

## Lab 3 – Database Design

### This is a two-week lab

The objective of this lab is to assist you in the process of converting your ER diagram design from a conceptual design into an actual database design using Tables in relational databases.

To implement the database design in SQL, you will need to create a table for each entity. A table to represent an entity should have

- A column for each attribute, with an appropriate datatype and with an additional decision as to whether the column can have null values or not.
- A primary key and potentially candidate keys
- An ID column (an integer usually) to act as a convenient primary key

Each column should be set as being NULL, meaning that it is not required, or NOT NULL, indicating a value must be given. The primary key should be NOT NULL. For instance, in my project, for the director entity, I will create a corresponding table, which would have a column called dirName. Given all directors have a name; therefore, I can set this column to NOT NULL.

Candidate keys are alternative primary keys. For instance in my director table, I can create a primary key of type integer that assigns every director entry a unique ID called dirID. I will designate this to be my primary key. However, the dirName is also a unique attributed for each director therefore this makes dirName a candidate key.

In short, for the director entity, I will perform the following steps:

I will create a table called director

The director will have columns called dirID and dirName

The dirID column will be of type INT and NOT NULL

The dirName column will be of type VARCHAR(255) and NOT NULL

dirID is the primary key, so there will be a PRIMARY KEY constraint on it

dirName is a candidate key, so there will be a UNIQUE constraint on it

The UNIQUE constraint ensures that we cannot have to distinct directors with the same name.

This leads to the following table definition:

```
CREATE TABLE director (  
  dirID INTEGER PRIMARY KEY AUTOINCREMENT,  
  dirName VARCHAR(255) NOT NULL,  
  CONSTRAINT ck_director UNIQUE (dirName)  
);
```

In order to design a table for the Movie entity, it follows much the same procedure as for Director, but a foreign key is needed since Movie is on the many side of a one-to-many relationship. Each movie is related to a director, so this information needs to be stored. To achieve this, we design a Movie table with a foreign key to the director table.

Part of a table definition for the Movie entity is shown below. The foreign key represents the many-to-one relationship from movies to directors. Given a movie entry we can use dirID to look up the corresponding director.

```
CREATE TABLE movie (  
  mvID INTEGER PRIMARY KEY AUTOINCREMENT,  
  dirID INT NOT NULL,  
  -- Some other attributes probably go in here  
  CONSTRAINT fk_dir FOREIGN KEY (dirID) REFERENCES director (dirID)  
);
```

Note: In order for SQLite to enforce foreign keys, you must ensure it is enabled on your installation of SQLite. You can try:

```
sqlite> PRAGMA foreign_keys = ON;
```

If the command "PRAGMA foreign\_keys" returns no data instead of a single row containing "0" or "1", then the version of SQLite you are using does not support foreign keys (maybe because it was compiled with SQLITE\_OMIT\_FOREIGN\_KEY or SQLITE\_OMIT\_TRIGGER defined).

You will need to create all of the tables and the appropriate relationships through tables for your ER design.

### **Deliverables**

You should complete the steps described above. Then, you will prepare and submit the results in one single zip file (**YourName\_Lab3.zip**) containing the following item:

- A lab report document: The lab report should be prepared using a word processor, and should be stored as a single PDF file. This PDF document should be named as follows: **YourName\_Lab3\_Report.pdf**. This is what should be included in your PDF file:
  - a. Your name and student number
  - b. A dump of your tables that would include the structure of your tables. You can do this by performing the following command on SQLite:

```
sqlite3 dbName .dump
```

You need to replace dbName with the name of your database.

- c. A short description of each table and how it relates to your ER design. Please include your earlier ER design and make clear how each table is related to this ER design.