LIS590DCL Theory & Practice of Data Cleaning — Spring 2017
Dr. Bertram Ludäscher — April 3, 2017

Assignment 3 – SQL(ite) and Integrity Constraints

Due: April 14, 2017

Reading Assignments: See Moodle postings.

Problem 1 (SQLite) The goal of this assignment is to acquire some basic hands-on skills for SQLite, using the Software Carpentry lessons at `http://swcarpentry.github.io/sql-novice-survey/`

(a) To get started, visit the SQLite download page and install the binary version of SQLite for your platform.

(b) Read and walk through the 10 Topics of the SQL novice tutorial. Take *notes* as you go through the material; in particular, write down your *questions* if you run into problems or don't understand something. Post questions to the Moodle as needed and report unresolved questions as part of your *narrative* with the homework submission.

(c) As you go through the topics, you will came across a number of questions. Out of these, submit the following as part of your homework submission:

1. **Selecting Data.** Write a query that selects only site names from the `Site` table.
2. **Sorting and Removing Duplicates.** (a) Write a query that selects distinct dates from the `Visited` table. (b) Write a query that displays the full names of the scientists in the `Person` table, ordered by family name.
3. **Filtering.** Normalized salinity readings are supposed to be between 0.0 and 1.0. Write a query that selects all records from `Survey` with salinity values *outside* this range.[1]
4. **Calculating New Values.** After further reading, we realize that Valentina Roerich was reporting salinity as percentages.

   (a) Write a query that returns all of her salinity measurements from the `Survey` table with the values divided by 100.

   (b) Use `UNION` to create a consolidated list of salinity measurements in which Roerich's have been corrected.

   (c) [Extra Credit] Some major site identifiers are two letters long and some are three. The "in string" function `instr(X,Y)` returns the 1-based index of the first occurrence of string Y in string X, or 0 if Y does not exist in X. The substring function `substr(X,I,[L])` returns the substring of X starting at index I, with an optional length L. Use these two functions to produce a list of unique major site identifiers. (For this data, the list should contain only "DR" and "MSK".)
5. **Missing Data.**

   (a) Write a query that sorts the records in `Visited` by date, omitting entries for which the date is not known (i.e., is `null`).

   (b) Some database designers prefer to use a sentinel value) to mark missing data rather than null. For example, they will use the date "0000-00-00" to mark a missing date, or -1.0 to mark a missing salinity or radiation reading (since actual readings cannot be negative). What does this simplify? What burdens or risks does it introduce?

[1] Such "outlier" queries are similar in spirit to our *IC denial* rules in Datalog.

6. **Aggregation.**

   (a) How many temperature readings did Frank Pabodie record, and what was their average value?

   (b) We want to calculate the difference between each individual radiation reading and the average of all the radiation readings. We write the query:

   ```
   SELECT reading - avg(reading) FROM Survey WHERE quant='rad';
   ```

   What does this actually produce, and why? Explain briefly.

   (c) [Extra Credit] Write a correct SQLite query that gets the job done!

   ```
   William Dyer, Frank Pabodie, Anderson Lake, Valentina Roerich, Frank Danforth
   ```

   Find a way to order the list by surname.

7. **Combining Data.**

   (a) Write a query that lists all radiation readings from the DR-1 site.

   (b) Write a query that lists all sites visited by people named "Frank".

   (c) Write a query that shows each site with exact location (`lat`, `long`) ordered by visited date, followed by personal name and family name of the person who visited the site and the type of measurement taken and its reading. Please avoid all `null` values. Tip: you should get 15 records with 8 fields.

Problem 2 (Integrity Constraints in SQL)

(a) Implement the ICs from Problem 2 on Assignment 2 in SQLite as "denial" queries (or *views*), analogous to the `icv(...)` rules in Datalog.

(b) Are there additional ways to check (and enforce) integrity in SQL(ite)? Research and explain briefly.

What to submit. For parts of this assignment an autograder tool will be offered that allows you to preview the autograder assessment. Details of the required submission format will be posted on Moodle.