Database: Overwatch

Designed by Danny Mulick

Fall 2016

Table of Contents

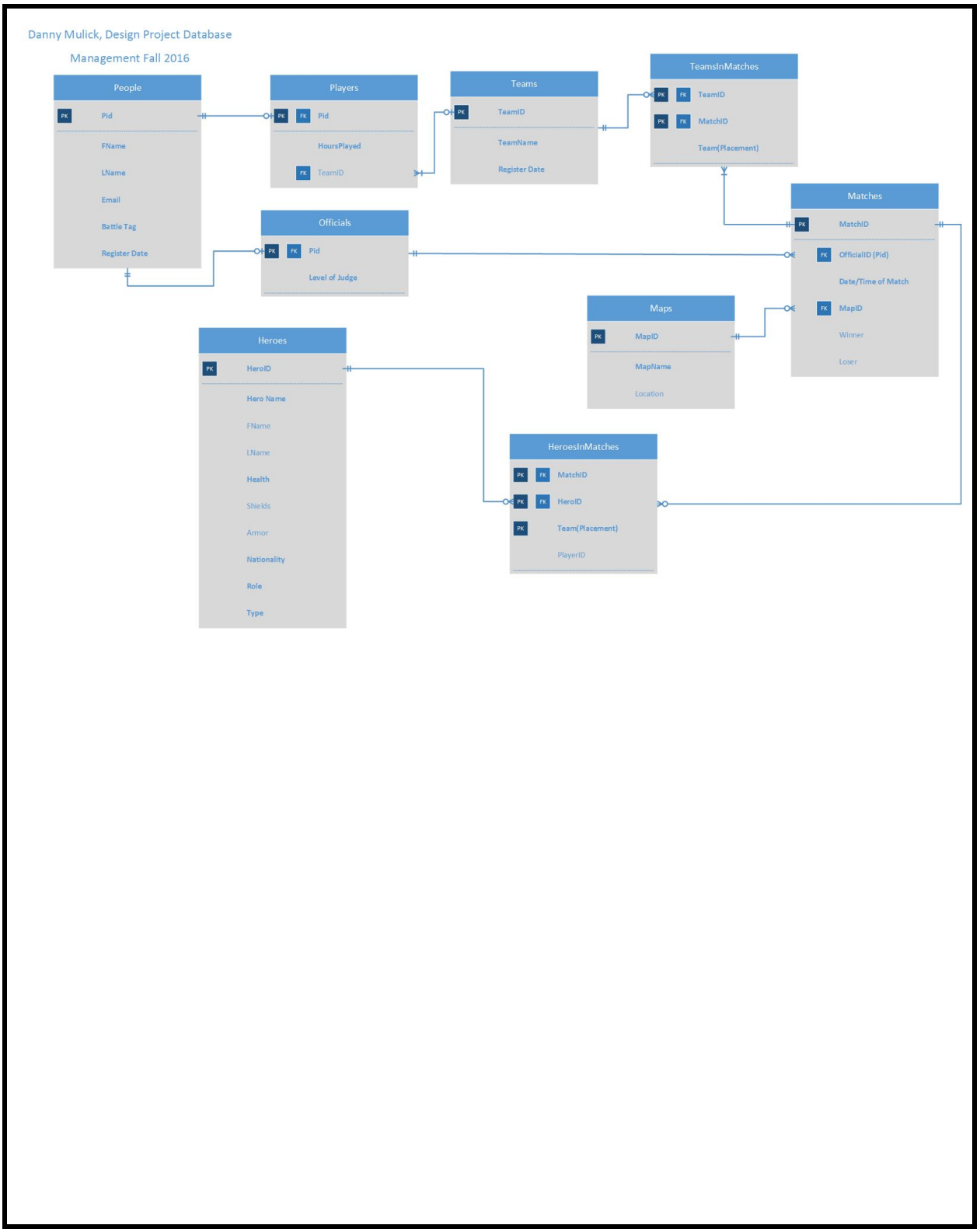
Executive Summary

The coordination and recording of matches in the Overwatch public tournament circuit has always been a hassle. Between the disputes over match rulings that arise from the lack of officials, to conflicts over what team has contracts to which players, and even to the minutia such as what hero was played on which team in what match, there is a need for a system to be put in place to aid in the tournaments' running.

