Lab 3-1: Potpourri Part 1

Due date: Friday, April 21, 11:59pm

Note: **Lab 3-2** will be assigned on Friday, April 21, in class.

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

Those of you who have submitted `CREATE TABLE` statements for Lab 2 ahead of time, are expected to fix the structure of their databases by full Lab 2 submission deadline. If you have done so, skip this section.

If you did not submit your `CREATE TABLE` statements on time, but did submit full Lab 2, we will give you feedback on your `CREATE TABLE` statements on Friday, April 21. You shall make appropriate changes to your `CREATE TABLE` statements. Read the instructions below.

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).

Dataset Submission.

For this part of the lab, you will submit three datasets: STUDENTS, WINE, and CARS.

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>-insert.sql`: basically a script that, when run inserts ALL tuples into all tables of the database. (A script that consists of `source <DATABASE>-build-<file>.sql` commands usually works. So does a script that is the results of concatenating all `<DATABASE>-build-<file>.sql` files in the correct order).
- `<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.

Traceability

Your `<DATABASE-modify.sql>` scripts will consist of a number of SQL DDL and DML commands. Each script must have a top comment specifying your full name and cal poly email (login id).

Additionally, each SQL statement you place into the file must be prefaced with a short comment specifying its purpose. If you are skipping a command (e.g., because you were not able to make it work), place the comment specifying that you skipped a SQL statement into your script. For example, if you are asked to write a SQL command that adds a new attribute `foo` to table X, deletes a few records from this table, and then instantiates it `foo` to 10 for all remaining tuples, your script shall have comments similar to the ones shown below (assuming you skip the second command):

...

```
-- Add attribute foo to table X

ALTER TABLE ....
...
;

-- Delete tuples [Not Implemented]

-- Set value of foo to 10

UPDATE ....
...
;
```

Tasks

The assignments in this part are specific to individual databases you created in **Lab 2**. Please execute them only on the specified databases. 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 5 and 6.
- Add a new classroom to the database. The classroom number is 120, and the teacher in that classroom is `GAWAIN AP-MAWR`.
- Move `JEFFREY FLACHS`, `TAWANNA HUANG` and `EMILE GRUNIN` to classroom 120.
- Set the GPA of sixth graders to 3.25.
- Set the GPA of fifth graders from room 109 to 2.9.
- Set the GPA of fifth graders from room 120 to 3.5.
- The following instructions apply to individual students and override all prior GPA assignments.
  - Set the GPA of `CHET MACIAG` to 4.0.
  - Set the GPA of `AL GERSTEIN` to be 0.3 higher than whatever it currently is.
  - Set the GPAs of `TAWANNA HUANG` and `ELVIRA JAGNEUX` to be 10% higher than their current GPAs.

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

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

and

```
SELECT *
FROM <teachers-table>;
```

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, and replacing `<teachers-table>` with the name of your table containing the list of teachers.

[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 Syrahs with a score of 93 or higher.
3. Modify the length of the attribute storing the winery name to be 15 characters long[2].
4. Add a new column to the table called `Revenue`. It should have the same type as your price column.
5. A case is 12 bottles of wine. Using the information available to you, set the revenue for each wine left in the table to be equal to the total amount of money that can be made by selling all the cases of the wine.
6. Output the list of wines using the following SQL query:

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

   (replace `<wine-table>` with the appropriate table name).

[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 15 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 or 1980 with MPG of 20 or above.
   (b) vehicles that have MPG of 26 miles per gallon or better what have an engine with more than 110 horsepowers.
   (c) vehicles that have 8 cylinders and accelerate to 60 mph in less than 10 seconds.
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.
4. Remove from the car data table information about all cars with 5 cylinders or fewer.
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 three directories named `CARS`, `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).

Additionally, the submitted directories shall contain the `<Dataset>-modify.sql` scripts: `CARS-modify.sql`, `STUDENTS-modify.sql` and `WINE-modify.sql`.

Submit your archive using the following `handin` command:

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

Testing

The graders will make the `lab3try` script available to you. Use it to check your submission.

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.