Assignment 2

In this assignment you exercise SQL and data analytics. The assignment consists of 2 parts.

The entire assignment can be performed with SQLite DBMS – this is a good choice for relatively small datasets and for analytics, because of its portability and ease of installation. SQLite has all the features you need to perform this assignment's tasks.

You can use views, subqueries and any other tools that you need to perform the tasks. The only requirement is to produce correct answers.

Part I. SQL and battleships.

This part introduces an example concerning World War II capital ships. It involves the following relations:

Classes (class, type, country, numguns, bore, displacement)

Ships (name, class, launched) --launched is for year launched

Battles (name, date_fought)

Outcomes (ship, battle, result)

Ships are built in "classes" from the same design, and the class is usually named for the first ship of that class. The relation Classes records the name of the class, the type (*bb* for battleship or *bc* for battlecruiser), the country that built the ship, the number of main guns, the bore (diameter of the gun barrel, in inches) of the main guns, and the displacement (weight, in tons). Relation Ships records the name of the ship, the name of its class, and the year in which the ship was launched. Relation Battles gives the name and date of battles involving these ships, and relation Outcomes gives the result (sunk, damaged, or ok) for each ship in each battle.

1.1. Create the battleship database in SQLite.

Create simple SQL statements to create the above relations (no constraints for initial creations).
Insert data into each table using data provided in the csv format in *battleship_data.zip*.
You can easily convert these comma-separated lines into SQL insert statements by using multiple available tools, for example: http://www.convertcsv.com/csv-to-sql.htm.

1.2. SQL queries

Write SQL queries for the following requirements:

1. The treaty of Washington in 1921 prohibited capital ships heavier than 35,000 tons. List the ships that violated the treaty of Washington.
2. List the name, displacement, and number of guns of the ships engaged in the battle of Guadalcanal.
3. List all the capital ships mentioned in the database. (Remember that not all ships appear in the Ships relation.)

4. Find those countries that had both battleships and battlecruisers.
5. Find those ships that "lived to fight another day"; they were damaged in one battle, but later fought in another.
6. Find the countries whose ships had the largest number of guns.
7. Find the names of the ships whose number of guns was the largest for those ships of the same bore.
8. Find for each class with at least three ships the number of ships of that class sunk in battle.

After successfully running each query, store it in a separate sql file: 1.2.1.sql, 1.2.2.sql etc.

1.3. Database modifications with SQL

Write the following modifications.

1. Two of the three battleships of the Italian Vittorio Veneto class – Vittorio Veneto and Italia – were launched in 1940; the third ship of that class, Roma, was launched in 1942. Each had 15-inch guns and a displacement of 41,000 tons. Insert these facts into the database.
2. Delete all classes with fewer than three ships.
3. Modify the Classes relation so that gun bores are measured in centimeters (one inch = 2.5 cm) and displacements are measured in metric tons (one metric ton = 1.1 ton).

Again, store each sql statement in a separate file: 1.3.1.sql, 1.3.2.sql and 1.3.3.sql.

1.4. Adding constraints to existing database

Add the following constraints.

[Some constraints might not be possible to add right away. In such cases, delete first the violating tuples. The deletion should be performed with a general SQL statement, not manually.]