Database: Overwatch is my proposed database that would remedy those issues for the tournament circuit. It would serve to manage all of the tournament's people, including officials and players, as well as record a match's results. Careful constraints and procedures on how to record matches and obtain the contact information of the officials would drastically cut back on errors and misinformation.

Following this summary there is an outline of the database which showcases its design. Database designed and tested in PostgreSQL 9.5.

Entity Relationship Diagram



Database Tables

- **Purpose:**
  - **This table is used to keep track of all people that reside within the database, as well as certain information about those people.**
- **Create Statement:**

```
CREATE TABLE people(
        pid             CHAR(4)     PRIMARY KEY,
        fName           CHAR(20)    NOT NULL,
        lName           CHAR(20)    NOT NULL,
        email           CHAR(30)    NOT NULL,
        battleTag       CHAR(20)    NOT NULL,
        regDate         DATE        NOT NULL
);
```

- **Dependencies:**
  - **Pid → fName, lName, email, battleTag, regDate**
- **Sample Data:**

Data Output | Explain | Messages | History

	pid character(4)	fname character(20)	lname character(20)	email character(30)	battletag character(20)	regdate date
1	P001	Mark	Hanson	mhanson@gmail.com	Handson#6364	2015-07-06
2	P002	John	Jacobs	jj332@gmail.com	JayJay#9909	2015-07-06
3	P003	Paul	Saunders	pauls4@gmail.com	pjsalt32#1115	2015-07-06
4	P004	Juli	Peters	jpeters@gmail.com	majorPete#4412	2015-07-06
5	P005	Anna	Black	blacka@aol.com	superNero#6522	2015-07-06
6	P006	Christian	Noel	noels@gmail.com	neverNoel#7776	2015-07-06
7	P007	Alan	Labouseur	alan@labouseur.com	alan#1337	2015-07-06
8	P008	Thomas	Famularo	tom.fam@marist.edu	nPPredator#1100	2015-07-06
9	P009	Jack	Wilson	wileyj@gmail.com	KcajWils#1213	2015-08-09
10	P010	Alex	Burns	burner@aol.com	Burny#4445	2015-08-10
11	P011	Lauren	Rannet	Laurannet@gmail.com	Rannit#7472	2015-08-16
12	P012	Peter	Parker	petep@gmail.com	Spooder#3332	2015-11-20
13	P013	Dennis	Murray	djm@marist.edu	DJMaster#4123	2015-12-26
14	P014	Daniel	Zhang	daniel.zhang@marist.edu	zhanged#0012	2016-12-30
15	P015	Gary	Amaz	amaz@battle.net	amaz7#1544	2016-01-06
16	P016	Max	Levitt	maxlev@gmail.com	maxlevel#1412	2016-01-08
17	P017	Carly	Rae	raecee@gmail.com	raycee#7784	2016-01-08
18	P018	Nick	Soun	nick.soun1@marist.edu	souns#9016	2016-01-15
19	P019	Tony	Redson	redTon@gmail.com	redz#2232	2016-01-30
20	P020	Mark	Shlep	mark.shlep@ualbany.edu	shlepper#8745	2016-02-09

- **Purpose:**
  - **This table was built to separate the teams and clearly define information about them.**
- **Create Statement:**

```
CREATE TABLE teams(
        teamID          CHAR(4)     NOT NULL PRIMARY KEY,
        teamName        CHAR(30)    NOT NULL,
        regDate         DATE        NOT NULL
);
```

- **Dependencies:**
  - **teamID → teamName, regDate**
- **Sample Data:**

Data Output | Explain | Messages | History

	teamid character(4)	teamname character(30)	regdate date
1	T001	Team Liquid	2015-07-06
2	T002	TSM	2015-07-06
3	T003	Fnatic	2015-07-06
4	T004	Cloud9	2015-08-11
5	T005	Dignitas	2015-11-23

- **Purpose:**
  - **A table made to keep track of people designated as players as well as some statistics about them.**
- **Create Statement:**

```
CREATE TABLE players(
        playerID      CHAR(4)     NOT NULL PRIMARY KEY REFERENCES people(pid),
        hoursPlayed   INT         NOT NULL,
        team          CHAR(4)     NOT NULL REFERENCES teams(teamID)
);
```

