**Name: Mahmood Rasheed**

**Roll no: 20SW047**

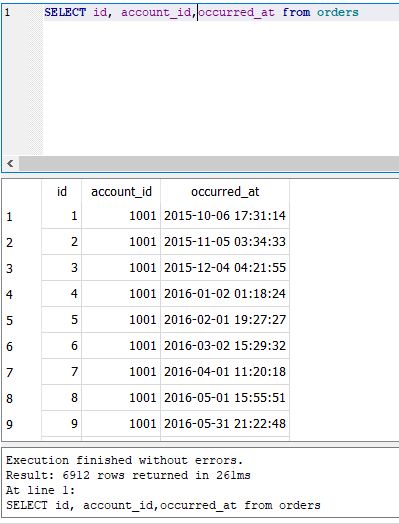
**Section: 1**

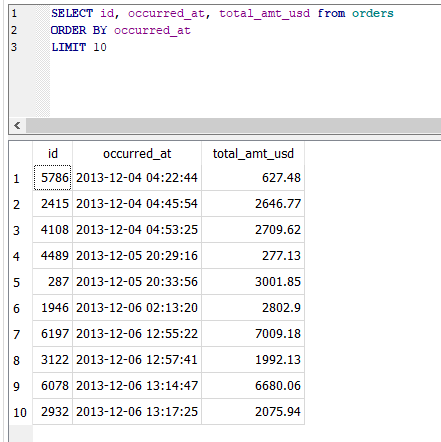
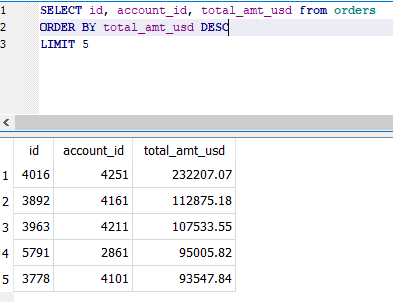
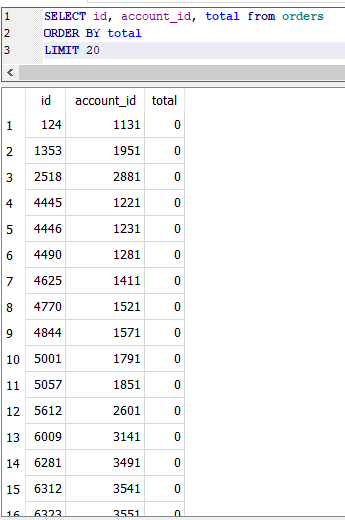
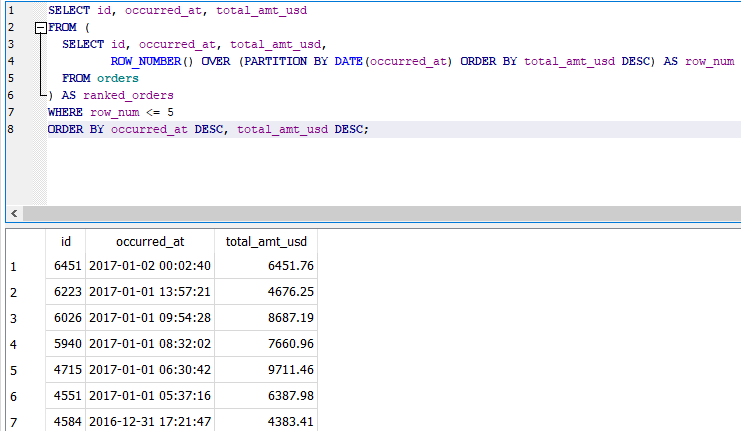
**Department: Software**

