Print Close

databases

From: **Radu Serban** (top_turkey@hotmail.com)
Sent: November-12-15 5:35:18 PM
To: adrian.serban@sympatico.ca (adrian.serban@sympatico.ca)

9 attachments
250766858ERDiagram.pdf (81.3 KB) , 250766858outputscript1.txt (1.5 KB) , 250766858outputscript2.txt (7.7 KB) , 250766858outputscript3.txt (4.5 KB) , 250766858outputscript4.txt (2.0 KB) , 250766858script1.txt (1.1 KB) , 250766858script2.txt (3.6 KB) , 250766858script3.txt (2.0 KB) , 250766858script4.txt (0.6 KB)

The only two parts of the assignment I didn't complete are the 1% bonus thing, and the first part of the 4th file, which is 60% done.

The Problem:

A movie ranking company wants to put together a database to keep track of customers and what movies they go to and what rankings they give movies.

In this assignment you will create the database that will be used as the backend database for assignment 3. Thus, in assignment 3 you will write the application that uses this database you are about to create.

Your database will need to keep track of the following information:

- The customers who buy the tickets:
  - first name (up to 30 characters, must have at least 1)
  - the last name (up to 30 characters, must have at least 1)
  - the customer ID number (unique - 3 digits, make this the primary key)
  - the sex of the customer
  - the customers email address
- The movies:
  - A movie id number (unique, - 3 digits, make this the primary key)
  - The movie name (up to 50 characters, must have at least 1 character)
  - The year the movie was released (integer, 4 digits)
  - The genre of the movie (e.g. Comedy, Drama), a movie can have multiple genres.
- The showings:
  - A showing id number (unique - 3 digits, make this the primary key)
  - A date of the showing
  - A time for the showing
- The theatre rooms:
  - A room number (unique integer, make this the primary key)
  - A capacity (how many seats it can hold)
- You will need to keep track of which movie is being shown at a showing
- You will need to keep track of which theatre room was used for a showing.
- You will need to keep track of which showing a customer selected and how much he/she paid for the showing. After the customer finishes watching the movie, he/she will give the showing a rating of 1 to 5 stars.
- A customer is only added to the customer table IF they go to a showing.
- A movie is only added to the movie table IF we have a showing of it.
- If the movie is unpopular, no one might buy tickets for a showing
- Sometimes the rooms have no showings

Instructions:

There are 2 parts for this assignment. In part 1 you will draw an ER diagram and in part 2 you will create the database on your virtual machine using MySQL.

Part 1:

For the above scenario, draw an ER diagram using MS Visio or draw.io. Then save your ER diagram as yourwesternuseridERDiagram.jpg (or .pdf, or .gif). You can just show the total or partial participation using the single/double lines (you do NOT need the (min,max) info. Make sure you indicate the primary key and the cardinality.

Part 2:

Create the following 4 script files (more details to follow):

1. File 1: Delete the database and create a new database and create the tables
2. File 2: Insert some data and update the data
3. File 3: Query the data
4. File 4: Delete some data, add a constraint, add a view

HINTS:

- You might want to use WinSCP or Filezilla, to ftp back and forth your text file with your commands from your home machine to the virtual machine if you are more comfortable with Notepad then with Nano or EMACS or VI. REMEMBER THAT WE ARE NOT BACKING UP YOUR AMAZON MACHINE, so it is safer to keep your data on your home machine as well as your Amazon machine.

File 1 - Creation

- List all the current databases
- Delete the database called yourwesternuseriddb if it exists
- Create a database called yourwesternuseriddb
- Select (use) that database
- List all the tables (should be none initially)
- Create the tables you need with the appropriate types and keys, foreign keys. Make sure that:
    - if an theatre room is deleted and there are movies showing in it, then the theatre can not be deleted until the showing is assigned to another theatre room.
    - if a customer is deleted and this customer had bought some tickets, then any rows showing that the customers ticket purchases should also be deleted from the ticketing information table.
- List the tables again
- Call this script: *yourwesternuseridscript1.txt*
- After you have the script working, save the output from it using the following command:
- ***sudo mysql --verbose -pyourpassword < yourwesternuseridscript1.txt > yourwesternuseridoutputscript1.txt 2>&1***