- **Dependencies:**
  - **playerID → hoursPlayed, team**
- **Sample Data:**

Data Output | Explain | Messages | History

	playerid character(4)	hoursplayed integer	team character(4)
1	P008	20	T002
2	P006	10	T001
3	P001	10	T002
4	P002	16	T003
5	P003	8	T002
6	P005	7	T002
7	P004	4	T001
8	P011	19	T001
9	P015	22	T003
10	P009	27	T003
11	P010	5	T001
12	P018	16	T003
13	P019	40	T005
14	P014	60	T005
15	P017	45	T005
16	P016	32	T005

- **Purpose:**
  - **Table for holding people who are designated as officials for games.**
- **Create Statement:**

```
CREATE TABLE officials(
        officialID    CHAR(4)     NOT NULL PRIMARY KEY REFERENCES people(pid),
        judgeLevel    INT         NOT NULL
);
```

- **Dependencies:**
  - **officialID → judgeLevel**
- **Sample Data:**

Data Output | Explain | Messages | History

	playerid character(4)	hoursplayed integer	team character(4)
1	P008	20	T002
2	P006	10	T001
3	P001	10	T002
4	P002	16	T003
5	P003	8	T002
6	P005	7	T002
7	P004	4	T001
8	P011	19	T001
9	P015	22	T003
10	P009	27	T003
11	P010	5	T001
12	P018	16	T003
13	P019	40	T005
14	P014	60	T005
15	P017	45	T005
16	P016	32	T005

- **Purpose:**
  - **This table will be used to hold the maps for matches to be played on.**
- **Create Statement:**

```
CREATE TABLE maps(
        mapID          CHAR(4)     NOT NULL PRIMARY KEY,
        mapName        CHAR(30)    NOT NULL,
        LOCATION       CHAR(30)
);
```

- **Dependencies:**
  - **mapID → mapName, location**
- **Sample Data:**

Data Output | Explain | Messages | History

	mapid character(4)	mapname character(30)	location character(30)
1	MP01	Hanamura	Japan
2	MP02	Temple of Anubis	Egypt
3	MP03	Volskaya Industries	Russia
4	MP04	Dorado	Mexico
5	MP05	Watchpoint: Gibraltar	Gibraltar
6	MP06	Route 66	United States
7	MP07	Kings Row	England
8	MP08	Numbani	Western coast of Africa
9	MP09	Hollywood	Los Angeles, United States
10	MP10	Eichenwalde	Stuttgart, Germany
11	MP11	Nepal	Nepal
12	MP12	Lijang Tower	China
13	MP13	Ilios	Greece
14	MP14	Ecopoint: Antarctica	Antarctica

- **Purpose:**
  - **This table will be built to house critical information about matches.**
- **Create Statement:**

```
CREATE TABLE matches(
        matchID       CHAR(4)    NOT NULL PRIMARY KEY,
        officialID    CHAR(4)    NOT NULL REFERENCES officials(officialID),
        timeOfMatch   TIMESTAMP  NOT NULL,
        map           CHAR(4)    NOT NULL REFERENCES maps(mapID),
        winner        CHAR(4)    NOT NULL REFERENCES teams(teamID),
        loser         CHAR(4)    NOT NULL REFERENCES teams(teamID)
);
```

- **Dependencies:**
  - **matchID → officialID, timeOfMatch, map, winner, loser**
- **Sample Data:**

Data Output | Explain | Messages | History

	matchid character(4)	officialid character(4)	timeofmatch timestamp without time zone	map character(4)	winner character(4)	loser character(4)
1	M001	P007	2015-07-15 19:00:00	MP03	T001	T002
2	M002	P007	2015-07-15 19:00:00	MP04	T002	T003
3	M003	P012	2015-11-30 19:00:00	MP02	T001	T003
4	M004	P007	2015-12-01 19:00:00	MP02	T001	T002
5	M005	P007	2015-12-12 19:00:00	MP01	T003	T001
6	M006	P007	2015-12-20 19:00:00	MP09	T005	T001
7	M007	P020	2016-02-15 19:00:00	MP07	T003	T002
8	M008	P020	2016-02-20 19:00:00	MP01	T005	T003
9	M009	P020	2016-03-02 19:00:00	MP01	T005	T003
10	M010	P020	2016-03-03 19:00:00	MP12	T005	T003

- **Purpose:**
  - **A table used to represent what teams appear in which matches**
- **Create Statement:**

```
CREATE TABLE teamsInMatches(
        teamID        CHAR(4)     NOT NULL REFERENCES teams(teamID),
        matchID       CHAR(4)     NOT NULL REFERENCES matches(matchID),
        teamNum       INT         NOT NULL,
        PRIMARY KEY   (teamID, matchID)
);
```