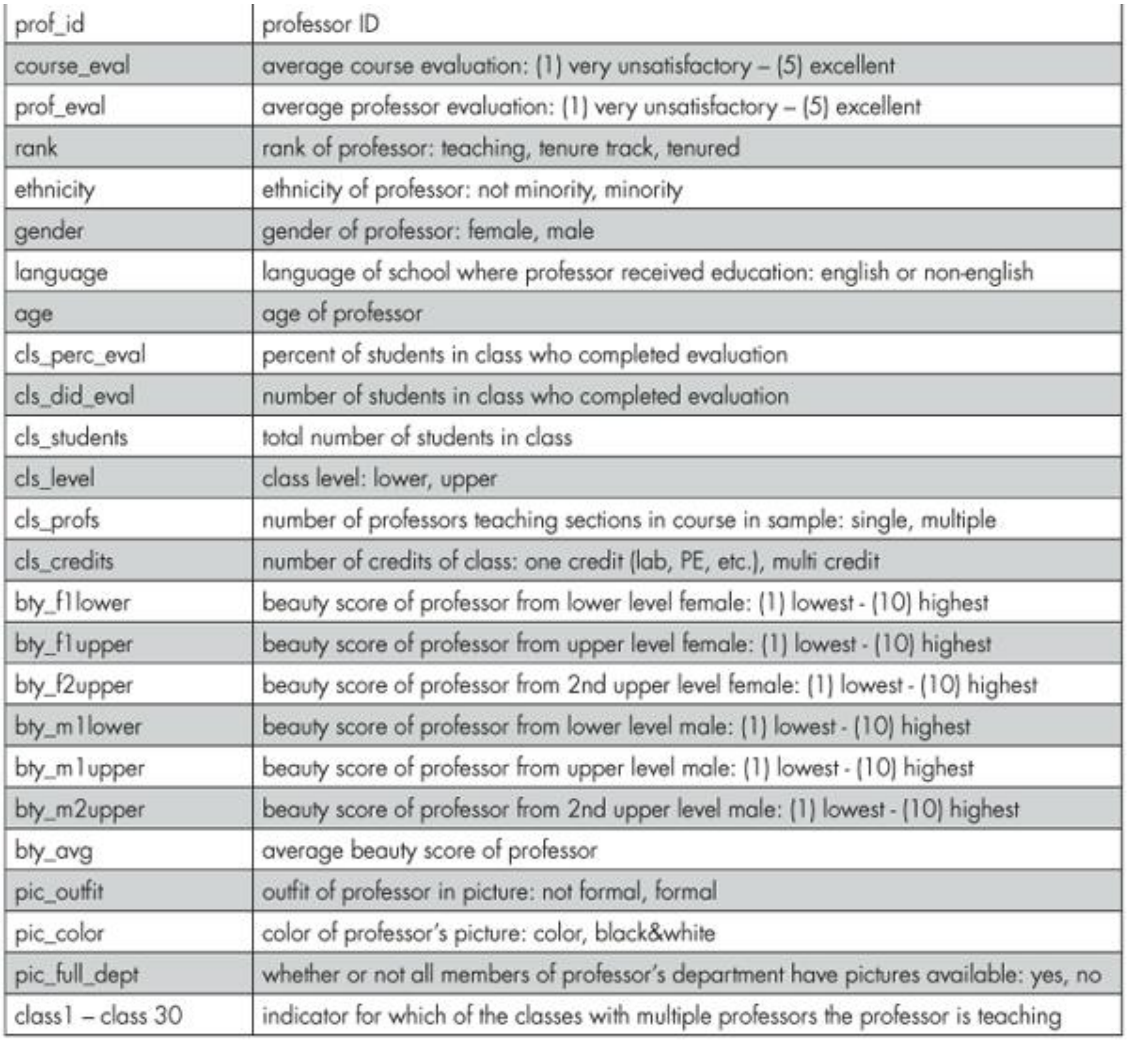
1. Every class mentioned in Ships must be mentioned in Classes.
2. Every battle mentioned in Outcomes must be mentioned in Battles.
3. Every ship mentioned in Outcomes must be mentioned in Ships.
4. No class of ships may have guns with larger than 16-inch bore.
5. If a class of ships has more than 9 guns, then their bore must be no larger than 14 inches.

Again, store solution to each separate problem (including deletion statements) in a separate file: 1.4.1.sql, 1.4.2.sql etc.

Part 2. Data analytics with SQL and Excel

In this part you are provided with an anonymous dataset of instructor evaluations collected at Columbia University and publicly released for the following paper: http://chance.amstat.org/2013/04/looking-good/.