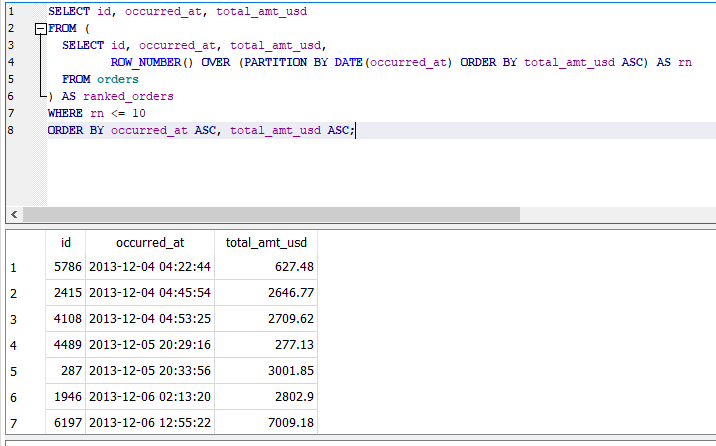
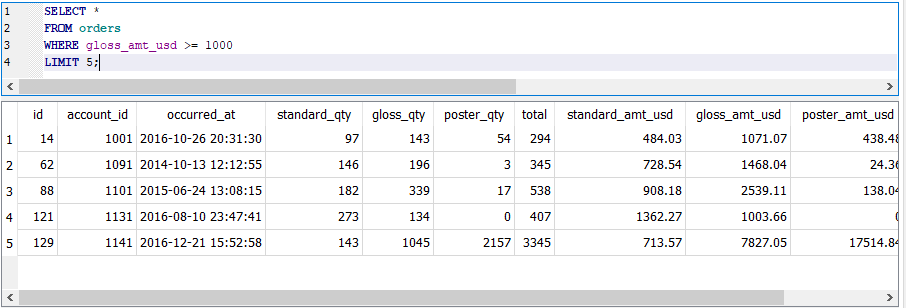
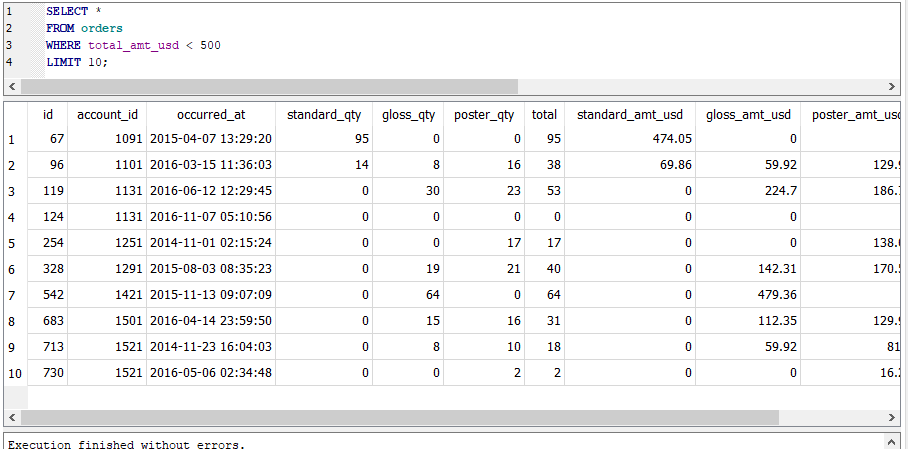
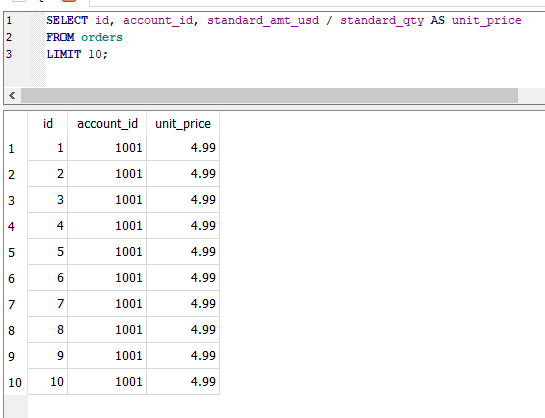
**Subject: Data Science & Analytics**