- **Dependencies:**
  - **teamID, matchID → teamNum**
- **Sample Data:**

Data Output | Explain | Messages | History

	teamid character(4)	matchid character(4)	teamnum integer
1	T001	M001	1
2	T002	M001	2
3	T002	M002	1
4	T003	M002	2
5	T001	M003	1
6	T003	M003	2
7	T001	M004	1
8	T002	M004	2
9	T003	M005	1
10	T001	M005	2
11	T005	M006	1
12	T001	M006	2
13	T003	M007	1
14	T002	M007	2
15	T003	M008	1
16	T005	M008	2
17	T005	M009	1
18	T003	M009	2
19	T005	M010	1
20	T003	M010	2

- **Purpose:**
  - **A table used to house information about the heroes within the game**
- **Create Statement:**

```
CREATE TABLE heroes(
        heroID          CHAR(4)     NOT NULL PRIMARY KEY,
        heroName        CHAR(15)    NOT NULL,
        fName           CHAR(15)    ,
        lName           CHAR(20)    ,
        health          INT         NOT NULL,
        shields         INT         ,
        armor           INT         ,
        nationality     CHAR(20)    NOT NULL,
        -- role set needs to be made - Offense, Defense, Tank, Support
        ROLE            ROLE        NOT NULL,
        -- type is based off the possible subtypes of hero i.e. sniper, healer, builder.
        -- not all heroes can have a type
        heroType        CHAR(10)
);
```

- **Dependencies:**
  - **heroID → heroName, fName, lName, health, shields, armor, nationality, role, heroType**
- **Sample Data:**

Data Output | Explain | Messages | History

	heroid character(4)	heroname character(15)	fname character(15)	lname character(20)	health integer	shields integer	armor integer	nationality character(20)	role role	herotype character(10)
1	H001	Genji	Genji	Shimada	200	0	0	Japanese	Offense	
2	H002	McCree	Jesse	Mecree	200	0	0	American	Offense	
3	H003	Pharah	Fareeha	Amari	200	0	0	Egyptian	Offense	
4	H004	Reaper	Gabriel	Reyes	250	0	0	American	Offense	
5	H005	Soldier: 76	Jack	Morrison	200	0	0	American	Offense	
6	H006	Sombra			200	0	0	Mexican	Offense	
7	H007	Tracer	Lena	Oxford	150	0	0	English	Offense	
8	H008	Bastion	Siege Automaton	E54	200	0	100	Omnic	Defense	
9	H009	Hanzo	Hanzo	Shimada	200	0	0	Japanese	Defense	Sniper
10	H010	Junkrat	Jamison	Fawkes	200	0	0	Australian	Defense	
11	H011	Mei	Mei-Ling	Zhou	250	0	0	Chinese	Defense	
12	H012	Torbjörn	Torbjörn	Lindholm	200	0	0	Swedish	Defense	Builder
13	H013	Widowmaker	Amélie	Lacroix	200	0	0	French	Defense	Sniper
14	H014	D.Va	Hana	Song	150	0	400	Korean	Tank	
15	H015	Reinhardt	Reinhardt	Wilhelm	300	0	200	German	Tank	
16	H016	Roadhog	Mako	Rutledge	600	0	0	Australian	Tank	
17	H017	Winston	Winston		400	0	100	Lunarian	Tank	
18	H018	Zarya	Aleksandra	Zaryanova	200	200	0	Russian	Tank	
19	H019	Ana	Ana	Amari	200	0	0	Egyptian	Support	Sniper
20	H020	Lucio	Lucio	Correia dos Santos	200	0	0	Brazilian	Support	Healer
21	H021	Mercy	Angela	Ziegler	200	0	0	Swiss	Support	Healer
22	H022	Symmetra	Satya	Vaswani	100	100	0	Indian	Support	Builder
23	H023	Zenyatta	Tekhartha	Zenyatta	50	150	0	Omnic	Support	Healer

- **Purpose:**
  - **Table used to represent what heroes appear in differing matches**
- **Create Statement:**

```
CREATE TABLE heroesInMatches(
        matchID       CHAR(4)     NOT NULL REFERENCES matches(matchID),
        heroID        CHAR(4)     NOT NULL REFERENCES heroes(heroID),
        teamNum       INT         NOT NULL,
        playerID      CHAR(4)     NOT NULL,
        PRIMARY KEY   (matchID, heroID, teamNum)
);
```

