

## CS 327E Project 1, due Thursday, 02/10.

1. Open a terminal window in JupyterLab and download the sakila dataset from Google Cloud Storage. Run the following commands to download and extract the dataset:

```
gsutil cp gs://cs327e-open-access/sakila.zip .  
unzip sakila.zip
```

Open the sakila folder and look at the three files in this folder:

`sakila-database.sql`, `sakila-data.sql`, and `sakila-diagram.png`. If there is a data type in the create table statements which you don't recognize, look it up in the [MySQL documentation](#).

2. Create a new Python Jupyter notebook and name it `project1.ipynb`. Implement the following logic in your Jupyter notebook:

- Create the sakila database and database objects by running `sakila-database.sql`.
- Populate the tables by running `sakila-data.sql`.
- Get a row count for each table in the database.
- Write a query to sample a few records from each table using the LIMIT clause.
- Write one query on any table that uses both a WHERE clause and ORDER BY clause. Add a short comment above your SQL statement to describe the query.
- Write an INSERT statement to add a record into any one of the tables. Add a short comment above your SQL statement to describe the SQL.
- Write an UPDATE statement to update one or more records from any one of the tables. Add a short comment above your SQL statement to describe the SQL.
- Write a DELETE statement to delete one or more records from any one of the tables. Add a short comment above your SQL statement to describe the SQL.

CS 327E Project 1 Rubric

**Due Date: 02/10/22**

Download and extract the sakila dataset to your jupyter notebook instance. -5 no dataset or incorrect dataset found in Jupyter instance	5
Create a new Python Jupyter notebook named <code>project1.ipynb</code> . -5 incorrect file name	5
Create the database based on <code>sakila-database.sql</code> . Populate the tables from the <code>sakila-data.sql</code> file. -30 missing sakila database -7 for each missing table or incorrect data load	30
Run a row count of each table in the database. -3 each missing row count	15
Run a query that samples the data from each table using the <code>LIMIT</code> clause. -3 for each missing query	15
Run a query that includes a <code>WHERE</code> clause and <code>ORDER BY</code> clause. Include a short comment above your query to explain its function. -5 missing <code>WHERE</code> clause -5 missing <code>ORDER BY</code> clause -2 missing comment or comment is not descriptive	10
Run other CRUD operations: <ul style="list-style-type: none"> <li>• An <code>INSERT</code> statement into a table.</li> <li>• An <code>UPDATE</code> statement on a table</li> <li>• A <code>DELETE</code> statement on a table</li> <li>• A short comment above each statement to explain their function. -5 each incorrect statement -2 for each missing comment or comment is not descriptive</li> </ul>	20
<code>project1.ipynb</code> pushed to your group's private repo on GitHub. Your project <b>will not</b> be graded without this submission.	<b>Required</b>
<code>submission.json</code> submitted into Canvas. Your project <b>will not</b> be graded without this submission. The file should have the following schema:  <pre>{   "commit-id": "your most recent commit ID from GitHub",   "project-id": "your project ID from GCP" }</pre> Example:	<b>Required</b>

<pre>{   "commit-id": "dab96492ac7d906368ac9c7a17cb0dbd670923d9",   "project-id": "some-project-id" }</pre>	
<b>Total Credit:</b>	<b>100</b>