Acknowledgment: Gang Luo lectures

Homework 2: SQL Basics

Objectives:

To create and import databases and to practice simple SQL queries.

Assignment tools: SQLite and the **flights dataset** (a .tar.gz archive)

(To extract the contents of a tar file)

Post on: Monday, March 18th, 2019

Due date: Monday, April 8th, 2019 by 10:00pm.

Turn in your solution using the assignment drop box linked from the main course web page.

What to turn in:

The two files, `create-tables.sql` and `hw2-queries.sql`, described below.

Overview

In this homework, you will write several SQL queries on a relational flights database. The data in this database is from the Bureau of Transportation Statistics, covering the years 2005–2015. The database consists of four tables:

- **Flights** (fid, year, month_id, day_of_month, day_of_week_id, carrier_id, flight_num, origin_city, origin_state, dest_city, dest_state, departure_delay, taxi_out, arrival_delay, canceled, actual_time, distance)
- **Carriers** (cid, name)
- **Months** (mid, month)
- **Weekdays** (did, day_of_week)

We leave it up to you to decide on the types to use for the various fields.

In addition, however, you should impose the following constraints:

- The primary key of the FLIGHTS table is fid.
- The primary keys for the other tables are cid, mid, and did respectively.
- Flights.carrier_id references Carrier.cid
- Flights.month_id references Months.mid
- Flights.day_of_week_id references Weekdays.did

We provide the flights database as a set of plain-text data files in the linked .rar archive. Each file in this archive contains all the rows for the named table, one row per line.

In this homework, you need to do two things: (A) import the flights dataset into SQLite and (B) run SQL queries to answer a set of questions about the data. We describe each of those steps in detail below.

A. Importing the Flights Database [15 points]

To import the flights database into SQLite, first, run `sqlite3` with a new database file (for example, "`sqlite3 hw2.db`"). Next, use the `CREATE TABLE` SQL statement to create the tables, choosing appropriate types for each column and specifying all key constraints as described above:

```
CREATE TABLE table_name ( . . . );
```

Currently, SQLite does not enforce foreign keys by default. To enable foreign keys use the following command. (The command will have no effect if your version of SQLite was not compiled with foreign keys enabled. If so, do not worry about that.)

```
PRAGMA foreign_keys=ON;
```

Finally, use the SQLite `.import` command to read data from each text file into its table:

```
.import filename tablename
```

See examples of `.import` statements in the lecture notes, and also look at the SQLite documentation or `sqlite3`'s `.help` for details.

Put all the SQL code for part A (that is, four `create table` statements and four `.import` statements) into a file called `create-tables.sql`.

B. SQL Queries [35 points]

HINT: *You should be able to answer all the questions below with queries that do not include any subqueries!*

For each question below, write a single SQL query to answer that question. Put your queries in a file called `hw2-queries.sql`. Add a comment to each query indicating (1) the question it answers and (2) the number of rows in the query result.

Important Notes

To receive full credit, make sure your answers obey the following rules:

- The predicates in your queries should correspond to the English descriptions. For example, if a question asks you to find flights by Alaska Airlines Inc., your query should include a predicate that checks for the name (Alaska Airlines Inc.) as opposed to checking for the matching carrier ID. Likewise for predicates over months, weekdays, etc. — use names not IDs.

- If a query uses a `GROUP BY` clause, make sure that all attributes in your `SELECT` clause for that query are either grouping keys or aggregate values. SQLite will let you select other attributes, but such queries are illegal in SQL and other database systems would reject them.

Questions

For each question below, write a SQL query to find the answer. As a hint, after each question, we provided the number of rows that should be in your answer (written between [ ] brackets after the question).

1. Find the distinct flight numbers of all flights from Seattle to Boston by Alaska Airlines Inc. on Mondays. Also notice that, in the database, the city names include the state. So Seattle appears as Seattle WA. [3 rows]

2. Find all flights from Seattle to Boston on July 15th 2015. Search only for itineraries that have one stop. Both legs of the flight must have occurred on the same day and must be with the same carrier. The total flight time (actual_time) of the entire itinerary should be less than 7 hours (but notice that actual_time is in minutes). For each itinerary, the query should return the name of the carrier, the first flight number, the origin and destination of that first flight, the flight time, the second flight number, the origin and destination of the second flight, the second flight time, and finally the total flight time. [488 rows]

   Also, put the first 20 rows of your result right after your query as a comment.

3. Find the day of the week with the longest average arrival delay. Return the name of the day and the average delay. [1 row]

4. Find the names of all airlines that ever flew more than 1000 flights in one day. Return only the names. Do not return any duplicates. [11 rows]

5. Find all airlines that had more than 0.5 percent of their flights out of Seattle be canceled. Return the name of the airline and the percentage of canceled flight out of Seattle. Order the results by the percentage of canceled flights in ascending order. [6 rows]

Be sure to save all files you create during this homework: you will need them in later.