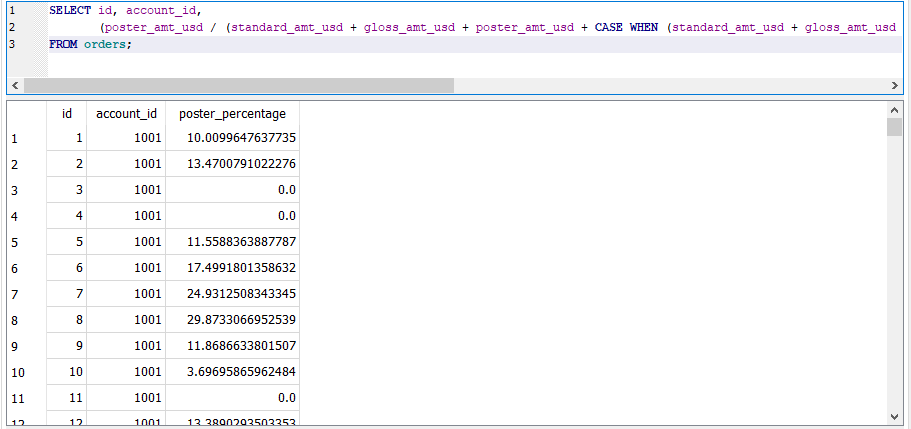
**Submitted To: Madam Rafia Sheikh**

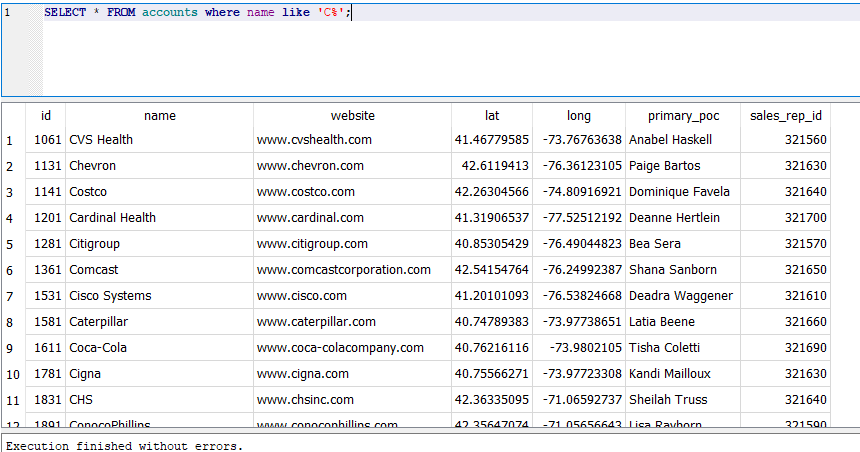
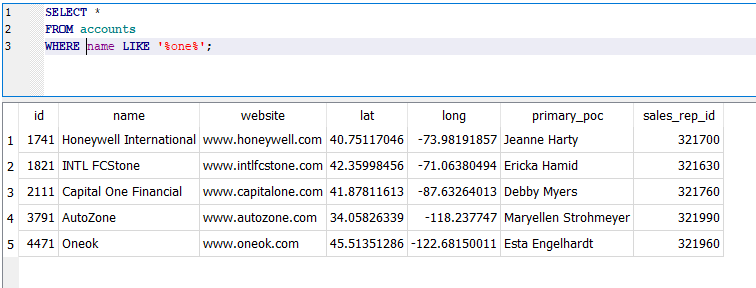
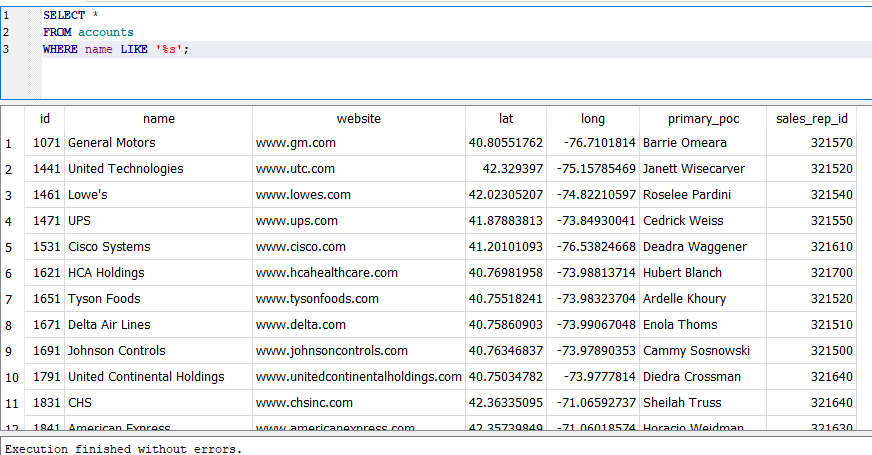
