SQL – The basics

Before you begin please do the following:

1. Read this document to install MAMP and log into phpMyAdmin: http://rachelfadler.com/database/documents/InstallMAMP.pdf
2. Watch the **video lecture** posted on how to use MySQL. Many of the commands I will show you on the video.
3. You can also look at the SQLCheatsheet tab for many of the common SQL commands.
4. The below exercises correspond to the lecture titled SQL Basics (2 lectures on D2L, long and short).
5. Please note that all assignments (unless otherwise specified) must be your own and done independently or you will receive a 0. If you provide your answers to another student you will also receive a 0.

0. Getting Started

- **a.** On phpMyAdmin:
  - **i.** Click on the database name you would like to work in (ex. cs315_fall2020)
  - **ii.** Click on the SQL tab
- b. Open up a Microsoft Word document - save it with an appropriate filename on your computer. When your assignment is complete, submit this file to the folder titled SQL in the assignments tab on D2L.

I. Creating Tables

Write each SQL statement in phpMyAdmin. Before hitting Go, paste it into the Word document, since you will need to submit each statement.

After hitting ‘Go’, if there are errors, remember to paste the corrected version back into your Word document.

Note: Copying quotes from sql statements typed in Microsoft Word to the SQL editor will cause errors as Quotes (‘ “) from Microsoft Word are not the same symbols as quotes in text files (', ").

Write the statements to create the following tables:

Note: If you click on a table name from the left column and select the Structure tab, you will be able to see the structure of the tables you just created.

1. Movie – **Create** a table Movie which contains the following attributes:
   - i. *movieId (integer/int – unique number for each movie – you do not need () after using INT)*
   - ii. *title (representing the name of the movie - use varchar – you decide up to what length you think you will need in ())*
   - iii. *genre (varchar)*
   - *iv.* *release_year (the year it was released – use YEAR)*

2. Actor – **Create** a table Actor which contains
    - i. *actorId (int)*
    - ii. *fname(varchar – you decide the length)*
    - iii. *lname(varchar – you decide the length)*
    - **iv.** a column of your choice that you think is important to know about each actor!

II Inserting Rows

Note: If you click on a table name from the left column and select the Browse tab, you will be able to see the rows of the tables you just inserted into.

The **INSERT** statement is used to insert new records in a table.

Write the following insert statements:

3. Insert a movie into the table using the following SQL (go to the SQL tab):

INSERT into Movie (movieId, name, genre, release_year) VALUES (1, 'Sully', 'Drama', 2016)

If a "1 row inserted" message is displayed, you successfully populated the table.

4. If you are inserting all columns anyway, you can save time by inserting without specifying the column names. Try this:

INSERT into Movie VALUES (2, 'La La Land', 'Musical', 2016)

5. Insert at least **5** additional rows into the Movie table. Choose your favorite movies and complete all attributes. Have at least 2 in the same genre.

6. Insert at least **6** actors or actresses that are from the movies you have selected. Note: Actor with id 1 should be Emma Stone. You can choose the other 5. Make sure to specify a value for every column for each row.

Hint: You can insert multiple rows at one time like in this example where the table has 3 columns:(see lecture notes for details)

```
INSERT INTO Table-Name
VALUES
(col-1, col-2, col-3),
(col-1, col-2, col-3),
(col-1, col-2, col-3),
(col-1, col-2, col-3),
(col-1, col-2, col-3)
```

When you are done clicking on the table name will show you the columns in the table.

III Selecting Rows

In order to retrieve data from a database table you have to use the SELECT statement. **Write each of the below statements:**

7. Select all columns from your Actor table using the below format.
   SELECT *column_name,column_name FROM table_name*;
8. Paste the results of the statement (you should see all of your data for all of your rows)
9. Now enter SELECT * FROM Actor (* selects all columns!)
   What do you see? (paste the results and let me know)
10. Write the statement to list all the details for all rows in the Actors table **ordered by the actors' last name.**
11. Paste the results of the statement.
12. Write the statement to list **ONLY** the genre from the Movie Table.
13. Paste the results of the statement.
14. Write the statement to select all **distinct** genres from the Movie table.
15. Paste the results of the statement.
16. Write the statement to select only the release year **where** the movie is La La Land.
17. Paste the results of the statement.
18. Oh no! You can't quite remember the title of the movie, was it La Land or La La or La La Land. Write the statement to select the release year given that it's something **LIKE** La La.
19. Paste the results of the statement.

IV Deleting Rows

Write statements to delete rows from your table.

20. Write a statement to delete Sully from the Movie table.
21. Please verify that the movie is no longer in the table by running the necessary select statement. Write the statement.
22. Paste the results of the statement.

V Updating Rows

Write statements to edit rows in your table.

23. Write a statement to update the genre of one of your movies.
24. Please write a statement to view that your changes took place correctly.
25. Paste the results of the above statement.
26. Write a statement to update two attributes for one specific row in any table. You must do this in one statement.
27. Please write a statement to view that your changes took place correctly.
28. Paste the results of the above statement.

Submission: Submit the Word document (which contains both the SQL code and output results) to the folder titled SQL in the assignments tab on D2L.