EXPLANATION OF HOW PROGRAM WORKS

We obtained the data for football database from the openfootball-worldcup github repository (https://github.com/openfootball/world-cup/)

Initially we created a country code dictionary which lists down all the countries that have participated in the world cup from 1930 to 2014 and their 2 or 3 lettered country code. This has been done by parsing the filenames in all the worldcup folders.

Rest of the document describes the procedure for how we populated the records for each relation in the database.

team :

Filename : team_info.py

For a given country code in the dictionary mentioned above we have obtained the rest of the attributes values by parsing the file 'stats/alltime.txt' and extracting relevant values using the country name.

player :

Filename : player_info.py

For every year in which the world cup took place we scanned through the list of players from each country that participated in the world cup in order to create tuples for this relation. We extracted information from the year--countryname/squad folder. We have generated date of births randomly assuming player ages between 20 and 30.

matches :

Filename : match_info.py

For every year in which the world cup took place we scanned through the list of matches scheduled for that year in order to fill the tuple values for this relation. We extracted information from year—countryname/cup.txt or cup_finals.txt files.

world_cup :

Filename : world_cup_info.py

The hosting country and the year in which world cup took place can be obtained from the name of the folders listed in the repository. Winner and Runner up teams information was obtained in match_info.py while we were scanning through each match information (In this case we require final match to decide winner and runner up for that year).

stadium :

Filename : match_info.py

While populating the match relation tuples we have simultaneously populated the stadium relation tuples (namely stadium name where the match was scheduled and the stadium address)

team_participates_in_world_cup :

Filename : group_info.py

Every year the teams participating in world cup are formed into groups i.e. a team participating in world cup in a particular year is assigned a group. We populated the tuples of this relation by parsing the year—countryname/cup.txt file. (We took care of the foreign constraints with respect to team and world_cup relations)

world_cup_played_by_player :

Filename : player_info.py

Player relations lists all the players without taking into consideration the years in which they participated. The tuples of this relation are filled simultaneously while populating the tuples of player relation by including the year attribute.

match_played_by :

Filename : match_played_by_info.py

We have created two dictionaries in player_info.py : 1. {(year,country_code):list of goalkeepers} 2. {(year,country_code):list of players excluding goalkeeper}.

We have created a list in match_info.py [(match_no,year,country_code,goals_scored,decision)]. Separate tuples are created for each team participating in a match.

The tuples of this relation are populated by traversing through the list and dictionaries mentioned above . Ten players(1) and one goalkeeper(2) are randomly selected from the dictionaries. These are the players who have actually played in this particular match.

goal_and_player_scores_goals :

Filename : match_info.py

We created a dictionary in match_played_by_info.py : {(match_no,year,country_code) : List of 10 players randomly selected for this match excluding goalkeepers.}

We have created a list in match_info.py [(match_no,year,country_code,goals_scored,decision)]

For each goal made in a particular match of a particular year by a team we have randomly selected a player who made this goal among the list of players who played in that match and for that particular team. The time at which goal was scored was also generated randomly. In case the decision was as result of penalty then the time of goal scored was randomly chosen from 120 to

130 minutes (post the normal match duration). This procedure ensures primary and foreign key constraints are met.