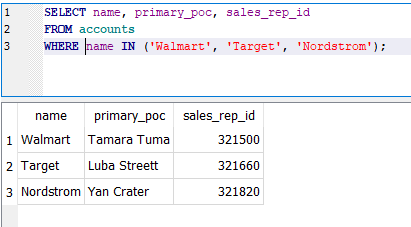
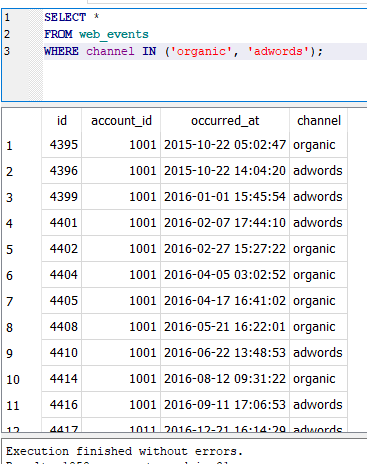
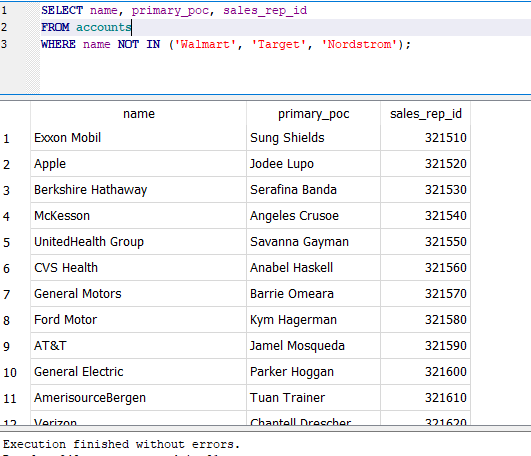
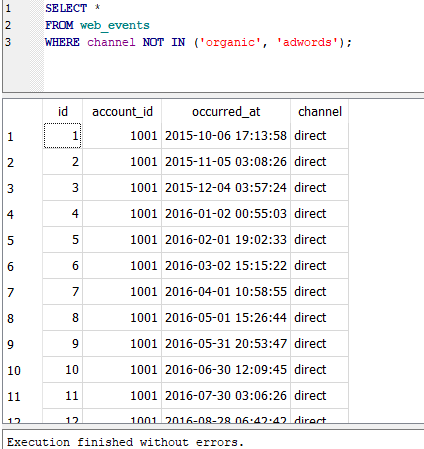
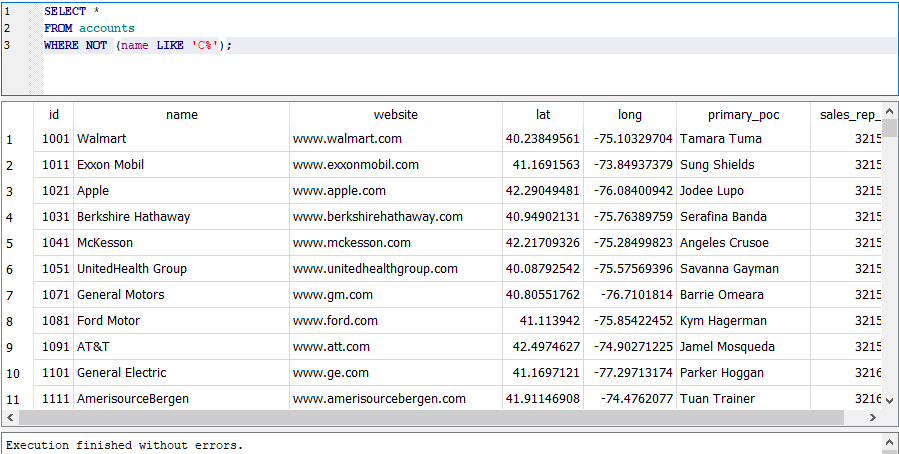
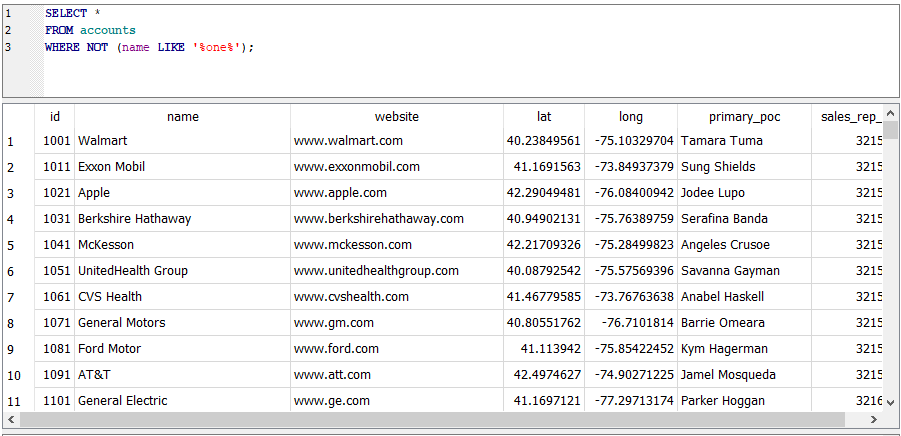
