Project #3: Working with Multiple Tables

1 Tasks

1.1 Write SQL (.6 points each = 7.8 points)

Write SQL to accomplish the following tasks using the World sample database provided with MySQL. Express the full problem request. Before each SQL command, provide a comment giving the question number and how many rows resulted from the query. Unless specifically directed to do so, never look up IDs and use them yourself; let SQL do the work. Databases are fluid; records are added and edited all the time. Do not to encode your assumptions about current data in a database!

1. You're interested in finding the countries in which languages are spoken by over 40% of the population. Find these and list the country, language, and percentage. The results should be sorted by country. Use table names only in the FROM clause and then only when you must.
2. Same as previous, but use Implicit, rather than Explicit, syntax
3. You speak French and want to know all the places you can visit where that language is spoken. Get a list of cities and associated countries where you'll be comfortable. Results should have column headers "City" and "Country", respectively. Sort these by Country and, within Country, City.
4. Get a list of cities, languages spoken in that country in which that city exists, and populations. Requirements: (1) do not refer to the country table in this query, and (2) make use of the USING clause.
5. Same as previous, but instead of USING do a natural join.
6. Generate a list of ALL cities along with the languages associated with their countries. Cities should be listed regardless of whether there is an associated language. Do not refer to the country table in this query. Note that existing data may not fully test the result as there may be no unassociated records.
7. Same as previous, but this time list all LANGUAGES and their associated cities. Do not change the order in which tables appear in the FROM clause.
8. Get a list of country names (with column heading CountryName) along with whether they are considered Small, Medium, or Large. Small countries are those with a population under one million. Medium countries are those at or above one million but under three million. Large countries are those at or above three million. The resulting list should be sorted by country name in alphabetical order. Hint: you'll need a *union* (see the textbook's example) for this one.

These next questions use the **Ex** database, focusing on the Employees table:

9. Employees at the company each have a Manager. The manager's manager is their Executive. Use the self-join example in the slides, get a list of the employees, their Manager, and their Executive. Show all names in a single column in the form "John Doe". Label columns "Employee", "Manager", and "Executive", respectively. Sort by the employees last name, then, within that, first name.

10. You're running a new, small company. During the upcoming month, each employee is expected to briefly interview every other employee to get acquainted. Write a query that generates a complete list of interviewers and interviewees, with columns titled "Interviewer" and "Interviewee," respectively. Make sure employees aren't listed as meeting *with themselves*. Note that each pair will meet twice, however, with roles reversed. Sort by last name, then first name, of the interviewer.

For these questions use the **Sakila** database provided as a sample with MySQL:

11. Get a list of the name (first and last), city, state, and zip of all USA customers.
12. Actors with last name of Hoffman are some of your favorite actors. Get a list of all the films they've been in, along with the length of the film. The list shouldn't contain any duplicates (i.e., if more than one Hoffman is in a film, the film should only be listed once).
13. Get a list of the titles of all the English language comedies in the database.

1.2 Design Tables, Fields, Properties (2.2 points)

You want to expand the World database to track country alliances. Your research indicates that, surprisingly, countries can participate in more than one alliance. Each alliance has a name and a year in which it was created. You also want to track the year in which each country joined an alliance.

Come up with design for table(s) necessary to support this change. For each field identify a name, a data type, size, and whether the field must be non-null (marked as "NN"). Identify primary and foreign keys (with "PK" or "FK"). Show your results as a multiline (aka block) SQL comment.

2 Check Your Work

Check your work in MySQL and ensure it meets the requirements. Remember to think and double-check; just because the SQL runs and produces output doesn't mean it's the *right* output.

3 Submitting Your Work

Put all work into a single .SQL file and submit the file via Canvas. Place comment headers over each comment, indicating what question it is answering.

4 Hints

- Save your work. When you exit MySQL Workbench some queries may disappear; you don't want this to happen to you. Savvy technical folks save their work often, and in versions.
- For the Design portion of the assignment, use the existing tables wisely; they will give you a lot of good information about what goes here.

5 Grading

Projects are worth ten points. You'll be graded on form and correctness, i.e., does your work produce the correct output and does it produce results in a straightforward and efficient way?