DSCI 551 – Spring 2026
Homework 3: Data Modeling & SQL

(100 points)

Due: 11:59pm, March 6, 2026, Friday

NO Late Submissions will be accepted!

This homework is based on two MySQL sample databases.

- world: https://dev.mysql.com/doc/world-setup/en/
- sakila: https://dev.mysql.com/doc/sakila/en/

1. [20 points] Data Modeling (world):

Consider a modified world data model with the following tables.

- country(code, name, continent, gnp)
- city(id, name, countrycode, population)
- countrylanguage(countrycode, language_id, isofficial, percentage)
- language(id, name)

Note: language id refers to the ISO code of the language, e.g., "en" for "English" (language name) and "fr" for French. Note also that city id and population are integers. Note also that key attributes are underlined, and countrycode in the city table can not take null value.

a. Write "create table" statement to create the above tables. Properly define keys and foreign keys in the tables.
b. Reverse-engineer the tables into a corresponding ER model such that the conversion of the model using the procedure described in class will produce the above tables. Be sure to indicate the multiplicity of relationships (including the directions, e.g., which entity set is on the one-side or many-side). For many-to-one or one-to-one relationships, indicate if it is at most one or exactly one. Note: You do not need to draw the model, instead state the entities, relationships, attributes, and keys in the model clearly.

2. [80 points] Parent–Child Schema Analysis (Sakila)

The following two tables store the foreign key relationships among tables in the Sakila database and columns of the tables. The data of the tables are extracted from the information_schema of MySQL.

You should only use these two tables in answering the following questions.

- child_parent(child_table, child_col, parent_table, parent_col)
- tbl_columns(table_name, column_name, data_type)

Note that as discussed in class, if a table S has a foreign key referencing a table R, we call S a child table and R parent table.

You are provided with an SQL script, hw3.sql, which creates and populates the above tables in a "dsci551" database (it will create the database if it does not exist yet).

Rules

- Use only child_parent and tbl_columns tables.
- Do NOT query information_schema directly.
- No recursive queries.
- Use proper JOINs.
- Use CTEs where requested.

Questions

For each of the following questions, write an SQL query to answer the question (question 9 also requires you to create a view first). Note that data provided to you are the **sample** data produced from Sakila. Your queries should also work on the data extracted from other databases.

1. Parent Reference Count (GROUP BY + ORDER BY)
   For each parent table, count how many child references it has. Return parent_table and reference_count sorted from highest to lowest.

2. Child Dependency Count (GROUP BY + HAVING + ORDER BY)
   For each child table, count how many parent tables it depends on. Return only child tables that depend on more than 1 parent table, sorted by the count from highest to lowest.

3. Tables Acting as Both Parent and Child (JOIN; INTERSECT)
   Find tables that appear as both parent_table and child_table. Sort the table names in the ascending order. (Note you need to provide two solutions to this question, one using JOIN, the other INTERSECT)

4. Tables With No Relationships (LEFT JOIN; EXCEPT)
   Using tbl_columns to get the full table list, find tables that do not appear as a child and do not appear as a parent. (Note you need to provide two solutions to this question, one using JOIN, the other EXCEPT)

5. Multi-Dependence Detection (GROUP BY + HAVING)
   Identify child-parent pairs where a child references the same parent using more

than one column. Note the references may be made from a single or multiple foreign keys. Return child_table, parent_table, and number of columns used.

6. Foreign Key Data Type Comparison (JOIN)
   Join child_parent and tbl_columns to display child table, child column, child data type, parent table, parent column, and parent data type.

7. Most Common Foreign Key Data Types (CTE + GROUP BY)
   Using a CTE, extract all foreign key columns with their data types, count occurrences per data type, and return the most common one(s). Note that there may be multiple such data types.

8. Connectivity Score (CTE + JOIN + ORDER BY)
   Connectivity Score = (# times table appears as parent) + (# times table appears as child). Compute and rank all tables from highest to lowest connectivity score.

9. Tables With Above-Average Connectivity (View + Subquery)
   Create a view "tbl_score" to store the results of the previous question. Using the view, find tables whose connectivity score is greater than the average connectivity score of all tables. Return only the table names.

10. Find tables that have at least one parent or child relationship and have more than 5 columns (Subquery + Group by + Having + Order by).

    Return table name and number of columns, sorted by column count descending.

Objective

This assignment evaluates your ability to analyze relational structure, use JOINs, GROUP BY, HAVING, ORDER BY, CTEs, subqueries, and aggregation to understand schema relationships in Sakila.

Submission:

- Submit .ipynb file with your answer in clearly labelled markdown cells. For SQL query, also include the query result in a cell right after the query. You can run the queries in MySQL workbench or terminal, and copy and paste the results into the cells.
- Prepend your full name to the submission file name, e.g., JaneDoe_hw3.ipynb.