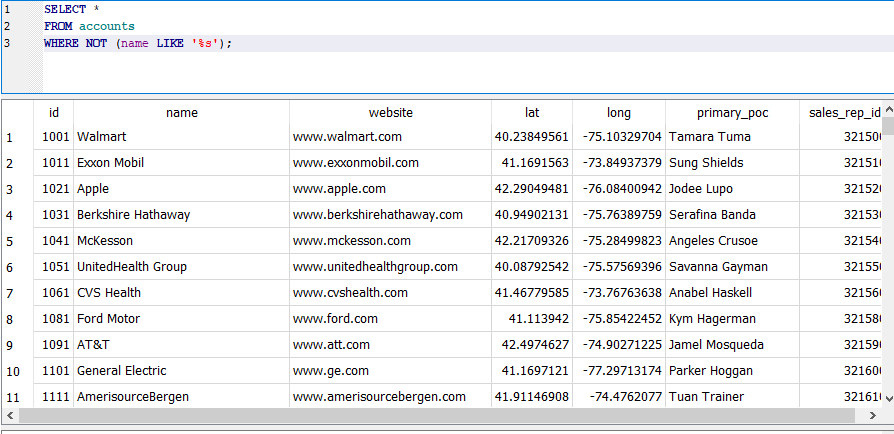
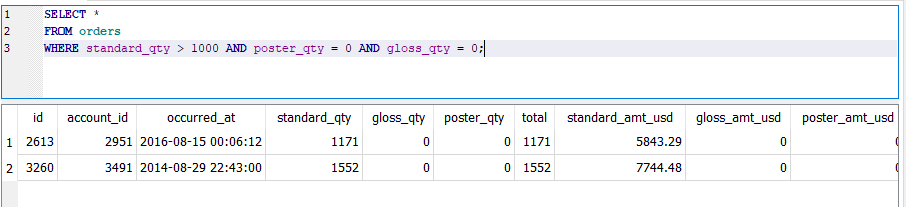
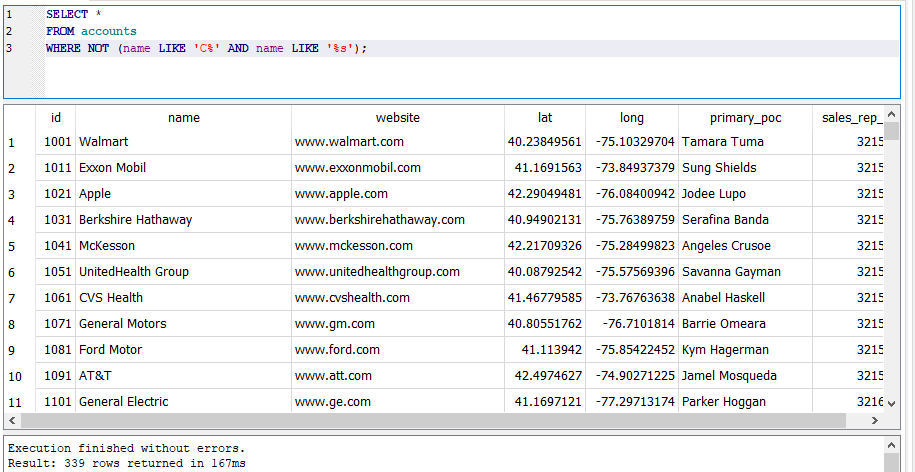
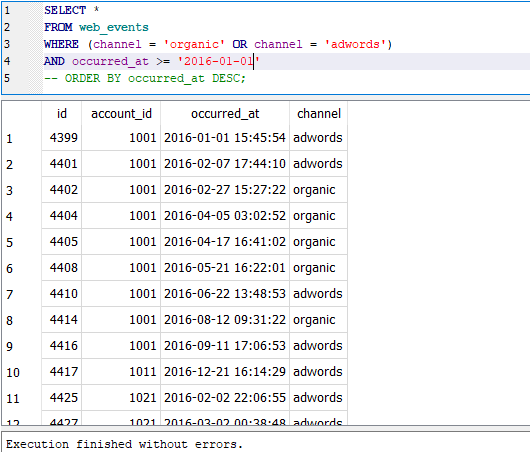
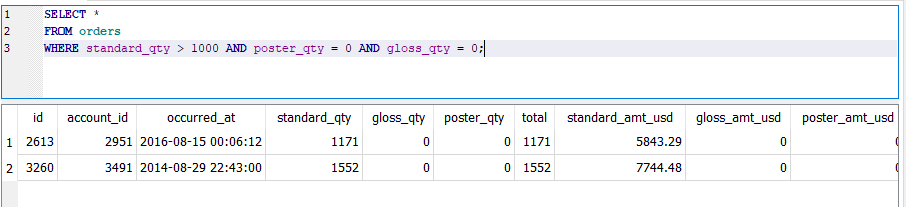
