Summary/Subquery Assignment (MIS 381) Aug 2020.v1

# Instructions:

* **Do your own work:** This is an individual assignment and you must do your own work and create your own SQL statements. If you are caught cheating on this, you will receive a zero on this assignment and be reported to the Dean of Students. Also, this homework prepares you for the exam coming up so doing the work now will help you learn and do well on when it counts more.
* **What to turn in**
  + Clearly separate your code for each question. Save your code into one SQL file with the naming format: LastName\_FirstName\_UTEid. Please make sure the lastname and firstname you use matches what is in Canvas.
  + Submit your .sql file on Canvas before the deadline. Late submissions receive 50% off. No submissions will be accepted 24 hours after the deadline.
  + Only submissions in .sql or .txt format will be accepted. All other file formats will be given a 0.
* The SQL problems below will be based on the DDL script that is posted on the Canvas instructions. Download that script and run it before you start.

# Section 1 Problems:

1. We would like summary information listing in one row the total number of content creators that exist, the minimum video length, the maximum video length, and the maximum views on a video. Use appropriate column aliases.
2. What is the number of user comments and most recent date of comment activity for each video (show video title)? Please use table and column aliases. Sort results from oldest to newest comments
3. Write a query that shows us the number of content creators in each city, along with the average number of likes for videos from each city. Show the city with the highest number of average video likes first.
4. Write a SELECT statement that properly joins *topic* and *video* tables and then returns information on topic\_id, topic name, average video size, total number of likes within each topic.

4b. Repeat problem 4 using window functions instead of GROUP BY.

1. Write a statement that will return **one row for each user** that has a ContentCreator account and videos attached to the account and reports user first name, lst name, and a column called *awards\_earned* that provides the number of awards earned on videos for each user. The formula to calculate awards is views minus the 100 baseline requirement and then divide the result by 5,000. Be sure to only show the awards earned as a whole number and do not round up (i.e. Videos that have earned 4.8 awards have only earned 4 awards).

Filter results to only show videos that have earned 10 or more awards. Sort results by *awards\_earned* largest to smallest and then last\_name A-Z.

1. Part A - Write a SELECT that creates a report to show the count of credit cards broken out by user first\_name, city\_billing and state\_billing. The report should include the following

The first\_name column from the user\_table

The city\_billing column from the CreditCard table

The state\_billing column from the CreditCard table

The count of cart\_id in the CreditCard table with an appropriate column alias

Filter data to show only records related to users with billing addresses in Texas and NY and sort by city.

Use the ROLLUP operator to include a row that gives the subtotal city\_billing and state\_billing.

Part B – Explain in a commented sentence how the CUBE operator is different than ROLLUP and why it is useful.

1. Write a query that shows us the main billing address of each content creator and then returns the credit card ID and a flag (Y/N) specifying whether that credit card billing address matches the primary billing address for each content creator. You may choose to also return the credit card billing address if you wish. Return a row for each credit card listed.
2. Write a SELECT statement that displays the cc\_id and count of video\_id for all the videos a content creator has made. In a 3rd column called *unique\_topics* display the count of distinct topics for videos created by a certain user. Lastly only show results that have had at least 2 or more distinct topics. Sort the by cc\_id descending.

8b. Now write using window functions. Do not filter by number of distinct topics.

# Subquery Problems:

1. Write a SELECT statement that returns the same result set as this SELECT statement, but don’t use a join in the main clause. Instead, use a subquery in a WHERE clause that uses the IN keyword.

**SELECT DISTINCT topic\_name  
FROM topic t JOIN video\_topic\_link vtl  
ON t.topic\_id = vtl.topic\_id JOIN video v  
ON vtl.video\_id = v.video\_id  
ORDER BY topic\_name DESC;**

1. Write a SELECT statement that answers this question: Which users have created a video that has greater than the average likes for all videos?

Return the user\_id, video\_id, and likes for each video.

Sort the results by the likes.

1. Real-World Business Scenario! The Marketing Team has asked us to pull together a target customer contact list that contains all the users who are signed up with Content Creator accounts but haven’t uploaded a video yet. Write a query that returns the first\_name, last\_name, email, CC\_flag, and birthdate columns from the *User*\_table. Return one row for each user that has never had a video on the Video table.
2. Write a SELECT statement that returns the title, subtitle, video\_size, views, and the number of comments for videos that have at least two comments on them. We want you to use a subquery to do this!

Sort the final results by the number of comments.

1. Write a query that returns user information (id, first and last name) along with the count of videos for each user. You must do this by utilizing a left join with the subquery as an inline view. Order final results by last\_name.  
     
   **BONUS!** Obtain the same output by using the subquery in the WHERE clause (+2 points).
2. Using an inline view, write a statement that will return one row per content creator, representing the most recent video that person has made. Each row should include these three columns: cc\_id, username, and days\_since\_latest\_upload.