# Marist Athletics

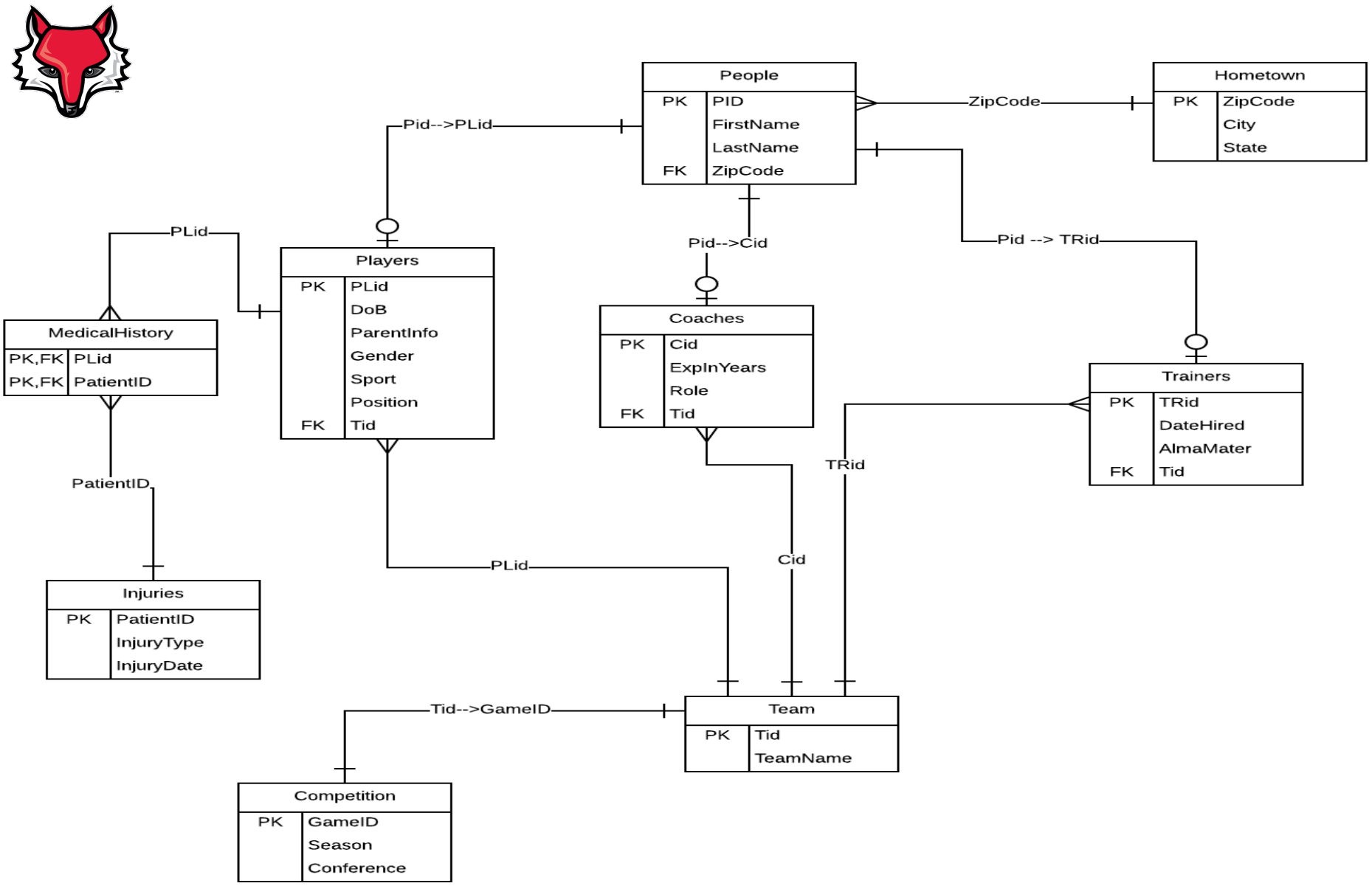
Database Design by:
Mark Rajovic

# Table of Contents

# Executive Summary

This document outlines the design to hold all data for the Marist College Athletic Department, regarding all parties involved with a Red Fox sponsored team. The Marist athletic department sponsors 23 sports in which about 600 athletes compete for. This database shows the basic design in order to keep all information regarding athletes, coaches and trainers useful and safe. There are tables listing all people in the athletic department, further breaking them down into their specific role in the athletic department. It also allows for Marist to keep track of which intercollegiate conference a team competes in. In sports, it is an unfortunate reality that players will sustain injuries, therefore the database allows for Trainers to keep track of players injuries in order to fast track the athletes back to competitive shape.

