In-class Exercise for Week 5: Getting Information Out of a Database

Objective: Learn to retrieve information from a relational database using the SQL SELECT statement.

Learning Outcomes:

- Understand and apply SQL syntax
- Transform an English-language business question into a SQL statement.
- Identify data necessary to include in a query based on a schema.

Step 1: Individual (30 minutes)

To do this exercise, you will be working with a movie rental database. The schema for this database is provided on the accompanying document. All of the tables are in a schema called “moviedb.” You can’t write to any of the tables – you can only use SELECT statements to read from them (so don’t worry about causing any damage).

Spend some time looking at the schema carefully. The field names are pretty self-explanatory. For example, here are three tables:

You can see that a film has a title, description, rating, and length (among other things). You can also see that an actor has a first name and a last name. The film_actor table implements the many-to-many
relationship between actor and film (i.e., a film can have more than one actor, and an actor can be in more than one film). You’ll also notice that data types are listed for each field, but they should be pretty obvious – for example, first_name is a VARCHAR because it is a string value.

In MySQL Workbench, open the connection to the class1 server using your username and password. Click on the moviedb schema to see the list of tables.

Now try a simple query. In the Query 1 pane, type the following:

```
SELECT film.title FROM moviedb.film
```

And then click the Execute SQL Script button (the lightning bolt):

And you’ll see a list of all movie titles (this is just the first few):

<table>
<thead>
<tr>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACADEMY DINOSAUR</td>
</tr>
<tr>
<td>ACE GOLDFINGER</td>
</tr>
<tr>
<td>ADAPTATION HOLES</td>
</tr>
<tr>
<td>AFFAIR PREJUDICE</td>
</tr>
<tr>
<td>AFRICAN EGG</td>
</tr>
<tr>
<td>AGENT TRUMAN</td>
</tr>
<tr>
<td>AIRPLANE SIERRA</td>
</tr>
<tr>
<td>AIRPORT POLLOCK</td>
</tr>
<tr>
<td>ALABAMA DEVIL</td>
</tr>
<tr>
<td>ALADDIN CALENDAR</td>
</tr>
<tr>
<td>ALAMO VIDEOTAPE</td>
</tr>
<tr>
<td>ALASKA PHANTOM</td>
</tr>
</tbody>
</table>

On the following page you’re going to create a series of SQL SELECT queries to answer questions about the information in this database. Some of the questions can be answered by querying one table; others will require joining multiple tables to get the answer. **Make sure you return only the information being requested!**

For each question you’ll write down the SQL query (which you can copy and paste it from SQL Workbench) and the answer you get as a result of the query (which you can copy and paste from the results).
You should try to do as much of this on your own as you can, but you should feel free to consult with your colleagues (or me) in writing your queries.

**PART 1: SINGLE TABLE QUERIES**

1. Which actors have the first name of “Ed”?

2. How many PG-13 movies are in the database?

3. What is the length of the film “Confessions Maguire”?

4. What movie has the longest running time?

5. Are PG movies, on average, longer than R movies? Prove it!
PART 2: MULTI-TABLE QUERIES (JOINS)

6. How many movies are in French?

7. What were the titles of the film starring Julianne Dench? (create a query to get them all, but only list the first five)

8. How many films star Julianne Dench?
**FOR AN EXTRA CHALLENGE!**

How many actors have a last name that starts with ‘W’?

Two hints:

1) You can use LIKE with a WHERE clause, like this:

   ```sql
   SELECT fieldname FROM tablename WHERE fieldname LIKE value
   ```

   LIKE looks for close matches, not exact ones like = does.

2) You can use % as a wildcard value. So LIKE ‘ap%’ will match with “apple,” “application,” “apex,” etc.

**Step 3: Class Debrief (20 minutes)**

Review answers and queries for each question.