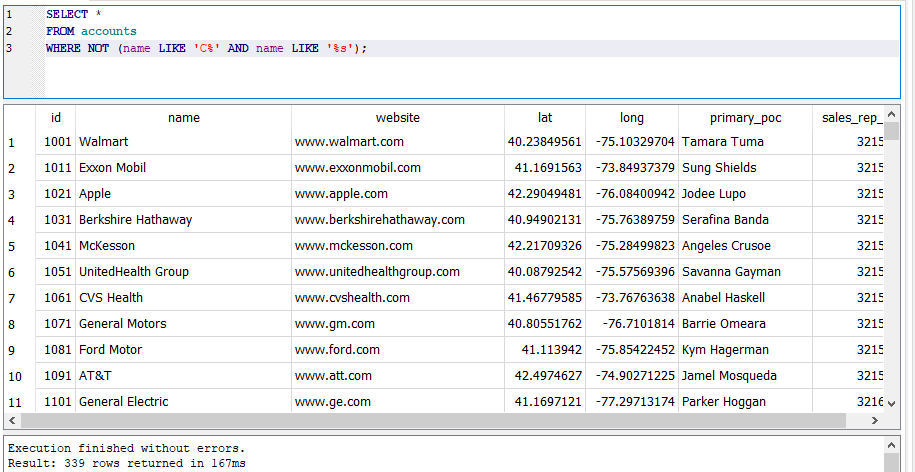
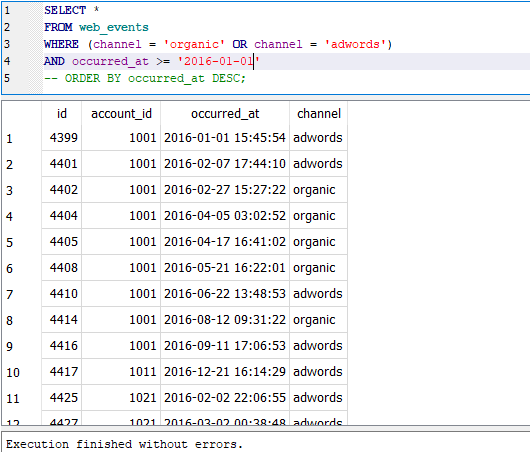
1. Try writing your own query to select only the id, account\_id, and occurred\_at columns for all orders in the orders table.