- **Dependencies:**
  - **matchID, heroID, teamNum → playerID**
- **Sample Data:**

Data Output | Explain | Messages | History

	matchid character(4)	heroid character(4)	teamnum integer	playerid character(4)
1	M001	H021	1	4
2	M001	H001	1	4
3	M001	H008	1	4
4	M001	H017	1	4
5	M001	H021	2	4
6	M001	H018	2	4
7	M001	H010	2	4
8	M001	H009	2	4
9	M002	H020	1	4
10	M002	H016	1	4
11	M002	H011	1	4
12	M002	H007	1	4
13	M002	H023	2	4
14	M002	H014	2	4
15	M002	H004	2	4
16	M002	H003	2	4

Views

- **Purpose:**
  - **Return a window into the data to showcase what maps are played the most**
- **Definition:**

```
CREATE VIEW MostPlayedMaps
AS
SELECT mapid, mapname, LOCATION, count(*) FROM matches
 INNER JOIN maps ON matches.map = maps.mapID
 GROUP BY mapID
 ORDER BY count DESC;
```

- **Sample Data:**

Data Output | Explain | Messages | History

	mapid character(4)	mapname character(30)	location character(30)	count bigint
1	MP01	Hanamura	Japan	3
2	MP02	Temple of Anubis	Egypt	2
3	MP03	Volskaya Industries	Russia	1
4	MP07	Kings Row	England	1
5	MP12	Lijang Tower	China	1
6	MP04	Dorado	Mexico	1
7	MP09	Hollywood	Los Angeles, United States	1

- **Purpose:**
  - **Return a window into the data to showcase what heroes are played the most**
- **Definition:**

```
CREATE VIEW MostPlayedHeroes
AS
SELECT h1.heroID, h1.heroName, count(*) AS totalMatches FROM matches
 INNER JOIN heroesInMatches        AS h  ON m.matchID = h.matchID
 INNER JOIN heroes                 AS h1 ON h.heroID = h1.heroID
 GROUP BY h1.heroID
 ORDER BY totalMatches DESC;
```

- **Sample Data:**

Data Output | Explain | Messages | History

	heroid character(4)	heroname character(15)	totalmatches bigint
1	H020	Lucio	6
2	H015	Reinhardt	5
3	H021	Mercy	5
4	H011	Mei	5
5	H001	Genji	5
6	H014	D.Va	4
7	H010	Junkrat	4
8	H018	Zarya	4
9	H016	Roadhog	4
10	H012	Torbjörn	4
11	H004	Reaper	3
12	H003	Pharah	3
13	H023	Zenyatta	3
14	H009	Hanzo	3
15	H013	Widowmaker	3
16	H008	Bastion	3
17	H019	Ana	3
18	H006	Sombra	3
19	H007	Tracer	3
20	H017	Winston	2
21	H005	Soldier: 76	2
22	H002	McCree	2
23	H022	Symmetra	1

Trigger

- **Purpose:**
  - **This trigger is to be used to prevent duplicate entries of a hero on the same team in the same match. Only one of each hero may be allowed on a particular team.**
- **Definition:**

```
CREATE OR REPLACE FUNCTION oneHero()
RETURNS TRIGGER AS
$$
DECLARE
        result CHAR(100);
BEGIN
     IF new.heroID IN (SELECT heroID FROM heroesInMatches AS h
                       WHERE h.matchID = new.matchID AND h.teamNum = new.teamNum)
     THEN
        RAISE EXCEPTION 'Sorry, but your hero already exists in that match on that team. Please select another.';
        ROLLBACK;
     ELSE
        RETURN new;
     END IF;
END;
$$
LANGUAGE plpgsql;
```

- **Creation:**

```
DROP TRIGGER IF EXISTS oneHero ON heroesInMatches;
CREATE TRIGGER oneHero
    BEFORE INSERT
    ON heroesInMatches
    FOR EACH ROW
    EXECUTE PROCEDURE oneHero();
```

- **Sample of trigger:**

Data Output | Explain | Messages | History

```
ERROR:  Sorry, but your hero already exists in that match on that team. Please select another.
CONTEXT:  SQL statement "insert into heroesInMatches (matchID, heroID, teamNum, playerID)
   values (_matchID, _heroID, _teamNum, _playerID)"
PL/pgSQL function insertheroesintomatches(character,character,integer,character) line 10 at SQL statement
********** Error **********

ERROR: Sorry, but your hero already exists in that match on that team. Please select another.
SQL state: P0001
Context: SQL statement "insert into heroesInMatches (matchID, heroID, teamNum, playerID)
   values (_matchID, _heroID, _teamNum, _playerID)"
PL/pgSQL function insertheroesintomatches(character,character,integer,character) line 10 at SQL statement
```