Due to the design of the database, it allows for complex queries to find player rosters, or coaches that work on the same team together. These can be stored as views to simplify finding information that is relevant to a specific Red Fox athletic team of the Athletic Department as a whole. The ultimate goal of this database design is to create a fully normalized database in third normal form that can help Marist administrators facilitate athletic department operations.




# Entity Relationship Diagram

# People Table

Lists all People involved in the athletic department as well, as their hometown zip code.

```
CREATE TABLE People (
  Pid       char(4) not null,
  firstName      text not null,
  lastName      text  not null,
  ZipCode        char(5)  not null references Hometown(ZipCode),
 primary key(Pid)
 );
```

## Functional Dependencies:

Pid → firstName, lastName, ZipCode

Sample Data →

| pid character | firstname text | lastname text | zipcode character |
|---|---|---|---|
| 1 | Mark | Rajovic | 10536 |
| 2 | Blaise | Spinelli | 11798 |
| 3 | Zack | Reed | 10536 |
| 4 | Drew | Sullivan | 06830 |
| 5 | Mike | White | 12601 |
| 6 | Christina | Lampasi | 11456 |
| 7 | Mackenzie | Obrien | 45632 |
| 8 | Beau | Hornberger | 11456 |
| 9 | Steven | Rizzo | 12601 |
| 10 | Amanda | Greco | 12601 |
| 11 | Matt | Viggiano | 12601 |
| 12 | Brandon | Curtis | 45632 |
| 13 | Hope | Quinonez | 10536 |
| 14 | James | Parady | 12601 |
| 15 | Keith | Detelj | 45632 |
| 16 | Krystian | Witkowski | 11456 |
| 17 | Gene | Smith | 06830 |
| 18 | Mary | DiChiara | 10560 |
| 19 | Christian | Keenan | 12601 |
| 20 | Devon | Cummings | 11456 |
| 21 | Sarah | Cannon | 45632 |
| 22 | Justin | Giuliano | 12601 |
| 23 | Ryan | Spaulding | 06830 |
| 24 | Libby | Adams | 90098 |
| 25 | Brianna | Galleazzi | 12601 |
| 26 | Jake | Connors | 12601 |
| 27 | Laura | Brugnatelli | 90098 |
| 28 | Harry | Heffernan | 10560 |
| 29 | Lisa | Smith | 90098 |
| 30 | Meredith | Wertz | 45632 |
| 31 | Shannon | Gordon | 10536 |
| 32 | Georgia | Goldman | 11456 |
| 33 | Dierdre | Newton | 12601 |
| 34 | Jack | Ryan | 11456 |
| 35 | Mitch | Standera | 11456 |
| 36 | Dave | Meyer | 10560 |
| 37 | Kyle | Quinn | 45632 |
| 38 | Jack | Monnes | 90098 |
| 39 | Cam | Harr | 10536 |
| 40 | Taylor | Swift | 10560 |
| 41 | Lauren | Amundson | 12601 |
| 42 | Mark | Jackson | 11456 |
| 43 | Alex | Rajovic | 10536 |
| 44 | Daniel | Chung | 10560 |
| 45 | Peter | Yeung | 90098 |
| 46 | Brock | Lesnar | 45632 |
| 47 | John | Cena | 11798 |
| 48 | Nicole | Ficano | 11798 |

# Hometown Table

Lists all Zip Codes found in data and connects them to the cities and states they belong to.

```
CREATE TABLE Hometown (
  ZipCode      char(5) not null,
  City     text not null,
  State     text  not null,
 primary key(ZipCode)
);
```

**Functional Dependencies:**

ZipCode → City, State

Sample Data →

| zipcode character | city text | state text |
|---|---|---|
| 10536 | Katonah | NY |
| 11798 | Mobile | AL |
| 06830 | Greenwich | CT |
| 12601 | Poughkee... | NY |
| 11456 | Pittsburgh | PA |
| 45632 | Chicago | IL |
| 10560 | Croton | NY |
| 90098 | Los Angel... | CA |

