1 Preliminaries

Before starting on this assignment, please be sure to read the General Instructions that are on Piazza (under 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 Lab1. The tables have the same names, attributes and data types as 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 have been 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 Airlines, Airports, Flights, Customers and Tickets. 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 Customers: CustName
- In Airports: City and State
- In Tickets: CustID

Also, the following must be unique for the specified table. That is, there cannot be identical rows in that table that have exactly the same (non-NULL) values for all of those attributes (composite unique constraint).

- In Airlines: AirlineName
- In Flights: the 4 attributes AirlineID, Origin, Destination, DepartureTime
- In Tickets: the 3 attributes CustID, AirlineID, FlightNum

(The second constraint says that an airline can't have multiple flights flying from an origin to a destination that leave at the same time. The third constraint says that a customer can only have one ticket on a particular flight. Note that we're assuming that all flights are on the same day.)

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 5 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., your SQL statement for Query 1 will be in the file query1.sql and so forth. Follow the directions as given; you will lose points if you give extra tuples or attributes in your results, or if you have missing or wrong results.

4.1 Query 1

Give the AirlineID for each different airline that has a flight which departs before noon. Your result should appear in alphabetical order.

4.2 Query 2

Give the name for each customer whose name begins with 'W' and who has a ticket on UAL. (UAL is an AirlineID). No customer name should appear more than once in your result.

4.3 Query 3

We're looking for flights that go from airport SFO to airport JFK. Your result should include the name of the airline (not the AirlineID), the flight number, its departure time and arrival time.

4.4 Query 4

Now we're looking for flights that go from airport SFO to airport JFK with exactly one stopover airport (so there's a first flight from SFO to some stopover airport, and a second flight from the same stopover airport to JFK). Assume that flights arrive and depart on-time. The time spent at the stopover airport should be at least 30 minutes but also should be no more than 2 hours. Your result should include FlightNum1, DepartureTime1, ArrivalTime1, StopOverAirportID, FlightNum2, DepartureTime2, ArrivalTime2. Use those names for the attributes in your result.

4.5 Query 5

For each flight that has paid tickets, how many paid tickets are there for that flight? Your result should include the number of the flight (FlightNum) and the flight's AirlineID, as well as the number of paid tickets for that flight, which should appear as the attribute PaidTickets. Note that you should answer this just by looking at only the Tickets relation, without using the Flights table.

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>.

Under Resources→Lab2, we have also given you a load script named *lab2_date_loading.sql*, that will load data into the 5 tables of the database. You will be able to execute the script with the command: \i *lab2_date_loading.sql*. You should test your 5 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 query5.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 generate the zip file you can use the Unix command:

   ```
   zip Lab2_1234567 create.sql query1.sql query2.sql  query3.sql query4.sql query5.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, October 29. Late submissions will not be accepted, and there will be no make-up Lab assignments.

5. Be sure to follow directions about Academic Integrity that are in the Syllabus and Lecture1. If you have any questions about those directions, please speaker to the instructor as soon as possible.