Date: May 17th, 2017

Team Members:
Avik Bag (ab3433)
Ryan Efendy (re88)

Application URI:
http://resin.cci.drexel.edu:8080/~re88/neighborhood
http://resin.cci.drexel.edu:8080/~ab3433/neighborhood

(a) High-level description of the domain

The purpose of the application is to help user whether they been living in Philadelphia for a while or planning to move to the City of Brotherly Love, they should be able to identify which neighborhoods is right for them. Each neighborhood is distinct, we came up with several factors to decide if the area is right for you. The entities are neighborhood, schools, demographics, recreational, public transportation, food and shootings. All these will be used to find relational queries matching the user's requirements of finding a neighborhood to live in.

(b) Entities and relationships (key / participation constraints)

Neighborhood	**Schools**	**Demographics**	**Recreational**
neighborhood_name **zipcode**	**school_id** name grade_level type enrollment	**demographic_id** median_age_group majority_ethnicity median_salary unemployment	**rec_id** name type age_group rating price

Public_Transportation	**Property detail**	**Food**	**Shooting_crimes**
transportation_id station type route	**property_id** type status price size bedroom_bathroom	**name** **address** category cuisine rating price	**incident_id** time fatality officer_involved

Business rules for the above entity set:

1. **Neighborhood** - is the main entity that other entity will have a relationship with. It has the **neighborhood_name** and **zipcode** as its primary key.
2. **Schools** - *schools* are located in exactly one *neighborhood*, and a *neighborhood* can have zero or more *schools*.
3. **Demographics** - a set of *demographics* is based off exactly one *neighborhood*, and a *neighborhood* will have zero or many set of *demographic* detail.
4. **Recreational** - *recreational* places are exactly in one *neighborhood*, and a *neighborhood* can have zero or many *recreational* places .
5. **Public transportation** - *public transportation(stations)* are located in exactly one *neighborhood*, and a *neighborhood* can have zero or *many* public transportation(stations)
6. **Property** - A particular property is owned by exactly one *neighborhood*, and a *neighborhood* can zero or many *property*.
7. **Food** - An *eatery(food)* are exactly in one *neighborhood*, and a neighborhood have zero or more places to eat.
8. **Shooting crimes** - *incidents* occur in exactly one neighborhood, and a *neighborhood* may have zero or multiple *incidents*

(c) ER diagram conversion

Since all the relationships follow a one-to-many relationship from the main table, which is the Neighborhood table, each entity will have it's own table. The participation constraint on each of these tables, except the Neighborhood table, will be done by setting the foreign keys to not null. Since these tables are all connected to the Neighborhood table, the key constraint is set by the fact that it is related to the primary key in the Neighborhood table.

create table Neighborhood
create table Food
create table Schools
create table Shooting_Crimes
create table Demographics
create table Recreational
create table Property_Detials
create table Transportation

(d) How we're going to acquire data for the application
OpenDataPhilly(https://www.opendataphilly.org/) which is a catalog of open data in the Philadelphia region as well as visitphilly.com/neighborhoods for our data collectection. Most of the data is fabricated, but looks real as the basic data format is derived from these sites.

(e) How the user will interact with the database
The main user interface will allow the user to run the basic queries from the database, like displaying the tables that are present in the database using the help of buttons that toggle the

display of these table. There are also buttons that execute sql queries that are defined within the application. The sql script is split up into 3 parts, one for setting up the schema of the database, one for data input and lastly a script that contains all the queries that are executed. These are the following queries that can be executed by the user in the user interface.

- The basic select all query that displays a given table of choice from the list of tables that are present in the database.

- A query that provides information on the safety score of the neighborhoods. The safety scores are calculated by using the neighborhood table and the shooting_crimes table. It takes all the neighborhoods and groups the incidents by group, and calculates the safety score by the following equation

  Safety_score = 2*(fatality = true) - (officer_involved = true)

  The reason why the value of fatality gets a multiplier of 2 is because an incidence of fatality should be weighed more than just an officer getting involved, and an officer presence should be a negative value. The lower the safety score is, the safer the neighborhood is.

- There is a query that will give you the list of all the restaurants in Manayunk

- A query that will give you all the neighborhoods that have a Bus service as well as a Subway service in the neighborhood.

- Query that will give you a list of all the neighborhoods that have property listings that are for sale.

- Lastly, the user can query the list of all schools that have a safety score of less than 0, based on the safety score that is defined in the earlier point.