CSC2141 Assignment Part 2

Designing and Building Your Database

Due date: Friday March 8, 11:59PM

Overview

In the first part of the course assignment, you assembled a dataset comprising at least four tables that will be used for design and query purposes. In **Part 2** you will design and create your database, and test with basic database queries. Part 2 is scored out of 25 and worth 10% of your final grade. Please refer to the main assignment document for general rules and guidelines.

The assignment components are as follows:

1. **Adjust** your dataset if necessary (add/remove rows or columns).
2. **Design** your database such that it is in third normal form (10 points).
3. **Create your database tables** and **populate them with your data** (10 points).
4. **Test your database** with a small number of queries (5 points).

Note that in this assignment you will submit some SQL statements but you will not submit your data. Data submission will be necessary in Part 3 of the assignment.

Section 0: Adjust Your Dataset (if Necessary)

Based either on feedback from Part 1 of the assignment, or your own desire to simplify your dataset or make it more interesting, you can adjust the number of attributes or rows. Please provide a **short update** indicating any changes you might have made to your intended structure, or indicate that nothing has changed. (0 points)

Section 1: Design Your Database in 3NF

Now that you have identified tables and attributes, you should examine the dependencies among them and perform the necessary steps of normalization. As you have seen in class, normalization can lead to the generation of many more tables; if the number of tables becomes excessive you can remove some of them from the assignment if you wish to do so. For example, if normalization turns your four tables into twelve, you can remove some of these tables and their corresponding data from the assignment. If you do this, please indicate the changes in Section 0.

As a reminder, at least two of your tables must have at least 100 rows; others can have fewer. Your tables should all have more than two attributes.

Requirements:

For this section you need to submit two pieces of information:

(1) An **internal schema** showing the structure of your normalized database. Connections between tables enforced by foreign keys should be shown. Provide a one-paragraph explanation of the schema. (5 points)

(2) **One Dependency diagram per table**, consistent with the examples shown in the Week 6 slides, showing the data dependencies in your tables. Provide a one-paragraph explanation of the dependencies in your tables. (5 points)

Your schema and diagrams must be clearly and professionally presented (ideally computer generated).

Section 2: Create Your Tables

With your database normalized (and possibly streamlined for simplicity's sake), you're ready to construct your CREATE TABLE statements and load your data.

Each CREATE TABLE statement should specify appropriate datatypes, primary keys, foreign keys, and constraints. An example of an appropriate constraint would be limiting the value of a given attribute to a specific set of text strings, or constraining it to be no greater than a given value.

You can load the data into the database however you like; LOAD DATA and INSERT INTO statements are good options.

You must include comments in your SQL script explaining what each statement does!

Requirements:

(1) CREATE TABLE statements. (8 points)

(2) One example of a LOAD DATA or INSERT INTO statement for each of your tables, and the total rows and columns created. (2 points)

Section 3: Run Simple Queries

To verify that your data have loaded correctly, construct three queries with the following components at a minimum:
Query 1. SELECT from a single table with a WHERE clause, producing a derived attribute.

Query 2. A NATURAL, INNER, or OUTER JOIN between two of your tables.

Query 3. A query covering one or more tables that uses a GROUP BY statement on at least one of your variables.

Requirements:

(1) SELECT statements, images showing the first five rows (or the total number of rows, if fewer than five), and the total number of rows and columns returned from each query. (5 points)

Submitting Your Assignment

Please provide your submission as **two files:**

(1) A formatted PDF with the filename 'B########_Assignment_Part2.pdf', where 'B########' is your Banner ID.

(2) A single text file containing your SQL statements for Sections 2 and 3, with the filename 'B########_Assignment_Part2.sql'.

Note that the assignment submission portal will only accept files with .pdf or .sql extensions.

Upload your assignment through the Brightspace portal.

Please consult the section on academic integrity in the assignment description document.

Rubric

	Excellent (100%)	Very Good (80%)	Acceptable (60%)	Borderline (40%)	Unacceptable (0% - 20%)	Notes
DESIGN: Internal schema	Diagrams are clear and complete, with appropriate levels of detail. Foreign keys and relationships accurate and clearly shown. Schema explanation is clear and consistent.	Diagrams are clear and complete, with appropriate levels of detail. Foreign keys and relationships accurate. Schema explanation is clear and largely consistent with the table structure.	Diagrams are largely complete, possibly with minor omissions in keys / datatypes. Foreign keys and relationships are mostly accurate. Schema explanation is mostly accurate.	Diagrams are incomplete, with missing details. Foreign keys and relationships have major inaccuracies. Schema explanation is a poor match to the diagram.	Internal model representation is inaccurate with large omissions. Important details about relationships are missing. Explanation is poorly written or absent.	
DESIGN: Dependency diagrams	Dependency diagrams contain all necessary elements: primary keys clearly identified, all dependencies indicated with arrows. Complete consistency with internal schema.	Dependency diagrams contain all necessary elements: primary keys clearly identified, nearly all dependencies indicated with arrows. Largely consistent with internal schema.	Dependency diagrams contain most necessary elements although some keys, attributes, or dependencies may be missing. Some inconsistencies with internal schema.	Dependency diagrams lacking important details and/or contain significant inaccuracies. Possible inconsistencies with internal schema.	Dependency diagrams absent or inaccurate. No relationship between diagram and internal schema.	
CREATE	CREATE TABLE statements are perfectly aligned with Section 1 and execute correctly. Statements are well commented. Loading / insertion commands are correct.	CREATE TABLE statements are aligned with Section 1 and execute correctly. Statements include some level of commenting. Loading / insertion commands are correct.	CREATE TABLE statements are slightly unclear, have some disparity with Section 1, or are poorly commented. Loading / insertion commands are largely correct.	Significant problems with CREATE TABLE statements: syntax errors, incorrect definitions, poorly structured, or poorly commented. Errors in loading / insertion commands.	CREATE TABLE statements do not execute and/or highly inconsistent with Section 1. Loading / insertion commands inaccurate.	
RUN	Queries conform exactly to guidelines and have valid syntax. Queries make logical sense. Results are shown clearly in tables (screen capture or other).	Queries conform to guidelines and have valid syntax. Queries largely make logical sense. Results are shown in tables (screen capture or other).	Queries largely conform to requirements, with possible syntax or logic errors. Tables are mostly accurate and clear.	Some queries fail to conform to requirements or execute correctly. Disconnection between query logic and implementation. Tables have significant issues of clarity or accuracy.	Most queries are structurally and/or logically invalid. Tables are missing or lack connection with the queries / database.	