**ZombieFox:**

**App Based Human Vs Zombie Game**



**Created by**

**Michael Shershin**

Table of Contents

Executive Summary

Entity Relationship Diagram

Tables: Create Statements, Functional Dependencies, and Sample Data

People

Game

Passwords

Player Administration

Killed

Signed up

Zombie

Enums:

Year

Area

Views:

Amount of Zombie Wins

Amount of Fox Wins

Predator

Prey

Stored Procedures:

Signed up to Battle

Zombies to Battle

Security

Problems

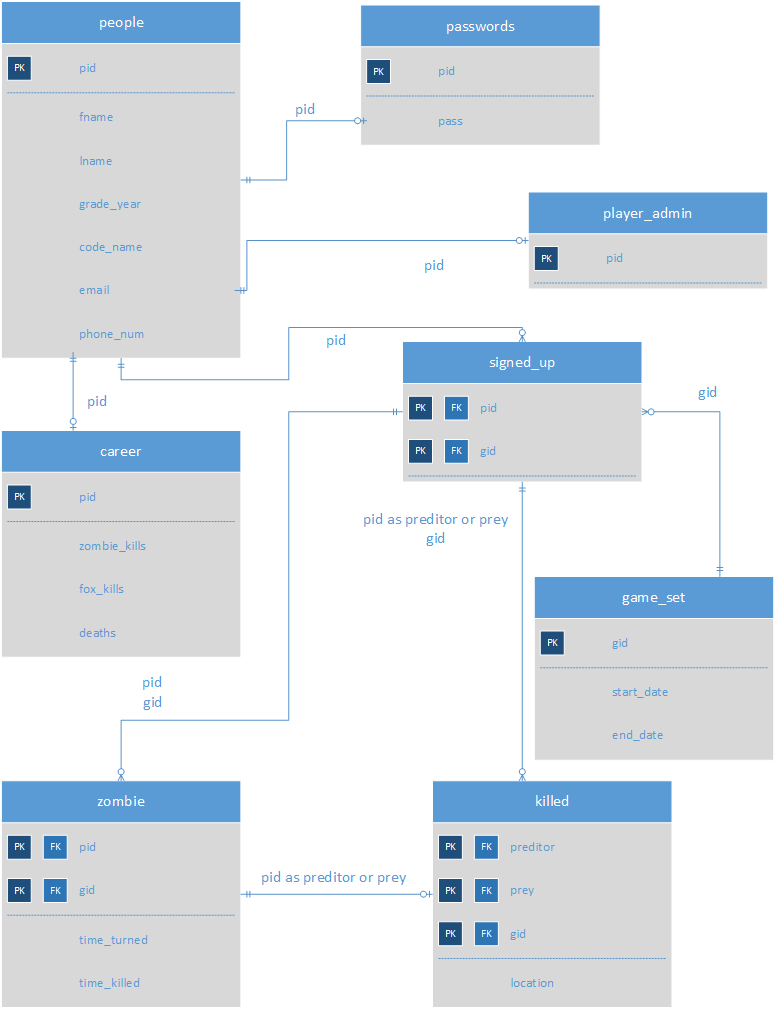
Plans for the Future

**Executive Summary**

This document is a strong representation of my design and implementation for my group’s app, ZombieFox. This database is a way to see all the players in the app as well as their current role in the game. The users would be anyone who uses the app and play the game.

ZombieFox is an app based off of the popular college game Humans Vs Zombies. The game consists of two groups of people Humans and Zombies, and they both hunt each other for victory. This app allows the users to have access to who is playing the game when and where ever they are.

To start things off I have the Entity Relationship Diagram to show all the tables and their relations. Then comes the tables and their create statements and a bit of data that is in each table. Followed by enums, views and stored procedures. Next will be Security, followed by known problems and plans for the future.

**Entity Relationship Diagram**

**Tables**

**People table**

The people table is the starting point for all users of the app.

CREATE TABLE people(

pid serial not null,

fname text,

lname text,

grade\_year year,

code\_name text UNIQUE,

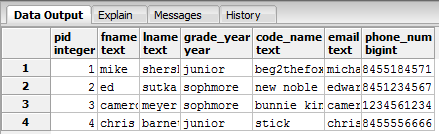
email text UNIQUE,

phone\_num bigint UNIQUE,

primary key(pid));

Functional Dependencies

pid -> fname, lname, grade\_year, code\_name, email, phone\_num.

Output:

**Game table**

The game table is where each session of a game is started.

CREATE TABLE game(

gid SERIAL not null,

start\_date TIMESTAMP,

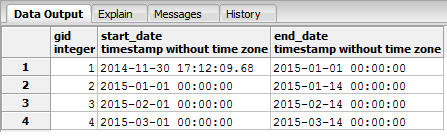
end\_date TIMESTAMP,

primary key(gid));

Functional Dependencies

gid -> start\_date, end\_date

Output:



**Passwords**

I decided to create the password table for added security.

CREATE TABLE passwords(

pid SERIAL not null references people(pid),

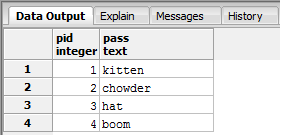
pass text,

primary key(pid));

Functional Dependencies

pid -> pass

Output:



**Player Admin table**

This table is to identify if a user of the app has admin power within the app itself and not so much in the database.

CREATE TABLE player\_admin(

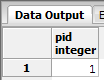
pid SERIAL not null references people(pid),

primary key(pid));

Functional Dependencies

pid

Output:



**Killed Table**

The killed table is to keep track of which player killed whom in a given game.

CREATE TABLE killed(

preditor SERIAL not null references people(pid),

prey SERIAL not null references people(pid),

gid SERIAL not null references game(gid),

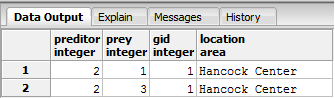
location area,

primary key(preditor, prey, gid));

Functional Dependencies

preditor, prey, gid -> location

Output:



**Signed Up Table**

The signed up table is to show which players are signed up for a select game.

CREATE TABLE signed\_up(

pid SERIAL not null references people(pid),

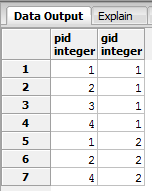
gid SERIAL not null references game(gid),

primary key(pid, gid));

Functional Dependencies

Pid, gid

Output:



**Career Table**

The career table keep tracks of a players personal stats with how many times they have killed another player and how many times they died.

CREATE TABLE career(

pid SERIAL not null references people(pid),

zombie\_kills int,

fox\_kills int,

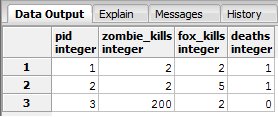
deaths int,

primary key(pid));

Functional Dependencies

pid -> zombie\_kills, fox\_kills, deaths

Output:



**Zombie Table**

The zombie table is so we can see out of the players signed up for a game who are zombies.

CREATE TABLE zombie(

pid serial not null references people(pid),

gid SERIAL not null references game(gid),

time\_turned TIMESTAMP,

time\_killed TIMESTAMP,

primary key(pid, gid));

Functional Dependencies

pid, gid -> time\_turned, time\_killed

Output:

