index category "Name" t make foreign key

CS3860 Lab3: lab3-video-multiple-entity.zip

Objectives

- Import and export text files into database tables.
- Analyze existing data to define an ERD using multiple entity relationships following the relational data model.
- Define database schema from ERD.
- Use SQL to answer business questions using *group by* aggregations, multi-table joins, sub-queries/nested queries.

Application

- Using the *relational data model*, design a database relating actors, videos, and video categories.
- Answer questions about actor, video, and category relations and relationships with SQL.

. **NOTE: The schema for your DB design is not the same as your import tables, though they may overlap!**

Part I – Create a video database

Complete and **document** each of the following steps (*all the sql you create/use for these steps should be included as part of your report*):

1) Review *videodb-readme.txt* which contains field names for the Video_Categories.txt, Video_Actors.txt, and Video_Recordings.txt data files.
2) Create a *Video* database.
3) Create an SQL script (a .sql file with the commands saved in it) to create *import tables* for importing the data as *tab-delimited* text files. Note: the differences between importing *csv*- and *tab-delimited* files. **Answer the following question as part of your documentation:**
   - **Why would you select one format over another?**
4) Load the tables from the data files. Use select commands on each table to verify the data has been successfully imported. See the load table primer at the end of this assignment if you need help with this step.

   Notes:
   - The tables were created on a Windows machine, so use:
     lines terminated by '\r\n' in your *load data local infile…* commands.
   - Columns are separated with tabs, so use:
     fields terminated by '\t' in your *load data local infile…* commands.
   - Be careful to review each table imported. Some of the text fields have extra quotations. You need to remove these to get the correct results. The following SQL will remove extra white-space and double-quotes. Note: replace a double-quote with nothing.

     update Video_Categories set name = replace(trim(name),'"','');
     Note that '"' is ' " ', single quote-double quote-single quote (without spaces)

     Another approach is to remove double quotes with your *load data local infile* command with the *optionally enclosed by* option as follows:

     Load Data local Infile 'C:/YOUR_PATH_HERE/lab3/Video_Recordings.txt' into table video.Video_Recordings fields terminated by '\t' ***optionally enclosed by*** *'"' lines terminated by '\r\n';*
     Note that '"' is ' " ', single quote-double quote-single quote (without spaces)

file forward engineer

5) Create both an ERD and a relational schema for your final data model: Use MySQL Workbench Modeling tool or erdplus (ERD and ***relational schema)*** to show the recording, category, and actor as entities and relationships for your final database schema.

Note: Your relational data schema will not necessarily be the same as your load tables. It is important for you to ***review the data***, identify the underlying entities, entity attributes, and relationships between entities and their cardinality.

6) From MySQL Workbench (or ERD plus), generate an SQL script to define and create the database tables for your DB schema (or draft the sql yourself). Once your diagram is complete, use *File -> Export -> Forward Engineer SQL CREATE Script* to generate your SQL create tables script.

- OR use erdplus relational schema, and then Export SQL under the Diagram gear menu in the upper left.

7) If you are using MySQL Workbench, use ENGINE=INNODB at the end of each create table statement, but before the semi-colon. If you are interested, here is some more information about that: https://stackoverflow.com/questions/1082194/when-should-you-choose-to-use-innodb-in-mysql.

8) Draft SQL to load your final schema tables from your import tables (and save this, remember all the sql you use for these steps should be part of your report).

9) Verify primary key and foreign key constraints for each relation. ***If not present, create them***. **Answer this question as part of the documentation for this step:**

- **Why would I create the primary key index *after* the table has been created and the data imported *versus* defining the primary key in the table definition?**

10) Run select commands on each table to verify the data has been successfully imported. ***Your goal should be to finish this step by the start of second lab period set aside for Lab 3***.

Part II – Document and Answer each of the following questions using SQL

Solve on original / schema

Answer each of the following questions. For each question, include the question, your SQL, and query results. For large query result sets, submit first 10 rows – however *you MUST include the total number of results* for all questions (i.e. mention you get 23 results for question X, even if only showing 10 of them).

1) Execute *select * from Video_Recordings, Video_Categories*. Note the cross-product effect of joining two tables. Record the number of rows generated. Do all permutations of Video_Recordings X Video_Categories make sense? Explain.

2) Execute *select * from Video_Recordings vr, Video_Categories vc where vr.category=vc.name*. Note the cross-product effect of joining two tables when restricted on the appropriate keys. Record the number of rows generated. Explain the purpose of the join.

3) List the number of videos for each video category.

4) List the number of videos for each video category where the inventory is non-zero.

5) For each actor, list the video categories.

6) Which actors have appeared in movies in different video categories?

7) Which actors have not appeared in a comedy?

8) Which actors have appeared in comedy and action adventure movies?

See Canvas for submission date. Include ERD, relational schema, documentation mentioned in Part I, and answers to questions in Part II in your lab report. For each question, list the question, your SQL statement, and results. For large results, just submit first 10 rows – but mention how many results total.

Importing/Exporting Text File Primer (review of what we did in Lab 1)

MySQL

1) Start the MySQL command line tool:
   ```
   $mysql -u root -p –local-infile=1
   > SET GLOBAL local_infile=1
   ```
2) Import your data file into the appropriate table:

```
Load Data local Infile 'C:/YOUR_PATH_HERE/lab3/Video_Categories.txt' into table
Videos.Video_Categories fields terminated by '\t' lines terminated by '\r\n';
```

3) Verify your table has been successfully imported by running a select command.
   ```
   > select * from Video_Categories limit 10;
   ```
4) Note: you use a similar command for export.

http://dev.mysql.com/doc/refman/5.1/en/load-data.html