The dataset consists of a single table ***evaluations*** with attributes described in Figure 1. The data is provided in file *StudentEvaluations.csv.*



prof_id	professor ID
course_eval	average course evaluation: (1) very unsatisfactory – (5) excellent
prof_eval	average professor evaluation: (1) very unsatisfactory – (5) excellent
rank	rank of professor: teaching, tenure track, tenured
ethnicity	ethnicity of professor: not minority, minority
gender	gender of professor: female, male
language	language of school where professor received education: english or non-english
age	age of professor
cls_perc_eval	percent of students in class who completed evaluation
cls_did_eval	number of students in class who completed evaluation
cls_students	total number of students in class
cls_level	class level: lower, upper
cls_profs	number of professors teaching sections in course in sample: single, multiple
cls_credits	number of credits of class: one credit (lab, PE, etc.), multi credit
bty_f1lower	beauty score of professor from lower level female: (1) lowest - (10) highest
bty_f1upper	beauty score of professor from upper level female: (1) lowest - (10) highest
bty_f2upper	beauty score of professor from 2nd upper level female: (1) lowest - (10) highest
bty_m1lower	beauty score of professor from lower level male: (1) lowest - (10) highest
bty_m1upper	beauty score of professor from upper level male: (1) lowest - (10) highest
bty_m2upper	beauty score of professor from 2nd upper level male: (1) lowest - (10) highest
bty_avg	average beauty score of professor
pic_outfit	outfit of professor in picture: not formal, formal
pic_color	color of professor's picture: color, black&white
pic_full_dept	whether or not all members of professor's department have pictures available: yes, no
class1 – class 30	indicator for which of the classes with multiple professors the professor is teaching

Figure 1. Attributes in the original relation: evaluations

You would need to create a single database table to store this data in SQLite database, preserving data domains described in Figure 1. Now you can fill it with data by converting comma-separated lines into INSERT statements by using any automated tool, such as http://www.convertcsv.com/csv-to-sql.htm.

Store all your SQL statements for table creation and data population in file evaluations_db.sql.

Using the *evaluations_db* answer the following questions with SQL queries. Collect each answer into a csv file, open it in Excel, and visualize the results of queries 1, 2, 3, 4, and 5 as charts.

Submit each query in a separate sql file (2.1.sql, 2.2.sql etc.), and submit your results and visualizations in files *2.1.xlsx, 2.2.xlsx etc.*, or convert your visualizations into *2.1.pdf, 2.2.pdf* if you use a different spreadsheet software.

Questions:

1. How many courses have a high evaluation score (>=3.5) for each course level?
2. How many high instructor evaluation scores for each instructor gender?
3. How many courses have score above average in their course level group?
4. What is the mode (the most frequent value) of the beauty score for the following 4 groups:
    a. Beauty score given by all female students to female instructors?
    b. Beauty score given by all male students to male instructors?
    c. Beauty score given by all female students to male instructors?
    d. Beauty score given by all male students to female instructors?

    Use values in columns bty_f1lower, bty_f1upper, bty_f2upper, bty_m1lower, bty_m1upper, and bty_m2upper, do not use an average beauty score.

    Plot the resulting values with the corresponding legend **in a single graph**.

5. What is an average instructor evaluation score for each of professor ranks: teaching, tenure track, tenured?
6. Who are the top 5 instructors (IDs) according to an average instructor evaluation score?
7. Who are the top 5 instructors by an average beauty score (from all students)? Are these the same as in question 6?
8. Find 15 nearest neighbors for an instructor with the following demographic characteristics: language='English', age=45, gender='female', bty_avg=6, ethnicity='not minority'. Based on the average instructor evaluation score for these closest neighbors, predict an average score for this instructor.

Finally, provide file *report.pdf* where you briefly describe what you have learned from answers for each of these questions.

Marking scheme

Part 1. (28 points)

1.1. --- 2 points
1.2. ---16 points (each sql query – 2 points)
1.3. ---4 points
1.4. ---6 points

Part 2 (37 points)

Creating and populating table with data:
--- 2 points

Queries:
2.1. ---2 points
2.2. ---2 points
2.3. ---3 points
2.4. ---3 points
2.5. ---2 points
2.6. ---2 points
2.7. ---3 points
2.8. ---4 points

Visualizations:
---2 points each for a total of 10 points

Report:
---4 points

Total: 37+28=65 points for 15% of the course mark