# Players Table

Lists all players and information relevant to the Marist Athletic Department.

```
CREATE TABLE Players (
  PLid            char(4) not null references People(Pid),
  DoB             date not null,
  ParentInfo        char(15),
  Gender           text,
  Sport          text not null,
  Position     text not null,
  Tid       char(4) not null references Team(Tid),
  CONSTRAINT check_gender CHECK (gender = 'M' OR gender = 'F'),
 primary key(PLid)
);
```

**Functional Dependencies:**

PLid → DoB, ParentInfo, Gender, Sport, Position, Tid

Sample Data →

| plid character | dob date | parentinfo character | gender text | sport text | position text | tid character |
|---|---|---|---|---|---|---|
| 1 | 1995-11-10 | 91470327... | M | Soccer | Goalie | 1 |
| 2 | 1997-12-04 | 91470327... | M | Football | Safety | 2 |
| 3 | 1995-04-06 | 91445327... | M | Soccer | Defense | 1 |
| 4 | 1996-11-10 | 20470327... | M | Soccer | Midfield | 1 |
| 5 | 1998-09-13 | 91470324... | M | Football | Quarterba... | 2 |
| 6 | 1994-11-15 | 91470327... | F | Soccer | Defense | 4 |
| 7 | 1996-08-10 | 91470327... | F | Swim | Backstroke | 5 |
| 8 | 1995-05-05 | 91470323... | M | Soccer | Defense | 1 |
| 9 | 1996-02-02 | 91470328... | M | Lacrosse | Attack | 3 |
| 12 | 1996-05-23 | 91985430... | M | Soccer | Forward | 1 |
| 13 | 1997-12-31 | 10293027... | F | Soccer | Forward | 4 |
| 18 | 1997-04-09 | 91427525... | F | Swim | Butterfly | 5 |
| 19 | 1995-11-10 | 45489894... | M | Football | Wide Rece... | 2 |
| 20 | 1998-03-19 | 91470527... | F | Soccer | Defense | 4 |
| 21 | 1995-03-03 | 93470327... | F | Soccer | Defense | 4 |
| 23 | 1995-03-09 | 91980327... | M | Football | Kicker | 2 |
| 24 | 1995-06-05 | 98470327... | F | Soccer | Goalie | 4 |
| 28 | 1996-09-30 | 03470327... | M | Lacrosse | Goalie | 3 |
| 29 | 1996-12-20 | 91470354... | F | Swim | Free | 5 |
| 30 | 1995-07-22 | 53470327... | F | Swim | Butterfly | 5 |
| 31 | 1998-07-11 | 91370327... | F | Swim | Backstroke | 5 |
| 32 | 1995-07-23 | 25470327... | F | Swim | Free | 5 |
| 33 | 1998-08-14 | 91470300... | F | Swim | Free | 5 |
| 34 | 1997-04-12 | 91470327... | M | Lacrosse | Attack | 3 |
| 35 | 1995-12-08 | 91470327... | M | Lacrosse | Defense | 3 |
| 36 | 1997-02-04 | 91470427... | M | Lacrosse | Goalie | 3 |
| 38 | 1998-08-09 | 91340327... | M | Soccer | Defense | 1 |
| 39 | 1998-06-10 | 91270327... | M | Soccer | Forward | 1 |
| 40 | 1999-11-13 | 00470327... | F | Swim | Free | 4 |
| 41 | 1996-12-01 | 80470327... | F | Swim | Backstroke | 4 |
| 42 | 1996-03-10 | 91473727... | M | Football | Corner Ba... | 2 |
| 43 | 1994-11-09 | 91465327... | M | Football | Running B... | 2 |
| 45 | 1998-11-10 | 91470355... | M | Lacrosse | Midfield | 3 |
| 46 | 1996-09-09 | 12908347... | M | Football | Offensive ... | 2 |
| 47 | 1995-12-30 | 91567827... | M | Football | Linebacker | 2 |
| 50 | 1995-05-10 | 91470398... | M | Football | Quarterba... | 2 |