Stored Procedures

For each table in the database, I have made a stored procedure for inserting data into that table. These procedures are:

1. **insertMatch**
2. **insertPerson**
3. **insertOfficial**
4. **insertTeam**
5. **insertPlayer**
6. **insertMap**

7. **insertHero**
8. **insertHeroesIntoMatches**
9. **insertTeamsIntoMatches**

- **Example of Insert Stored Procedure**

```
-- 5 teams
SELECT * FROM insertTeam('T001', 'Team Liquid', '2015-07-06');
SELECT * FROM insertTeam('T002', 'TSM',         '2015-07-06');
SELECT * FROM insertTeam('T003', 'Fnatic',      '2015-07-06');
SELECT * FROM insertTeam('T004', 'Cloud9',      '2015-08-11');
SELECT * FROM insertTeam('T005', 'Dignitas',    '2015-11-23');
--3 officials
SELECT * FROM insertOfficial('P007', 5);
SELECT * FROM insertOfficial('P020', 2);
SELECT * FROM insertOfficial('P012', 1);
-- 16 total players
SELECT * FROM insertPlayer('P008', 20, 'T002');
SELECT * FROM insertPlayer('P006', 10, 'T001');
SELECT * FROM insertPlayer('P001', 10, 'T002');
SELECT * FROM insertPlayer('P002', 16, 'T003');
SELECT * FROM insertPlayer('P003', 8, 'T002');
SELECT * FROM insertPlayer('P005', 7, 'T002');
SELECT * FROM insertPlayer('P004', 4, 'T001');
SELECT * FROM insertPlayer('P011', 19, 'T001');
SELECT * FROM insertPlayer('P015', 22, 'T003');
```

In addition, I have designed other stored procedures.

- **<u>Purpose:</u>**
  - **Return a set of data that contains all heroes that appear in a match on a certain team in that match.**
- **<u>Create Statement:</u>**

```
CREATE OR REPLACE FUNCTION heroesInAMatch(CHAR(4), INT, refcursor) RETURNS refcursor
AS
$$
DECLARE
-- use underscore sign notation to declare variables, helps to recycle names and
--   make easier to remember
   _matchID      CHAR(4)     :=$1;
   _teamNum      INT         :=$2;
   resultset     refcursor   :=$3;
BEGIN
   open resultset FOR
        SELECT heroID
          FROM heroesInMatches AS hIM
         WHERE _matchId = hIM.matchID AND _teamNum = hIM.teamNum;
   RETURN resultset;
END;
$$
LANGUAGE plpgsql;
```

- **Sample Data:**

```
SELECT heroesInAMatch('M002', 2, 'results');
FETCH ALL FROM results;
```

Data Output | Explain | Messages | History

	heroid character(4)
1	H023
2	H014
3	H004
4	H003

- **Purpose:**
  - **This table is a reference to look at what players exist on a given team**
- **Create Statement:**

```
CREATE OR REPLACE FUNCTION playersOnATeam(CHAR(4), refcursor) RETURNS refcursor
AS
$$
DECLARE
-- use underscore sign notation to declare variables, helps to recycle names and
--   make easier to remember
   _teamID      CHAR(4)     :=$1;
   resultset    refcursor   :=$2;
BEGIN
   open resultset FOR
        SELECT people.*, teams.*, players.hoursPlayed
          FROM teams
         INNER JOIN players ON teams.teamID = players.team
         INNER JOIN people  ON players.playerID = people.pid
         WHERE teams.teamID = _teamID;
   RETURN resultset;
END;
$$
LANGUAGE plpgsql;
```

- **Sample Data:**

```
SELECT playersOnATeam('T005', 'results');
FETCH ALL FROM results;
```