****

1. Write a query to return the 10 earliest orders in the orders table. Include the id, occurred\_at, and total\_amt\_usd.  
   
2. Write a query to return the top 5 orders in terms of largest total\_amt\_usd. Include the id, account\_id, and total\_amt\_usd.  
   
3. Write a query to return the bottom 20 orders in terms of least total. Include the id, account\_id, and total.  
   
4. Write a query that returns the top 5 rows from orders ordered according to newest to oldest, but with the largest total\_amt\_usd for each date listed first for each date. You will notice each of these dates shows up as unique because of the time element. When you learn about truncating dates in a later lesson, you will better be able to tackle this question on a day, month, or yearly basis.  
   

1. Write a query that returns the top 10 rows from orders ordered according to oldest to newest, but with the smallest total\_amt\_usd for each date listed first for each date. You will notice each of these dates shows up as unique because of the time element. When you learn about truncating dates in a later lesson, you will better be able to tackle this question on a day, month, or yearly basis.  
   
2. Write a query that Pull the first 5 rows and all columns from the orders table that have a dollar amount of gloss\_amt\_usd greater than or equal to 1000.  
   
3. Write a query that Pull the first 10 rows and all columns from the orders table that have a total\_amt\_usd less than 500.  
   
4. Using the orders table, Create a column that divides the standard\_amt\_usd by the standard\_qty to find the unit price for standard paper for each order. Limit the results to the first 10 orders, and include the id and account\_id fields.  
   
5. Using the orders table , Write a query that finds the percentage of revenue that comes from poster paper for each order. You will need to use only the columns that end with \_usd. (Try to do this without using the total column). Include the id and account\_id fields. NOTE - you will be thrown an error with the correct solution to this question. This is for a division by zero. You will learn how to get a solution without an error to this query when you learn about CASE statements in a later section. For now, you might just add some very small value to your denominator as a work around.



1. Use the accounts table to find, All the companies whose names start with 'C'.  
   
2. Use the accounts table to find, All companies whose names contain the string 'one' somewhere in the name.  
   
3. Use the accounts table to find, All companies whose names end with 's'.  
   
4. Use the accounts table to find the account name, primary\_poc, and sales\_rep\_id for Walmart, Target, and Nordstrom.  
   
5. Use the web\_events table to find all information regarding individuals who were contacted via the channel of organic or adwords.  
   
6. We can pull all of the rows that were excluded from the queries in the previous two concepts with our new operator.
   1. Use the accounts table to find the account name, primary poc, and sales rep id for all stores except Walmart, Target, and Nordstrom. 
   2. Use the web\_events table to find all information regarding individuals who were contacted via any method except using organic or adwords methods.  
        
      
7. Use the accounts table to find:
   1. All the companies whose names do not start with 'C'.  
      
   2. All companies whose names do not contain the string 'one' somewhere in the name.  
      
   3. All companies whose names do not end with 's'.  
      
8. Write a query that returns all the orders where the standard\_qty is over 1000, the poster\_qty is 0, and the gloss\_qty is 0.  
   
9. Using the accounts table find all the companies whose names do not start with 'C' and end with 's'.  
   
10. Use the web\_events table to find all information regarding individuals who were contacted via organic or adwords and started their account at any point in 2016 sorted from newest to oldest.  
    
11. Write a query that returns all the orders where the standard\_qty is over 1000, the poster\_qty is 0, and the gloss\_qty is 0.  
    
12. Using the accounts table find all the companies whose names do not start with 'C' and end with 's'.  
    
13. Use the web\_events table to find all information regarding individuals who were contacted via organic or adwords and started their account at any point in 2016 sorted from newest to oldest.  
    

**THE END**