# Coaches Table

Lists all coaches in the athletic department and their specific coaching role and amount of experience.

```
CREATE TABLE Coaches (
  Cid           char(4) not null references People(Pid),
  ExpInYears       int not null,
  Role          text not null,
  Tid       char(4) not null references Team(Tid),
 primary key(Cid)
);
```

**Functional Dependencies:**

Cid → ExpInYears, Role, Tid

Sample Data →

| cid character | expinyears integer | role text | tid character |
|---|---|---|---|
| 11 | 10 | Head Coach | 1 |
| 14 | 27 | Head Coach | 2 |
| 15 | 2 | Assistant | 1 |
| 16 | 4 | Director o... | 1 |
| 17 | 1 | Head Coach | 4 |
| 26 | 10 | Head Coach | 3 |
| 27 | 3 | Head Coach | 5 |
| 37 | 5 | Assistant | 3 |
| 40 | 1 | Director o... | 2 |
| 49 | 2 | Offensive ... | 2 |

# Trainers Table

Lists all of Marist's athletic trainers, their college alma mater, and the date they were hired.

```
CREATE TABLE Trainers (
  TRid            char(4) not null references People(Pid),
  DateHired        date not null,
  AlmaMater          text not null,
  Tid       char(4) not null references Team(Tid),
 primary key(TRid)
);
```

**Functional Dependencies:**

Trid → DateHired, AlmaMater, Tid

Sample Data →

| trid character | datehired date | almamater text | tid character |
|---|---|---|---|
| 10 | 2009-06-06 | Sacred He... | 1 |
| 22 | 2011-11-05 | UMass-A... | 2 |
| 25 | 2012-05-30 | Rhode Isl... | 3 |
| 44 | 1996-05-05 | Union Coll... | 4 |
| 48 | 2013-07-23 | Syracuse ... | 5 |

# Team Table

Lists all intercollegiate athletic teams fielded by the Marist athletic Department.

```
CREATE TABLE Team (
  Tid       char(4) not null,
  TeamName      text not null,
 primary key(Tid)
);
```

**Functional Dependencies:**

Tid → TeamName

Sample Data →

| tid character | teamname text |
|---|---|
| 1 | MensSoccer |
| 2 | Football |
| 3 | MensLacr... |
| 4 | WomensS... |
| 5 | WomensS... |

# Injuries Table

Lists all injured athletes by their PatientID as well as when they were injured and a brief description of the injury.

```
CREATE TABLE  Injuries(
  PatientID            char(4) not null,
  InjuryType          text,
  InjuryDate          date,
 primary key(PatientID)
);
```

**Functional Dependencies:**

PatientID → InjuryType, InjuryDate

Sample Data →

| patientid character | injurytype text | injurydate date |
|---|---|---|
| 1 | Back | 2015-11-11 |
| 2 | Left Knee | 2016-10-23 |
| 3 | Right Ankle | 2016-09-01 |
| 4 | Hip | 2016-06-04 |
| 5 | Torn ACL | 2015-08-08 |
| 6 | Broken Toe | 2017-01-01 |
| 7 | Separated... | 2016-02-11 |
| 8 | Concussion | 2016-07-04 |
| 9 | Concussion | 2016-11-30 |
| 10 | Hip | 2016-03-25 |

# MedicalHistory Table

Connects all players through PLid to their assigned PatientID, if a when a player contracts an injury.

```
CREATE TABLE MedicalHistory(
    PLid      char(4) not null references Players(PLid),
    PatientID    char(4) not null references Injuries(PatientID),
  primary key (PatientID, PLid)
);
```

**Functional Dependencies:**

(Plid,PatientID) → N/A

Sample Data →

| plid character | patientid character |
|---|---|
| 2 | 1 |
| 7 | 2 |
| 12 | 3 |
| 18 | 4 |
| 21 | 5 |
| 28 | 6 |
| 36 | 7 |
| 38 | 8 |
| 47 | 9 |
| 50 | 10 |

# Competition Table

Lists the calendar season and intercollegiate conference each team competes in.

```
CREATE TABLE Competition(
    GameID  char(4) not null references Team(Tid),
    Season  text not null,
    Conference  text not null,
  primary key (GameID)
);
```