Data Output | Explain | Messages | History

	pid character(4)	fname character(20)	lname character(20)	email character(30)	battletag character(20)	regdate date	teamid character(4)	teamname character(30)	regdate date	hoursplayed integer
1	P014	Daniel	Zhang	daniel.zhang@marist.edu	zhanged#0012	2016-12-30	T005	Dignitas	2015-11-23	60
2	P016	Max	Levitt	maxlev@gmail.com	maxlevel#1412	2016-01-08	T005	Dignitas	2015-11-23	32
3	P017	Carly	Rae	raecee@gmail.com	raycee#7784	2016-01-08	T005	Dignitas	2015-11-23	45
4	P019	Tony	Redson	redTon@gmail.com	redz#2232	2016-01-30	T005	Dignitas	2015-11-23	40

- **Purpose:**
  - **This table will be used to show the information for an official**
- **Create Statement:**

```
CREATE OR REPLACE FUNCTION officialInfo(CHAR(4), refcursor) RETURNS refcursor
AS
$$
DECLARE
-- use underscore sign notation to declare variables, helps to recycle names and
--   make easier to remember
   _matchID       CHAR(4)    :=$1;
   resultset      refcursor  :=$2;
BEGIN
   open resultset FOR
        SELECT m.officialID, o.judgelevel, p.*
          FROM matches m
         INNER JOIN officials o ON o.officialID = m.officialID
         INNER JOIN people p    ON p.pid        = m.officialID
         WHERE m.matchID = _matchID;
   RETURN resultset;
END;
$$
LANGUAGE plpgsql;
```

- **Sample Information:**

```
SELECT officialInfo('M007', 'results');
FETCH ALL FROM results;
```

Data Output | Explain | Messages | History

	officialid character(4)	judgelevel integer	pid character(4)	fname character(20)	lname character(20)	email character(30)	battletag character(20)	regdate date
1	P020	2	P020	Mark	Shlep	mark.shlep@ualbany.edu	shlepper#8745	2016-02-09

Security

The database is designed to support roles of ADMIN, OFFICIAL, and teamManager.

Role defining statements:

```
DROP   ROLE   IF EXISTS ADMIN;
DROP   ROLE   IF EXISTS Official;
DROP   ROLE   IF EXISTS teamManager;
CREATE ROLE   ADMIN;
CREATE ROLE   Official;
CREATE ROLE   teamManager;

-- admin has rights to everything
GRANT SELECT, INSERT, UPDATE, DELETE ON people          TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON players         TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON officials       TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON teams           TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON maps            TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON matches         TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON teamsInMatches  TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON heroes          TO ADMIN;
GRANT SELECT, INSERT, UPDATE, DELETE ON heroesInMatches TO ADMIN;

--teamManager has rights to their own team and the table to show they are in that match i.e. signing up for the match
REVOKE ALL PRIVILEGES ON people          FROM teamManager;
REVOKE ALL PRIVILEGES ON matches         FROM teamManager;
REVOKE ALL PRIVILEGES ON officials       FROM teamManager;
REVOKE ALL PRIVILEGES ON heroes          FROM teamManager;
REVOKE ALL PRIVILEGES ON heroesInMatches FROM teamManager;
REVOKE ALL PRIVILEGES ON maps            FROM teamManager;

GRANT SELECT, INSERT, UPDATE         ON teams                   TO teamManager;
GRANT SELECT, UPDATE                 ON players                 TO teamManager;
GRANT SELECT, INSERT, UPDATE         ON teamsInMatches          TO teamManager;

--official is able to call a match, and view info on other officials
REVOKE ALL PRIVILEGES ON people          FROM official;
REVOKE ALL PRIVILEGES ON players         FROM official;
REVOKE ALL PRIVILEGES ON teams           FROM official;
REVOKE ALL PRIVILEGES ON teamsInMatches  FROM official;
REVOKE ALL PRIVILEGES ON heroes          FROM official;
REVOKE ALL PRIVILEGES ON maps            FROM official;
REVOKE ALL PRIVILEGES ON heroesInMatches FROM official;

GRANT SELECT, INSERT, UPDATE ON matches  TO   official;
```

Implementation Notes

I assumed the tournaments would be played in the Competitive style, meaning that there would be no duplicates of a hero on a single team within a match. Also, I created teams that had a max of four people, while in game a team can have a max of 6 per game. I had created the stored procedures to insert data as a way of making the administrator's life easier, as someone other than him would be able to manipulate data with the correct privileges.

Known Issues

Some known issues of this system is the lack of abilities and data concerning them. There is minimal stats for heroes when they are not related to a player.

Future Enhancements

Future enhancements could include allowing for multiples of the same hero on the same team, adding in more maps, heroes, people, teams, or other entities.