**PRACTICE QUESTIONS FROM UDACITY**

* **JOINS**

1. Provide a table for all **web\_events** associated with the account **name** of Walmart. There should be three columns. Be sure to include the primary\_poc, time of the event, and the channel for each event. Additionally, you might choose to add a fourth column to assure only Walmart events were chosen.
2. Provide a table that provides the **region** for each **sales\_rep** along with their associated **accounts**. Your final table should include three columns: the region **name**, the sales rep **name**, and the account **name**. Sort the accounts alphabetically (A-Z) according to the account name.
3. Provide the **name** for each region for every **order**, as well as the account **name** and the **unit price** they paid (total\_amt\_usd/total) for the order. Your final table should have 3 columns: **region name**, **account name**, and **unit price**. A few accounts have 0 for **total**, so I divided by (total + 0.01) to assure not dividing by zero.
4. Provide a table that provides the **region** for each **sales\_rep** along with their associated **accounts**. This time only for the Midwest region. Your final table should include three columns: the region **name**, the sales rep **name**, and the account **name**. Sort the accounts alphabetically (A-Z) according to the account name.
5. Provide a table that provides the **region** for each **sales\_rep** along with their associated **accounts**. This time only for accounts where the sales rep has a first name starting with S and in the Midwest region. Your final table should include three columns: the region **name**, the sales rep **name**, and the account **name**. Sort the accounts alphabetically (A-Z) according to the account name.
6. Provide a table that provides the **region** for each **sales\_rep** along with their associated **accounts**. This time only for accounts where the sales rep has a **last** name starting with K and in the Midwest region. Your final table should include three columns: the region **name**, the sales rep **name**, and the account **name**. Sort the accounts alphabetically (A-Z) according to the account name.
7. Provide the **name** for each region for every **order**, as well as the account **name** and the **unit price** they paid (total\_amt\_usd/total) for the order. However, you should only provide the results if the **standard order quantity** exceeds 100. Your final table should have 3 columns: **region name**, **account name**, and **unit price**. In order to avoid a division by zero error, adding .01 to the denominator here is helpful total\_amt\_usd/(total+0.01).
8. Provide the **name** for each region for every **order**, as well as the account **name** and the **unit price** they paid (total\_amt\_usd/total) for the order. However, you should only provide the results if the **standard order quantity** exceeds 100 and the **poster order quantity** exceeds 50. Your final table should have 3 columns: **region name**, **account name**, and **unit price**. Sort for the smallest **unit price** first. In order to avoid a division by zero error, adding .01 to the denominator here is helpful (total\_amt\_usd/(total+0.01).
9. What are the different **channel**s used by **account id** 1001? Your final table should have only 2 columns: **account name** and the different **channel**s. You can try **SELECT DISTINCT** to narrow down the results to only the unique values.
10. Find all the orders that occurred in 2015. Your final table should have 4 columns: **occurred\_at**, **account name**, **order total**, and **order total\_amt\_usd**.

