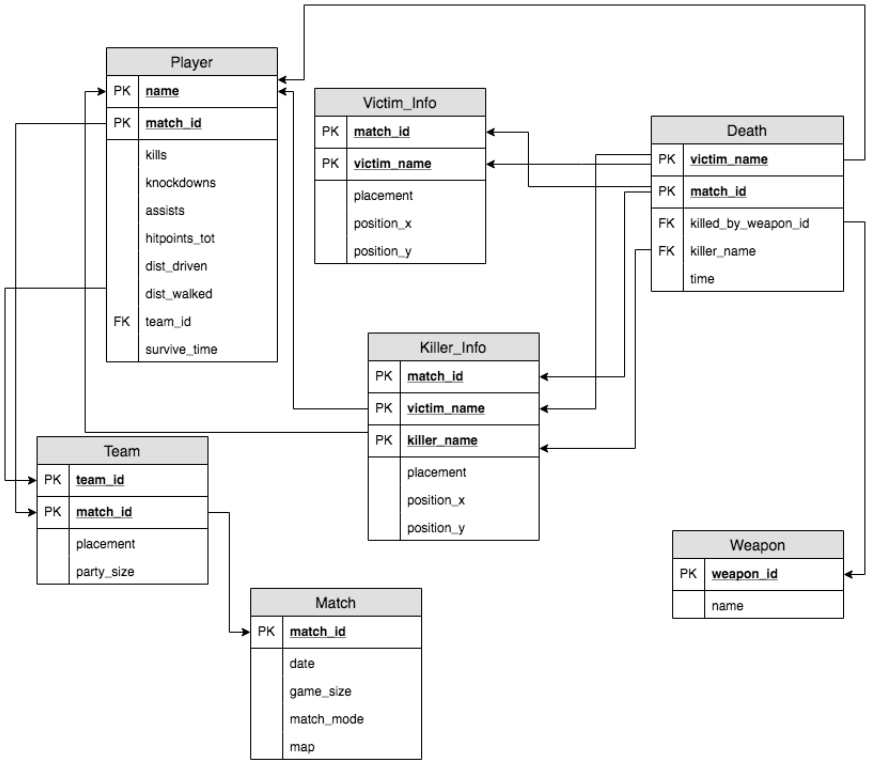
Project README - group 8

Adam Spindler, Pranshu Bheda,
Ted Xie, Aravind Vicinthangal Prathivaathi

February 2019

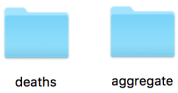
1 PUBG Data Schema



2 Separating the Data File

1. First of all is to split the data set by programming, since we only have two csv files folder (one is deaths and one is aggregate) and we need to

separate them in to different files.



Open the parser.py and parser_thread.py,

for the parser_thread.py we need to change the DIRECTORY to the path of your folder with include the data

```
6       DIRECTORY = r"/Users/Zongge/Desktop/pubg-match-deaths/"
```

then for the parser.py, under the main, change the path of the root_directory which will be the folder of your store the data

```
41      root_directory = Path(r"/Users/Zongge/Desktop/pubg-match-deaths/")
```

and then choose the line48 to deaths or aggregate.

```
48      source_data['deaths'] = file_names_deaths
```

Then run the parser.py, and waiting for the split files.

```
/Library/Frameworks/Python.framework/Versions/3.7/bin/python3 /Users/Zongge/PycharmProjects/pgtest/parser.py
Processing file:-  /Users/Zongge/Desktop/pubg-match-deaths/deaths/kill_match_stats_final_0.csv deaths
Start...
Finished...
Processing file:-  /Users/Zongge/Desktop/pubg-match-deaths/deaths/kill_match_stats_final_1.csv deaths
Start...
Finished...
Processing file:-  /Users/Zongge/Desktop/pubg-match-deaths/deaths/kill_match_stats_final_2.csv deaths
Start...
Finished...
Processing file:-  /Users/Zongge/Desktop/pubg-match-deaths/deaths/kill_match_stats_final_3.csv deaths
Start...
Finished...
Processing file:-  /Users/Zongge/Desktop/pubg-match-deaths/deaths/kill_match_stats_final_4.csv deaths
Start...
```

3 Uploading the Data

After all of the original files have been split into CSV files that are representative of the tables, they need to be uploaded directly into the database. The folder DataUploader has a program main.py that is used to upload all of the data. It iterates over all of the rows in each of the CSV files and puts them into the database as records for the appropriate tables.

There are no constraints initially in the database because it is much easier to use postgres and SQL to do the data cleaning. First we take all of the unique weapons and put them into a new table, then drop the old one using this query.

```
CREATE TABLE weapon_unique AS
    SELECT DISTINCT ON(weapon_name) id, weapon_name FROM weapon;
```

Some of the rows in the CSV files are incomplete in the original data set so those need to be removed. These rows have the value '#unknown' for the player name. Those are removed with the following query.

```
UPDATE death d
  SET killed_by_weapon_id = (SELECT id FROM weapon_unique WHERE weapon_name = d.weapon)
  WHERE 1 = 1;
```

After the data has been cleaned, the primary key and foreign key constraints can be added for efficient querying and to ensure data integrity.