Lab 3-1: Potpourri Part 1

Due date: Friday, October 15, 11:59pm

Note: Lab 3-2 will be assigned on Friday, October 15, in class. It will be due October 18.

Lab Assignment

Assignment Preparation

This is an individual lab. Each student has to complete all work required in the lab, and submit all required materials **exactly as specified** in this assignment.

Note on data. This lab will require you to use the files you prepared in **Lab 2**. You will complete four assignments, using four of the nine course databases.

Fixing the Databases

You will receive your **Lab 2** grades (for the portion of the lab covering `CREATE TABLE` statements). The graded `<DATABASE>-setup.sql` scripts are given to you in hardcopy, with any errors clearly marked.

You need to do the following:

1. fix all errors, and finalize the content of `<DATABASE>-setup.sql` files;
2. ensure that all insertions proceed correctly with the new `setup` scripts, fix any issues;
3. submit fixed script collections for ALL nine datasets (including the ones, for which there are no data manipulation assignments in this lab).

Submission filenames. You will submit the following files for each dataset:

- `<DATABASE>-setup.sql`: your `CREATE TABLE` statements.
- `<DATABASE>-build-<file>.sql`: one script per table that inserts all tuples in the table. Each tuple must be inserted using a separate INSERT INTO statement.
- `<DATABASE>-cleanup.sql`: the `DROP TABLE` script.
- `<DATABASE>-test.sql`: the database test script.

In addition, for the datasets mentioned in this part of the assignment, you will submit a `<DATABASE>-modify.sql` script which performs all the required tasks.

Tasks

The assignments in this part are specific to individual databases you created in **Lab 2**. Please execute them only on the specified datasets. The assignments ask you to change both the schemas and the instances of the databases.

[**STUDENTS dataset.**] Create an SQL script `STUDENTS-modify.sql` which performs the actions below.

Extend the database structure to include the information about the GPA for each student.

Update the database as follows:

- Keep in the database only the students from grades 3 and 4.
- Set the GPA of fourth graders from room 110 to 3.25.
- Set the GPA of fourth graders from other rooms to 2.9.
- Set the GPA of all third graders to 3.5.
- The following instructions apply to individual students and override all prior GPA assignments.
  - Set the GPA of `ROBBY PINNELL` to 4.0.
  - Set the GPA of `TOBIE SAADE` to 3.7
  - Set the GPA of `CYRUS RODEY` to be 0.3 higher than whatever it currently is.
  - Set the GPAs of `CRYSTA GROENEWEG` and `ADRIAN LEAPER` to be 25% higher than their current GPAs.

Include all necessary SQL commands to achieve this result into the `STUDENTS-modify.sql` script. Complete the script with the

```
SELECT * FROM <students-table>
ORDER BY <GPA-column>, <grade-column>, <student-lastname-column>;
```

query, replacing `<students-table>` with the name of your table containing the list of students and `<GPA-column>`, `<grade-column>` and `<student-lastname-column>` with the names of the columns storing the GPA, the grade level of each student and their last names respectively.

[**WINE dataset.**] Create an SQL script `WINE-modify.sql` which performs the actions below.

1. Remove the columns storing the appelation name and the name of the wine from the table storing the list of wines (we refer to this table as "the wine table")[1].
2. Keep in the wine table only the Zinfandels with a score of 93 or higher.
3. Modify the length of the attribute storing the winery name to be 18 characters long[2].
4. Output the list of wines using the following SQL query:

   ```
   SELECT * FROM <wine-table>
   ORDER BY <wine-id-column>;
   ```

   (replace `<wine-table>` and `<wine-id-column>` with appropriate, in your database, names).
5. Let's compute the expected revenue from each of the wines we have left. Create a new column called `Revenue` in the wines table.

   Compute the expected revenue under the assumption that each bottle of wine will sell at the declared price. The wines table have an attribute specifying how many cases of each wine has been produced. Each case of wine consists of exactly 12 bottle.

   Update the value of the `Revenue` attribute using the information provided to you.
6. Keep in the wines table only the wines with the expected revenue exceeding $150,000.
7. Output the list of wines using the following SQL query:

   ```
   SELECT * FROM <wine-table>
   ORDER BY <wine-id-column>;
   ```

   (replace `<wine-table>` and `<wine-id-column>` with appropriate, in your database, names).

[1] This is largely for the eventual final result/output to be compact.

[2] If you did everything right, all winery names in the remaining tuples will be shorter than 18 characters.

[CARS dataset.] Create a SQL script `CARS-modify.sql` which performs the following actions.

1. Keep in the table storing the technical characteristics about the cars (we refer to this table as "the car data table"), ONLY the records that satisfy *at least one* of the following conditions:
   (a) vehicles made in 1979 and after with accelerations between 13 and 14 (inclusive).
   (b) vehicles that have MPG of 26 miles per gallon or better what have an engine with more than 110 horsepowers.
2. Run the following SQL query:

   ```
   SELECT *
   FROM <car-data-table>
   ORDER BY <year-column>, <car-Id>;
   ```

   where `<car-data-table>` is the name of the car data table in your `CARS` database and `<year-column>` is the column in that table storing the year in which a vehicle was made and `<car-id>` is the unique Id of each tuple in the car data table.
3. Remove from the car data table all attributes except car id, car year, acceleration, MPG number of cylinders, and horsepower.
4. Remove from the car data table information about all cars with fewer than 5 cylinders.
5. Run the

   ```
   SELECT *
   FROM <car-data-table>
   ORDER BY <year-column>, <car-Id>;
   ```

   query again.

Submission Instructions

Please, follow these instructions exactly. Up to 10% of the Lab 3 grade will be assigned for conformance to the assignment specifications, **including the submission instructions**.

Please, **name your files exactly as requested** (including capitalization), and submit all files **in a single archive**. Correct submission simplifies grading, and ensures its correctness.

Please include your name and Cal Poly email address in all files you are submitting. If you are submitting code/scripts, include, at the beginning of the file a few comment lines with this information. Files that cannot be authentificated by observing their content will result in penalties assessed for your work.

Specific Instructions

You must submit all your files in a single archive. Accepted formats are `gzipped tar` (`.tar.gz`) or `zip` (`.zip`).

The file you are submitting must be named `lab3.zip` or `lab3.tar.gz`.

Inside it, the archive shall contain nine directories named `AIRLINES`, `BAKERY`, `CARS`, `CSU`, `INN`, `KATZENJAMMER`, `MARATHON`, `STUDENTS` and `WINE` In addition, the root of the directory must contain a `README` file, which should, at a minimum, contain your name, Cal Poly email, and any specific comments concerning your submission.

Each directory shall contain all SQL scripts built by you for the specific dataset in response to all parts of the lab. The Lab 2 scripts must be resubmitted, with the correct names. (these are the `<Dataset>-setup.sql`, `<Dataset>-build-<table>.sql` and `<Dataset>-cleanup.sql` files).

Your `CARS`, `STUDENTS` and `WINE` directories also shall contain the `CARS-modify.sql`, `STUDENTS-modify.sql` and `WINE-modify.sql` scripts.

Submit your archive using the following `handin` command:

```
handin dekhtyar lab03-1 <file>
```

Testing

Your submission will be tested by running all scripts you supply and checking the produced output for correctness. I may also use some extra scripts to verify the correctness of the databases you have constructed.

If you are aware of any bugs, or incorrect behavior of your SQL scripts, I strongly suggest that you mention it in the README file.