1 Preliminaries

In Lab 3, you will add constraints to the database you built in Lab 2. So that we are all using the same database, you will need to create the database using the `createdb.sql` script and the data files provided in the `lab3-createdb.zip` archive.

2 Goal

The third assignment builds on the database you created in the second assignment. We will add constraints on the existing table and will create views based on the existing tables.

1. Add primary key constraints
2. Add foreign key constraints
3. Add general constraints
4. Create views

3 Description

3.1 Add Primary Key Constraints

Primary keys are in **bold**.

```
mg_customers (customer_id, first_name, last_name, email, address_id, active)
dv_address (address_id, address, address2, district, city_id, postal_code, phone)
dv_film (film_id, title, description, length, rating)
cb_books (title, author_id)
cb_authors (author_id, first_name, last_name)
```

Write commands to add primary key constraints in the order the keys are described above, and save your commands to `script1.sql`.

3.2 Add Foreign Key Constraints

Foreign keys are:

```
mg_customers (customer_id, first_name, last_name, email, address_id, active)
is foreign key that references
dv_address (address_id, address, address2, district, city_id, postal_code, phone)

cb_books (title, author_id)
is foreign key that references
cb_authors (author_id, first_name, last_name)
```

Write commands to add foreign key constraints in the order the keys are described above, and save your commands to `script2.sql`.

3.3 Add General Constraints

General constraints are:

1. The length of a film must be a positive integer.

   Note: You will need to give a name to this constraint when you create it. I use the name `positive_length`, but you may use another name.

2. A customer record may not have an empty foreign key to the address table.

   Note: Before adding this constraint, you will need to set the `address_id` for every customer record where it is `NULL`. You are can set it to the `address_id` of an existing address record, or create a new address record. Include this as part of your script.

3. An address record may not have an empty `address` field.

Write commands to add general constraints in the order the constraints are described above, and save your commands to `script3.sql`.

3.4 Write unit test

Unit tests are the key to verifying that your constraints are working precisely as you expect. For every foreign key constraint, provide:

1. A `DELETE` command the violates the foreign key constraint (and elicits an error).

For every general constraint you add, provide:

1. An `INSERT` command that meets the constraint
2. An `INSERT` command that violates the constraint (and elicit an error)

Group all unit tests for each constraint. You should end up with a group of unit tests for each constraint. Arrange the groups so that they appear in the same order as the constraints. Save the unit tests for each constraint (in the order the constraints are given in this document) in the file `script4.sql`.

3.5 Create views

1. Create a view *customer_city* that contains the first name, last name and city_id of all people who are customers of both *Downtown Video* and *City Books*
2. Create a view *district_stats* that contain the district and the number of *Downtown Video* customers living in that district (sorted in the ascending order on the number of customers)
3. Which district has the least number of customers? (Query from the *district_stats* view and list any 5 districts)
4. Alter the name of the *customer_city* view to a new name of your choice

Save the queries in the above order in the file `script5.sql`.

4 Testing

Before you submit, login to your database via `psql` and execute the provided database-creating script and your five scripts in order. The command to execute a script is: `\i <filename>`. Verify that every table has been created and that the attributes are in the correct order. Verify that each attribute is assigned its correct data type using the following command: `\d <table>`. This command will also show you the sequence associated with a particular attribute.

5 Submitting

1. Remember to add comments to the scripts so that the intent of your commands is clear. Put any other information for the grader in a separate `README` file.
2. Copy the scripts to your home directory on `unix.ic.ucsc.edu`.
3. Login to `unix.ic.ucsc.edu`. At the shell prompt, submit your work. **Do not submit the `createdb.sql` script provided for you**.

   ```
   > submit cmps180-wt.w15 lab3 script1.sql script2.sql script3.sql script4.sql
   script5.sql
   ```

   You can submit more than once. Only your latest submission will be graded.