File 2 - Insertion and Modification

- Using the ***LOAD DATA INFILE ...*** mysql command (you will need to google to see the syntax for this command), load the following data into the movie table: [http://www.csd.uwo.ca/~lreid/cs3319/assignments/assignment2/fall2015data.txt]
    - NOTE: for this command, you will have to copy the above datafile to the /tmp directory on your virtual machine AND you will have to make mysql the owner of the data text file using the command: ***sudo chown mysql /tmp/fall2015data.txt***
- Insert your favourite movie into the movie table also using the INSERT command
- Show everything in the movie table
- Into the customer table, load the following data using the SQL INSERT command (the first 3 digit number is the customer id):
    - *222,Laura,Reid,F,lreid@csd.uwo.ca*
    - *223,**Your,Name,?**,Your@email.com* (change to your name and sex)
    - *224,Bill,Clinton,M,billy@prez.com*
    - *225,Hilary,Clinton,F,nextprez@pres.com*
    - *226,Donald,Trump,M,nextprezalso@prez.com*
    - *227,Chelsea,Clinton,F,firstdaug@prez.com*
    - *228,Ivanka,Trump,F,apprentice@trump.com*
    - *229,Sandra,Bullock,F,sandy@blindside.com*
    - Show all the rows in the t.a. table
- Using your keys and foreign keys, add the genres for each movie by making a table that somehow indicates the following data (make sure your design is correct).
    - *Star Wars->Action,SciFi*
    - *Up->Animated*

  - *Pulp Fiction->Action*
  - *Aloha->Romance*
  - *The Martian->Action,SciFi*
  - *The Empire Strikes Back->Action,SciFi*
  - *Midnight In The Garden Of Good and Evil->Drama*
  - *Tommy Boy->Comedy*
  - *Die Hard->Action*
  - *There's Something About Mary->Comedy*
  - *Two Weeks Notice->Comedy,Romance*
  - *Back To The Future->SciFi,Action,Comedy*
  - *Add at least one genre for your favourite movie*
  - Show all the rows in the rows in this table
- Make your theatre table have the following data:
  - *RoomNumber,Capacity*
    - *1,30*
    - *2,5*
    - *3,100*
    - *4,30*
    - *5,6*
  - Show all the rows in this table after you finish your inserts
- Using your keys and foreign keys store the following SHOWING information (info that needs to go into the showing table)you can do this at the same time as your insert command if you want): NOTE: the last number indicates which theatre room the showing was assigned. The second 3 digit number indicates the showingid:
  - *Star Wars--333--11/Feb/15--9:30 PM--1*
    Star Wars--334--11/Feb/15--7:30 PM--2
    Up--335--11/Feb/15--7:00 PM--3
    Pulp Fiction--337--12/Mar/15--7:00 PM--1
    Aloha--338--12/Mar/15--9:00 PM--1
    Aloha--339--13/Mar/15--7:00 PM--2
    Aloha--340--20/Mar/15--7:00 PM--1
    The Martian--341--20/Mar/15--9:00 PM--1
    Tommy Boy--346--22/Mar/15--8:30 PM--2
    Die Hard--345--22/Mar/15--9:00 PM--3
    There's Something About Mary--342--20/Mar/15--7:00 PM--2
    Two Weeks Notice--344--21/Mar/15--7:00 PM--1
    Two Weeks Notice--347--23/Mar/15--10:00 PM--1
    Back To The Future--336--11/Feb/15--9:00 PM--2
  - Add one more showing for your added movie
  - Show all the rows in this table after your inserts
- Using your keys and foreign keys store the following information about who watched which movies. Remember, don't use names to show this instead use the keys (I am just showing you the names but you should look up the customer keys), also note that the first 3 digit number is the showing id and the second number is the rating out of 5 stars that the customer gave the movie:
  - *333--5--$4.00--Laura--Reid*
    340--1--$12.00--Laura--Reid
    334--3--$10.00--Laura--Reid
    346--4--$5.00--Laura--Reid
    334--5--$30.00--Your--Name
    340--1--$5.00--Your--Name
    334--4--$20.00--Bill--Clinton
    334--2--$100.00--Donald--Trump
    334--1--$5.00--Chelsea--Clinton
    347--5--$12.00--Sandra--Bullock
    334--3--$5.00--Sandra--Bullock
  - Add a row that has you viewing the movie you added.
  - Show all the rows in this table after your inserts
- Change/Modify *Ivanka*'s first name to be *Sue*
- Change it so that any rating for *Star Wars* is always a 5 (don't use the Star Wars movie id to do this, instead refer to the movie name in your WHERE ... query so that you are force to try to do a join).
- Call this script: *yourwesternuseridscript2.txt*
- After you have the script working, save the output from it using the following command:

  ```
  sudo mysql --verbose -pyourpassword < yourwesternuseridscript2.txt  > yourwesternuseridoutputscript2.txt 2>&1
  ```

