1 Preliminaries

Before starting on this assignment, please be sure to read the General Instructions that are on Piazza (under Resources->Resources->General Resources). If you did Lab1, you should already know how to log in to the class PostgreSQL server. You'll get help on Lab2 in your Discussion Section, not the Lectures, so *be sure to attend Discussion Sections*.

2 Goal

The goal of the second assignment is to create a PostgreSQL data schema with 5 tables that are very similar to the tables that you created in Lab2. The tables have the same names, attributes and data types with the tables of Lab1, and the same primary keys but there are some UNIQUE constraints and some restrictions on NULLs.

After you create the data schema with the 5 tables, you will be required to write some SQL statements that use those tables. You will be given data to load into your tables so that you can test the results of your queries. Testing can prove that a query is wrong, but not that it is right, so be careful. Lab2 is due in two weeks, so you will have an opportunity to discuss the assignment during the Discussion Section in the first week of the assignment, and to discuss issues you have had in writing a solution to the assignment during the Discussion Section of the second week. Instructions for submitting the assignment appear at the end of this document.

3 Lab2 Description

3.1 Create PostgreSQL Schema Lab2

You will create a Lab2 schema to set apart the database tables created in this lab from ones you will create in future, as well as from tables (and other objects) in the default (public) schema. Note that the meaning of schema here is specific to PostgreSQL and distinct from the general meaning. See here for more details on PostgreSQL schemas. You create the Lab2 schema like this:

```
CREATE SCHEMA Lab2;
```

Now that you have created the schema, you want to set Lab2 to be your default schema when you use psql. If you do not set Lab2 as the default schema, then you will have to qualify your table names with the schema name (e.g. Lab2.Customers). To set the default schema, you modify your search path. (For more details, see here.)

```
ALTER ROLE username SET SEARCH_PATH to Lab2;
```

3.2 Create tables

You will create tables in schema Lab2 for the tables Persons, Houses, Landlords, Ownerships and Tenants. The attributes of the tables are the same as the tables of Lab1. Data types for the attribute names in these tables are also the same as the ones specified for the tables of Lab1. The primary keys are also the same. The tables will additionally have the constraints described in the following section.

3.2.1 Constraints

The following attributes cannot be NULL. All other attributes can be (but remember that attributes in Primary Keys also cannot be NULL).

- In Persons: Name
- In Landlords: OwnerSSN
- In Tenants: LeaseTenantSSN and LeaseStartDate

Also, the following must be unique for each table. That means that there cannot be identical rows in a table that have the same (non-NULL) values for all of those attributes.

- In Houses: HouseAddress
- In Landlords: OwnerSSN
- In Tenants: the pair of attributes (HouseId, LeaseTenantSSN)

(The meaning of that last constraint is that a person can't be the lease tenant for two apartments in the same house. Note that we're not making LeaseTentantSSN unique in Tenants, because a person could sign leases in multiple different houses … and might not live in any of them.)

You will write a CREATE TABLE command for each of the five tables. Save the commands in the file create.sql

4 SQL Queries

Below are English descriptions of the 6 SQL queries that you need to write for this assignment, which you will include in files queryX.sql, where X is the number of the query, e.g., the SQL statement for Query 1 will be in the file query1.sql and so forth.

[When writing your queries, you should assume that a landlord/house owner is a person, and you should assume that a lease tenant is a person. Moreover, assume that whenever an attribute refers to a house (HouseId attribute in Persons, Ownerships and Tenants), then there's a row for that house in the Houses table. We will talk about these assumptions more formally when we get to Referential Integrity and Foreign Keys.]

4.1.1 Query 1

Find the names and SSN for persons whose salary is more than 20,000.

4.1.2 Query 2

Find the different names of all persons who live at 1730 Alma Street. The output of your query should not contain duplicates; that is, the same name should not appear more than once.

4.1.3 Query 3

Find the different colors and apartment counts of houses where persons named John Smith live. The output of your query should not contain duplicates; that is, the same color and apartment count should not appear more than once.

4.1.4 Query 4

Find the names of all persons who lease an apartment whose rent is higher than half of that person's salary.

4.1.5 Query 5

Find the addresses and landlord names for single-dwelling homes, i.e., houses which have only one apartment.

4.1.6 Query 6

For the tenancies that have an overdue rent, find the address and apartment number of the tenancy, the ssn, name and salary of the lease tenant, and the name of the landlord of the leased apartment. *In the result for Query6, the attribute for the lease tenant's name should appear as TenantName, and the attribute for the landlord's name should appear as LandlordName.*

5 Testing

While your solution is still a work in progress, it is a good idea to drop all objects from the database every time you run the script, so you can start fresh. Of course, dropping each object may be tedious, and sometimes there may be a particular order in which objects must be dropped. The following commands (which you can put at the top of create.sql if you want), will drop your Lab2 schema (and all objects within it), and then create the (empty) schema again:

```
DROP SCHEMA Lab2 CASCADE;
CREATE SCHEMA Lab2;
```

Before you submit, login to your database via psql and execute your script. As you've learned already, the command to execute a script is: \i <filename>.

Soon, we will provide you a load script named load_values_Lab2.sql, that will load data into the 4 tables of the database. You will be able to execute the script with the command: \i load_values_Lab2.sql. You should test your 6 queries using that data. (You will have to figure out whether answers are correct on your own.) But you may want to test your SQL statements on your own data as well.

6 Submitting

1. Save your scripts for table creations and query statements as create.sql and query1.sql through query6.sql You may add informative comments inside your scripts if you want (the server interprets lines that start with two hyphens as comment lines).

2. Zip the file(s) to a single file with name Lab2_XXXXXXX.zip where XXXXXXX is your 7-digit student ID, for example, if a student's ID is 1234567, then the file that this student submits for Lab2 should be named Lab2_1234567.zip
   To create the zip file you can use the Unix command:

   ```
   zip Lab2_1234567 create.sql query1.sql query2.sql query3.sql query4.sql query5.sql query6.sql
   ```

   (Of course, you use your own student ID, not 1234567.)

3. You should already know how to transfer the files from the UNIX timeshare to your local machine before submitting to canvas. If you are still not familiar with the process, use the instructions we provided at the Lab1 assignment.

4. Lab2 is due by 11:59pm on Sunday, February 5. Late submissions will not be accepted, and there will be no make-up Lab assignments.