* **AGGREGATIONS**

1. Find the total amount of **poster\_qty** paper ordered in the **orders** table.
2. Find the total amount of **standard\_qty** paper ordered in the **orders** table.
3. Find the total dollar amount of sales using the **total\_amt\_usd** in the **orders** table.
4. Find the total amount spent on **standard\_amt\_usd** and **gloss\_amt\_usd** paper for each order in the orders table. This should give a dollar amount for each order in the table.
5. Find the **standard\_amt\_usd** per unit of **standard\_qty** paper. Your solution should use both aggregation and a mathematical operator. -----SUM
6. When was the earliest order ever placed? You only need to return the date.
7. Try performing the same query as in question 1 without using an aggregation function.
8. When did the most recent (latest) **web\_event** occur?
9. Try to perform the result of the previous query without using an aggregation function.
10. Find the mean (**AVERAGE**) amount spent per order on each paper type, as well as the mean amount of each paper type purchased per order. Your final answer should have 6 values - one for each paper type for the average number of sales, as well as the average amount.
11. Via the video, you might be interested in how to calculate the MEDIAN. Though this is more advanced than what we have covered so far try finding - what is the MEDIAN **total\_usd** spent on all **orders**?
12. Which **account** (by name) placed the earliest order? Your solution should have the **account name** and the **date** of the order.
13. Find the total sales in **usd** for each account. You should include two columns - the total sales for each company's orders in **usd** and the company **name**.
14. Via what **channel** did the most recent (latest) **web\_event** occur, which **account** was associated with this **web\_event**? Your query should return only three values - the **date**, **channel**, and **account name**.
15. Find the total number of times each type of **channel** from the **web\_events** was used. Your final table should have two columns - the **channel** and the number of times the channel was used.
16. Who was the **primary contact** associated with the earliest **web\_event**?
17. What was the smallest order placed by each **account** in terms of **total usd**. Provide only two columns - the account **name** and the **total usd**. Order from smallest dollar amounts to largest.
18. Find the number of **sales reps** in each region. Your final table should have two columns - the **region** and the number of **sales\_reps**. Order from the fewest reps to most reps.
19. For each account, determine the average amount of each type of paper they purchased across their orders. Your result should have four columns - one for the account **name** and one for the average quantity purchased for each of the paper types for each account.
20. For each account, determine the average amount spent per order on each paper type. Your result should have four columns - one for the account **name** and one for the average amount spent on each paper type.
21. Determine the number of times a particular **channel** was used in the **web\_events** table for each **sales rep**. Your final table should have three columns - the **name of the sales rep**, the **channel**, and the number of occurrences. Order your table with the highest number of occurrences first.
22. Determine the number of times a particular **channel** was used in the **web\_events** table for each **region**. Your final table should have three columns - the **region name**, the **channel**, and the number of occurrences. Order your table with the highest number of occurrences first.
23. Use **DISTINCT** to test if there are any accounts associated with more than one region.
24. Have any **sales reps** worked on more than one account?
25. How many of the **sales reps** have more than 5 accounts that they manage?
26. How many **accounts** have more than 20 orders?
27. Which account has the most orders?
28. Which accounts spent more than 30,000 usd total across all orders?
29. Which accounts spent less than 1,000 usd total across all orders?
30. Which account has spent the most with us?
31. Which account has spent the least with us?
32. Which accounts used facebook as a **channel** to contact customers more than 6 times?
33. Which account used facebook most as a **channel**?
34. Which channel was most frequently used by most accounts?
35. Find the sales in terms of total dollars for all orders in each year, ordered from greatest to least. Do you notice any trends in the yearly sales totals?
36. Which **month** did Parch & Posey have the greatest sales in terms of total dollars? Are all months evenly represented by the dataset?
37. Which **year** did Parch & Posey have the greatest sales in terms of the total number of orders? Are all years evenly represented by the dataset?
38. Which **month** did Parch & Posey have the greatest sales in terms of the total number of orders? Are all months evenly represented by the dataset?
39. In which **month** of which **year** did Walmart spend the most on gloss paper in terms of dollars?
40. Write a query to display for each order, the account ID, the total amount of the order, and the level of the order - ‘Large’ or ’Small’ - depending on if the order is $3000 or more, or smaller than $3000.
41. Write a query to display the number of orders in each of three categories, based on the total number of items in each order. The three categories are: 'At Least 2000', 'Between 1000 and 2000' and 'Less than 1000'.
42. We would like to understand 3 different levels of customers based on the amount associated with their purchases. The top-level includes anyone with a Lifetime Value (total sales of all orders) greater than 200,000 usd. The second level is between 200,000 and 100,000 usd. The lowest level is anyone under 100,000 usd. Provide a table that includes the **level** associated with each **account**. You should provide the **account name**, the **total sales of all orders** for the customer, and the **level**. Order with the top spending customers listed first.
43. We would now like to perform a similar calculation to the first, but we want to obtain the total amount spent by customers only in 2016 and 2017. Keep the same **level**s as in the previous question. Order with the top spending customers listed first.
44. We would like to identify top-performing **sales reps**, which are sales reps associated with more than 200 orders. Create a table with the **sales rep name**, the total number of orders, and a column with top or not depending on if they have more than 200 orders. Place the top salespeople first in your final table.
45. The previous didn't account for the middle, nor the dollar amount associated with the sales. Management decides they want to see these characteristics represented as well. We would like to identify top-performing **sales reps**, which are sales reps associated with more than 200 orders or more than 750000 in total sales. The middle group has any **rep** with more than 150 orders or 500000 in sales. Create a table with the **sales rep name**, the total number of orders, total sales across all orders, and a column with top, middle, or low depending on these criteria. Place the top salespeople based on the dollar amount of sales first in your final table. You might see a few upset salespeople by this criteria!

