

# Project

## Database Systems (INFR 3810)

### Part 1 (Entity-Relationship E/R Design)

Think of a non-trivial, realistic and interesting scenario where a certain amount of data is to be stored, updated, and queried. You are to model such a database using an Entity-Relationship (E/R) diagram. The database should be of sufficient complexity such that the E/R diagram has *at least twelve* entity sets and relationships (e.g., 5 entity sets and 8 relationships among these entity sets). There should be attribute(s) associated with the entity sets and probably some of the relationships.

### Part 2 (Conversion to a Relational Schema)

Design a good relational schema for your database, by converting the E/R diagram in Part 1 to a relational schema. Follow these steps:

1. List the relations converted from the entity sets and the relationships in the E/R diagram. Specify all the keys that hold each relation. At this stage, you do not yet need to specify the attribute types.
2. State all the completely non-trivial functional dependencies that hold for each relation. Do not include the ones that follow directly from the definition of keys.
3. Check to see if all the relations are in Boyce-Codd Normal Form (BCNF). If not, redesign the relations and start over.

Document these five steps in a text file for submission.

### Part 3 (Loading Data into SQLite)

It is recommended that you test the correctness of the commands in an interactive SQLite session, before committing them to .sql batch files. It is important to save all data in files, in case you lose data in the database and need to start over again. Before you reload the database, you need to delete the contents of each table or the same data will be appended to the table and will double the table size.

The command for deleting a table T entirely is:

> drop table T;

The command for removing all the contents of a table, without deleting the table itself is:

> delete from T;

Create the tables for your database and put the commands in the file *create\_tables.sql*.

Load sample data into the tables in your database and put this in the file *populate\_tables.sql*.

It is helpful to save the “drop table ...;” commands in a *drop\_tables.sql* file. So every time you start working on the database, you can just run

```
> .read drop_tables.sql  
> .read create_tables.sql  
> .read populate_tables.sql
```

#### Part 4 (SQL Interaction)

##### 1. *Data Selection:*

Give *at least five* sample queries of your database that allows a user to extract interesting information from the database. These data selection queries should have variety and should not be trivial. They should include at least GROUP BY and nested queries.

Save these commands in the file *selection\_queries.sql*.

##### 2. *Data Modification:*

Give *at least five* sample data modification commands, e.g., INSERT, UPDATE, DELETE. Again, at least some of these commands must be non-trivial and include nested queries.

Save these commands in the file *modify\_queries.sql*.

#### Part 5 (Database programming)

Implement a program (using e.g., Python) to allow the user to interact with the database, providing the capability of a set of queries that the user can issue. This program could be a simple command-line user interface, or a more graphical user interface, or a Web interface --- the choice is yours.

#### Submission Instructions

Submit all your files in a zip file via Blackboard. The zip file should include the following:

- A report in Word or PDF format with the following content:
  - Description of the scenario and the database
  - E/R diagram
  - Relational schema, including documentation of the five steps from Part 2.

- o Data you populate the database with in *populate\_tables.sql*.
  - o Description of the data selection queries which you save in *selection\_queries.sql*, including the results returned by MySQL.
  - o Description of the data modification queries in *modify\_queries.sql*.
  - o Summary of the PHP implementation.
- *create\_tables.sql*
- *populate\_tables.sql*
- *selection\_queries.sql*
- *modify\_queries.sql*
- The program implementing the user interface that interacts with the database