
        UPDATE Players
        SET Career_Goals_Scored=0, Career_Penalty_Minutes=0,
      Season_Goals_Scored=0, Season_Assists=0, Season_Penalty_Minutes=0, Season_Powerplay_Points=0,

      Career_Games_Played=0, Career_Assists=0, Career_Powerplay_Points=0,

        WHERE Players.PID=NEW.PID;

      END IF;

    END IF;

    UPDATE Players

    SET Season_Games_Played=Season_Games_Played+1, Season_Goals_Scored=Season_Goals_Scored+NEW.Goals_Scored,
Season_Assists=Season_Assists+NEW.Assists_Earned, Season_Penalty_Minutes=Season_Penalty_Minutes+NEW.Penalty_Minutes,
Season_Powerplay_Points=Season_Powerplay_Points+NEW.Powerplay.Points, Career_Games_Played=Career_Games_Played+1,
Career_Goals_Scored=Career_Goals_Scored+NEW.Goals_Scored, Career_Assists=Career_Assists+NEW.Assists_Earned,
Career_Penalty_Minutes=Career_Penalty_Minutes+NEW.Penalty_Minutes,
Career_Powerplay_Points=Career_Powerplay_Points+NEW.Powerplay_Points

    WHERE Players.PID=NEW.PID;

    RETURN NEW;

  END;

$update_player_stats$ LANGUAGE plpgsql;
```

Triggers:

update_player_stats - This trigger executes the update_player_stats function (See: Stored Procedures) before data has been inserted into Player_Stats.

```
CREATE TRIGGER update_player_stats BEFORE INSERT ON Player_Stats
  FOR EACH ROW EXECUTE PROCEDURE update_player_stats();
```

m2r - This trigger executes the match_to_ref function (See: Stored Procedures) before data has been inserted into matches.

```
CREATE TRIGGER m2r BEFORE INSERT ON Games
  FOR EACH ROW EXECUTE PROCEDURE match_to_ref();
```

Roles:

Administrator Role - The database administrator, full access to entire database.

```
create role admin;

grant all on all tables in schema public to admin;
```

Referee Role - It is the referee's job to input new players, new referees, new games, new staff, and new game data in general.

```
create role referee;

revoke all on all tables in schema public from referee;

grant select on all tables in schema public to referee;

grant insert on People, Players, Referees, Games, Player_stats, Staff, Roster to referee;

grant update on People, Players, Referees, Games, Player_stats, Roster, Teams to referee;
```

JoeSmo Role - Anyone may query the database to learn about the players in the NHL.

```
create role JoeSmo;

grant select on all tables in schema public to JoeSmo;
```

Implementation Notes:

When coding the NHL database, there were a lot of game specific stats that needed to be accounted for, including powerplay points, penalty minutes, goals, and assists. This data needed to be accurately kept in order decide who wins the awards at the end of each season. In addition, the career stats for a player must be kept in order to figure out after a player retires whether or not they were good enough to make it into the Hockey Hall of Fame.

Known Problems and Future Enhancements:

Unfortunately, there quite a few problems with my database that would be corrected given more time. For starters, the database is able to determine the scores of a given game, however it is unable to take this data, determine which score is the winner and loser, and give the appropriate team a win or loss. In addition, the coding of overtime losses, ROW (which is given if you win in regulation or overtime, but not in a shootout), and the calculation of team points as a result of winning, losing, or losing in overtime were not completed.

If given more time, there are a few other functions that I wanted to include in this database, as well as more data. I wanted to include functions that would account for the other trophies given out at the NHL awards at the end of every season, though I only had time to code the Maurice Richard trophy. Another function I wanted to include would be one that would calculate how much salary each team had left. This function would have taken the salary of each player, added it up, and compared it to the current salary cap. This would allow general managers to figure out how much cap they have left, so that they know which players they have the cap to obtain for their team. A feature that would go along with this idea would be to implement a list of free agents. Finally, I wanted to include a system that would implement the affiliate team for each NHL team in the AHL, allowing teams to easily transfer players from the AHL to the NHL with ease. This would be an imperative feature had I also had the time to implement an injured reserve.