File 3 - Queries

1. Show the last name of all customers with duplicates
2. Show the last name of all customers without duplicates
3. Show the first name, last name and email address of your male customers
4. Show the movie title of all films that are comedies.
5. Show the movie title and movie year of all nineties movies order in ascending order by year.
6. Show the showing date and time and the capacity of the theatre room for any time the movie Star Wars was shown. Google to figure out how to use the MySQL function DATE_FORMAT and/or TIME_FORMAT to print the month out in long format (e.g. display February rather than 2) and print the time out with AM or PM. Also give good heading names for your columns in this query, for example, the top rows of your query should look like this:

   ```
   +------------------+-------------+----------+
   | Showing Date     | Showing Time | capacity |
   +------------------+-------------+----------+
   | February-11-2015 | 09:30:00 PM  |       30 |
   | February-11-2015 | 07:30:00 PM  |        5 |
   +------------------+-------------+----------+
   ```

7. List all movie titles that the word "the" in their title.
8. List all the first name and last name of people who went to see a movie on or after March 1, 2015 and the name of the movie that they went to see.
9. List the first name and last name of people who went to see a sci fi movie. Don't show duplicates.
10. List the movie titles of any movies that have no showings
11. List the movie titles of movies that have showing but no one has watched them (it is okay to have repeats here)
12. List the movie titles of movies that have NOT been seen by a customer (either had no showings or had showings but no one watched them, be careful, because some movies had several showings, if a movie was watched at any showing, the movie title should NOT be included in your result, you may have to build a view(s) to do this one in several steps or look up the UNION command in mysql.)
13. List each year a movie came out and count the number of movies that came out that year.
14. Display the total sales value of all showings (add up all the sales to get a final value).
15. Find the total number of customers who attended each showing AND display the capacity of each room to see if any showing was overbooked.
16. Call this script: *yourwesternuseridscript3.txt*
17. After you have the script working, save the output from it using the following command:

    ```
    sudo mysql --verbose -pyourpassword < yourwesternuseridscript3.txt  > yourwesternuseridoutputscript3.txt 2>&1
    ```

File 4 - Deletions/Views

- Create a view that shows the customer first name, last name, the movie names they have seen and the rating they gave the movies. IF they saw the same movie more than once, show the average rating for the movie. Then show all the rows in this view
- Delete any customer whose last name is "Trump". Show any tables affected by this deletion before and after the deletion.
- Delete room 3 from the theatre table. NOTE: if you can't delete this row, show what you had to do in order to eventually delete the row.
- BONUS: Unfortunately CHECK CONSTRAINTS don't work in MySQL, but you can use a trigger to make sure that the capacity of a theatre will always be more than 0. Figure out how to write a trigger (use Google) and then prove that it works and you will get a bonus 1% added to your mark out of 100 for this assignment.
- Call this script: *yourwesternuseridscript4.txt*
  - *NOTE: the script will break/stop executing when you hit a line that can't be executed, if this happens make 2 scripts: script4a.txt and script4b.txt but make sure you show everything you tried*
  - *However if you put 2>&1 at end of the unix command, it will redirect error output to the script as well, so this might save you from having to create 2 files)*
- After you have the script working, save the output from it using the following command:

  ```
  sudo mysql < yourwesternuseridscript4a.txt --verbose >  yourwesternuseridoutputscript4a.txt  2>&1
  sudo mysql < yourwesternuseridscript4b.txt --verbose >  yourwesternuseridoutputscript4b.txt  2>&1
  ```

  OR (if you can get the errors redirected, you can use this command with -f to force the continuation of the script)

  ```
  sudo mysql -f < yourwesternuseridscript4.txt --verbose >  yourwesternuseridoutputscript4.txt  2>&1
  ```

Notes:

- Here is a sample script file for mysql -->   [http://www.csd.uwo.ca/~lreid/cs3319/assignments/assignment2/samplescriptmysql.txt]
- To run the script, you would type this:

  ```
  sudo mysql  --verbose -pcs3319 < samplescriptmysql.txt > outputfromscript.txt  2>&1
  ```

- Do the steps (inserts, deletes, etc...) in the order given
- **For any delete or update commands remember to show the table before and after you modified it**

Handing in the Assignment

You are required to submit the following files via Owl:

- your ER diagram (named yourwesternidERDiagram.jpg/.pdf/.gif)
- your 4 or 5 script files
- your 4 or 5 files containing the output from the scripts.