**Functional Dependencies:**

GameID → Season, Conference

Sample Data →

| gameid character | season text | conference text |
|---|---|---|
| 1 | Fall | MAAC |
| 2 | Fall | PFL |
| 3 | Spring | MAAC |
| 4 | Fall | MAAC |
| 5 | Winter | MAAC |

# Views

**MensLaxRoster** lists the first and last names of each player on the team, as well as the position they play.

```
CREATE VIEW MensLaxRoster as
select people.firstName, people.lastName, players.position
from people
inner join players on players.plid = people.pid
inner join team on team.tid = players.tid
where team.tid ='3'
;
```

View Query

```
select *
from MensLaxRoster
order by position ASC
```

Lacrosse Roster Data Output →

| firstname text | lastname text | position text |
|---|---|---|
| Steven | Rizzo | Attack |
| Jack | Ryan | Attack |
| Mitch | Standera | Defense |
| Harry | Heffernan | Goalie |
| Dave | Meyer | Goalie |
| Peter | Yeung | Midfield |

# Views

**FootballCoaches- lists names, role and years of coaching experiences for all, Coaches working with the Football team.**

```
CREATE VIEW FootballCoaches as
select people.firstName,
       people.lastName,
       coaches.ExpInYears,
       coaches.role
from people
inner join coaches on coaches.cid = people.pid
inner join team on team.tid = coaches.tid
where team.tid ='2'
;
```

View Query:

```
select *
from FootballCoaches
order by ExpInYears DESC
```

| firstname text | lastname text | expinyears integer | role text |
|---|---|---|---|
| James | Parady | 27 | Head Coach |
| Todd | Gurley | 2 | Offensive ... |
| Taylor | Swift | 1 | Director o... |

← Football Coaches Output Data

# Reports
## *Interesting Queries*

Query to display the name and sport of all male athletes that compete in the MAAC conference.

```
-- Reports Male MAAC athletes --
select people.firstName,
       people.lastName,
       players.sport
from people
inner join players on players.plid = people.pid
inner join team on team.tid = players.tid
inner join competition on competition.gameid = team.tid
where competition.conference = 'MAAC' AND
players.gender ='M'
order by players.sport
;
```

Male MAAC Athletes
Sample Data →

| firstname text | lastname text | sport text |
|---|---|---|
| Peter | Yeung | Lacrosse |
| Steven | Rizzo | Lacrosse |
| Harry | Heffernan | Lacrosse |
| Jack | Ryan | Lacrosse |
| Mitch | Standera | Lacrosse |
| Dave | Meyer | Lacrosse |
| Mark | Rajovic | Soccer |
| Jack | Monnes | Soccer |
| Zack | Reed | Soccer |
| Drew | Sullivan | Soccer |
| Beau | Hornberger | Soccer |
| Cam | Harr | Soccer |
| Brandon | Curtis | Soccer |

# Reports
## *Interesting Queries*

Query to display the number of coaches working with a specified team.

```
select team.teamname,
       count (coaches.cid) as NumCoaches
from coaches
inner join Team on team.tid = coaches.tid
group by team.TeamName
order by count (coaches.cid) DESC;
```

| teamname text | numcoac... bigint |
|---|---|
| MensSoccer | 3 |
| Football | 3 |
| MensLacr... | 2 |
| WomensS... | 1 |
| WomensS... | 1 |

← Number Of Coaches On Each Team Sample Data

# Reports
## *Interesting Queries*

Query to display the name of the injured athlete,
as well as a description of their injury for all athletes injured after June, 1st 2016.

```
select people.firstName,
       people.lastName,
       injuries.InjuryType,
       injuries.InjuryDate
from people
inner join players on players.plid = people.pid
inner join medicalhistory on medicalhistory.plid = players.plid
full outer join injuries on injuries.PatientID = medicalhistory.PatientID
where injurydate >= '06/01/2016'
```

Sample Data Returned →

| firstname text | lastname text | injurytype text | injurydate date |
|---|---|---|---|
| Mackenzie | Obrien | Left Knee | 2016-10-23 |
| Brandon | Curtis | Right Ankle | 2016-09-01 |
| Mary | DiChiara | Hip | 2016-06-04 |
| Harry | Heffernan | Broken Toe | 2017-01-01 |
| Jack | Monnes | Concussion | 2016-07-04 |
| John | Cena | Concussion | 2016-11-30 |

