

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 and indexes
5. Write more queries

3 Description

3.1 Declaring Default Values

Write a statement that will declare a default value of `true` to the `active` attribute of `mg` customers and save this command as `script0.sql`

3.2 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`.

3.3 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`.

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. An `mg_customers` tuple must not have an empty foreign key to the address table i.e. the `address_id` field cannot be NULL.

Note: Before you place this constraint on the `address_id` attribute you should make sure that all the tuples in `mg_customers` table which have the `address_id` set to NULL should be given a default value i.e. the `address_id` which are equal to NULL should be set to some value between 0 and 605. (Please include the query for setting the `address_id` from NULL to a default value as a part of the script3)

3. Any `dv_address` table tuple must not have an empty `address` attribute i.e. the `address` attribute in the `dv_address` table cannot be NULL.

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

3.4 Create indexes

Indexes are data structures used by the database to improve query performance. Of course, the trade-off is that indexes must be maintained (as their corresponding tables are updated).

Write a query to lookup all customers with the first name John in `mg_customer`. Then ask the database to show how the query is performed using:

```
EXPLAIN ANALYZE <your query ...>
```

In the comments, write the kind of scan used, and the time required to execute.

When a customer is checking out, they typically provide their last and first name. To speed up the querying of their customer record, create an index called `firstlast` over the `first_name` and `last_name` columns (in that order) on the *mg_customer* table. The syntax is:

```
CREATE INDEX <index name> ON <table name> (<attribute 1>, ..., <attribute n>)
```

Again, ask the database to show how the above query is performed. Does the kind of scan change? Does the time required to execute change? Write your answers in the comments.

Save the command in the file `useindex.sql`.

3.5 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 that 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`.

3.6 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_stat` 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.Create a view `author_title` that contains the first name, last name and title for each book in `cb_books`.
- 4.Which are the books written by Stephen Fry ? (Query from the `author_title` view and you need to output only the title of the book)
- 5.Which district has the least number of customers? (Query from the `district_stat` view and list any 5 districts)
- 6.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`.

3.7 More Queries

1. Write a statement that will change the `active` attribute to `true` for every row in `mg_customers` where `active` is `NULL` and the customer's last name starts with 'B' or 'C'

2. John Smith is a new customer. His details are as follows:

- First Name: John
- Last Name: Smith
- Addressid : 700
- Address: Koshland Way
- City id : 654
- Postal code: 95064

Add the details of the new customer to the `mg_customer` and `dv_address` table. (You can assume the values that are not specified to be `NULL`) Note: one of these rows must be inserted before the other; you'll need to determine what is the correct order.

3. List the title of each film whose length exceeds the average length of all films

4. What is the maximum, minimum and average length of the films in each rating category?

5. Write a statement that will delete all the books written by Nevil Shute

Save the queries in the above order in the file `script6.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.lt.ucsc.edu`.
3. Login to `unix.ucsc.edu`. At the shell prompt, submit your work. **Do not submit the `createdb.sql` script provided for you.**

```
> submit cmps182-dm.u17 lab3 script0.sql script1.sql script2.sql script3.sql  
script4.sql script5.sql script6.sql useindex.sql
```

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