* **SUBQUERIES & TEMPORARY TABLES**

1. Provide the **name** of the **sales\_rep** in each **region** with the largest amount of **total\_amt\_usd** sales.
2. For the region with the largest (sum) of sales **total\_amt\_usd**, how many **total** (count) orders were placed?
3. **How many accounts** had more **total** purchases than the account **name** which has bought the most **standard\_qty** paper throughout their lifetime as a customer?
4. For the customer that spent the most (in total over their lifetime as a customer) **total\_amt\_usd**, how many **web\_events** did they have for each channel?
5. What is the lifetime average amount spent in terms of **total\_amt\_usd** for the top 10 total spending **accounts**?
6. What is the lifetime average amount spent in terms of **total\_amt\_usd**, including only the companies that spent more per order, on average, than the average of all orders?
7. Provide the **name** of the **sales\_rep** in each **region** with the largest amount of **total\_amt\_usd** sales.
8. For the region with the largest sales **total\_amt\_usd**, how many **total** orders were placed?
9. **How many accounts** had more **total** purchases than the account **name** which has bought the most **standard\_qty** paper throughout their lifetime as a customer?
10. For the customer that spent the most (in total over their lifetime as a customer) **total\_amt\_usd**, how many **web\_events** did they have for each channel?
11. What is the lifetime average amount spent in terms of **total\_amt\_usd** for the top 10 total spending **accounts**?
12. What is the lifetime average amount spent in terms of **total\_amt\_usd**, including only the companies that spent more per order, on average, than the average of all orders.

* **DATA CLEANING**

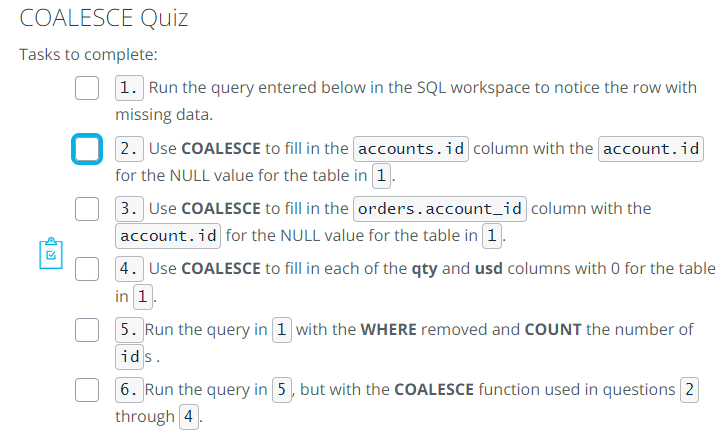
**LEFT & RIGHT Quizzes**

1. In the **accounts** table, there is a column holding the **website** for each company. The last three digits specify what type of web address they are using. A list of extensions (and pricing) is provided [here](https://iwantmyname.com/domains/domain-name-registration-list-of-extensions). Pull these extensions and provide how many of each website type exist in the **accounts** table.
2. There is much debate about how much the name [(or even the first letter of a company name)](https://www.quora.com/Does-a-companys-name-matter) matters. Use the **accounts** table to pull the first letter of each company name to see the distribution of company names that begin with each letter (or number).
3. Use the **accounts** table and a **CASE** statement to create two groups: one group of company names that start with a number and the second group of those company names that start with a letter. What proportion of company names start with a letter?
4. Consider vowels as a, e, i, o, and u. What proportion of company names start with a vowel, and what percent start with anything else?