# Stored Procedure

This procedure tells the exact age of an athlete down to the day when referenced by their Plid.

```
CREATE  OR  REPLACE FUNCTION AthleteAge (Playerid char (4))
RETURNS INTERVAL as
$$
 DECLARE

  birthday date :=  (select players.DoB
                     from players
                     where players.PLid=Playerid
                    );

BEGIN
    RETURN  age(birthday);
END;
$$
LANGUAGE    plpgsql;
```

Age of Athlete Sample Data →

```
select AthleteAge ('1')
```

| athleteage<br>interval |
| --- |
| 21 years 5 mons 21 days |

# Stored Procedure

This stored procedure returns a trigger to update the competition table, that is affected when a new team is added to the department.

```
CREATE  OR  REPLACE FUNCTION NewTeam ()
RETURNS TRIGGER as
$$

BEGIN
    INSERT INTO Competition (GameID, Season, Conference)
    values
    (NEW.tid, 'Fall', 'MAAC');
    RETURN NEW;

END;
$$
LANGUAGE plpgsql;
```

Sample Data is provided on the corresponding trigger slide

# Trigger

Trigger updates the competition table with corresponding TeamID
as well as information regarding the season and conference in which the team competes.

```
drop TRIGGER compete
on team

CREATE TRIGGER Compete
AFTER INSERT ON Team
FOR EACH ROW
EXECUTE PROCEDURE NewTeam();

INSERT INTO Team (Tid, TeamName)
    values
    ('7','MensVolleyball');

select * from competition
```

# Trigger Sample Data Example

Competition Table
Before Trigger

| gameid character | season text | conference text |
|---|---|---|
| 1 | Fall | MAAC |
| 2 | Fall | PFL |
| 3 | Spring | MAAC |
| 4 | Fall | MAAC |
| 5 | Winter | MAAC |

Competition Table
After Trigger

| gameid character | season text | conference text |
|---|---|---|
| 1 | Fall | MAAC |
| 2 | Fall | PFL |
| 3 | Spring | MAAC |
| 4 | Fall | MAAC |
| 5 | Winter | MAAC |
| 7 | Fall | MAAC |

# Security

The purpose of this section is to identify and define user roles associated with the system, then grant or revoke privileges regarding the data to the involved groups.

**Administrator →**

```
Drop Role if exists administrator;
create role administrator;
grant all on all tables in schema public to administrator;
```

**Trainer →**

```
Drop Role if exists trainer;
create role trainer;
revoke all on all tables in schema public to trainer;
grant select on all tables in schema public to trainer;
grant update on MedicalHistory, Injuries to trainer;
grant insert on MedicalHistory, Injuries to trainer;
grant delete on MedicalHistory, Injuries to trainer;
```

# Security Continued

**Coach →**

```
Drop Role if exists coach;
create role coach;
revoke all on all tables in schema public from coach;
grant select on all tables in schema public to coach;
grant insert on Players, Team, Competition to coach;
grant update on Players, Team, Competition to coach;
grant delete on Players to coach;
```

**Player →**

```
Drop Role if exists player;
create role player;
revoke all on all tables in schema public from player;
grant select on Team, Competition to player;
```

# Implementation Notes – Known Problems – Future Enhancements

If I were able to include the information for all people involved in the athletic department at Marist, more complex queries and especially queries using views would be facilitated. Also, the very first person involved in the athletic department is the database consultant himself, Alan Labouseur with a Pid = 0.

Currently, the database has some issues. The most obvious is that there is no way to implement player stats in the current model. Because statistics are one of the most important things in sports, tables would need to be added in order to incorporate this. Also there is no way for trainers to help out on more than one team when in reality, due to short staffing, some trainers actually work with 2 or 3 teams during one year.

Due to the small scope of the project design some aspects were limited. This opened my eyes to some things that could be implemented on a larger athletic department or even a conference wide level. For example, if GPA was included with every athlete in say a MAAC or Pioneer Football League level database, administrators would be able to query that in order to name the conference all academic teams. Similar to this example, with a larger set of data, greater and more complex records, views, stored procedures and triggers could be implemented and used to facilitate the gathering of desired information as well as to help operations run more smoothly.