**CONCAT, LEFT, RIGHT, and SUBSTR**

1. Suppose the company wants to assess the performance of all the sales representatives. Each sales representative is assigned to work in a particular region. To make it easier to understand for the HR team, display the concatenated sales\_reps.id, ‘\_’ (underscore), and region.name as EMP\_ID\_REGION for each sales representative.
2. From the accounts table, display the name of the client, the coordinate as concatenated (latitude, longitude), email id of the primary point of contact as <first letter of the primary\_poc><last letter of the primary\_poc>@<extracted name and domain from the website>.
3. From the web\_events table, display the concatenated value of account\_id, '\_' , channel, '\_', count of web events of the particular channel.
4. Use the accounts table to create **first** and **last** name columns that hold the first and last names for the primary\_poc.
5. Now see if you can do the same thing for every rep name in the sales\_reps table. Again provide **first** and **last** name columns.

**CONCAT**

1. Each company in the accounts table wants to create an email address for each primary\_poc. The email address should be the first name of the **primary\_poc** . last name **primary\_poc** @ company name .com.
2. You may have noticed that in the previous solution some of the company names include spaces, which will certainly not work in an email address. See if you can create an email address that will work by removing all of the spaces in the account name, but otherwise, your solution should be just as in question 1. Some helpful documentation is [here](https://www.postgresql.org/docs/8.1/static/functions-string.html).
3. We would also like to create an initial password, which they will change after their first log in. The first password will be the first letter of the primary\_poc's first name (lowercase), then the last letter of their first name (lowercase), the first letter of their last name (lowercase), the last letter of their last name (lowercase), the number of letters in their first name, the number of letters in their last name, and then the name of the company they are working with, all capitalized with no spaces.
4. ****

* **COALESCE**

1. Create a running total of standard\_amt\_usd (in the orders table) over order time with no date truncation. Your final table should have two columns: one with the amount being added for each new row, and a second with the running total.
2. Creating a Partitioned Running Total Using Window Functions

Now, modify your query from the previous quiz to include partitions. Still create a running total of standard\_amt\_usd (in the orders table) over order time, but this time, date truncate occurred\_at by year and partition by that same year-truncated occurred\_at variable.

Your final table should have three columns:

* One with the amount being added for each row
* One for the truncated date,
* A final column with the running total within each year

1. [**https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0005/concepts/8f45cae5-b384-4f18-ae8e-5846c6408a86**](https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0005/concepts/8f45cae5-b384-4f18-ae8e-5846c6408a86) **(FOR THE NEXT PART)**
2. Select the id, account\_id, and total variable from the orders table, then create a column called total\_rank that ranks this total amount of paper ordered (from highest to lowest) for each account using a partition. Your final table should have these four columns.
3. [**https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0005/concepts/607f25b0-f63c-48f7-a273-54d8afcd347d**](https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0005/concepts/607f25b0-f63c-48f7-a273-54d8afcd347d) **(WORK ON THE PREVIOUS WORKS BEFORE ANSWERING)**
4. Use the NTILE functionality to divide the accounts into 4 levels in terms of the amount of standard\_qty for their orders. Your resulting table should have the account\_id, the occurred\_at time for each order, the total amount of standard\_qty paper purchased, and one of four levels in a standard\_quartile column.
5. Use the NTILE functionality to divide the accounts into two levels in terms of the amount of gloss\_qty for their orders. Your resulting table should have the account\_id, the occurred\_at time for each order, the total amount of gloss\_qty paper purchased, and one of two levels in a gloss\_half column.
6. Use the NTILE functionality to divide the orders for each account into 100 levels in terms of the amount of total\_amt\_usd for their orders. Your resulting table should have the account\_id, the occurred\_at time for each order, the total amount of total\_amt\_usd paper purchased, and one of 100 levels in a total\_percentile column. **Note:** To make it easier to interpret the results, order by the account\_id in each of the queries.

* **ADVANCED JOINS AND PERFORMANCE TUNING**

[**https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0006/concepts/40c7a3a9-6392-4ad3-87a7-3c49da5acc88**](https://learn.udacity.com/nanodegrees/nd104-ent-bmann/parts/10c86eb0-eeca-4d6c-aeaf-b5fdd6562d6f/lessons/ls0006/concepts/40c7a3a9-6392-4ad3-87a7-3c49da5acc88)

**(USE THIS)**