Day 13 – Assignment 1 – 27 October 2023

1. Student ID
2. Char (1)
3. Big Query - Datawarehouse
4. Foreign key
5. Analytics and reporting (Data warehouses are OLAP)

Day 14 - Assignment 2 – 30 October 2023

1. Choose the correct order in which the following clauses are arranged in a SQL query?

C select > from > orderby > limit

1. Write a query to get all the products available in the market, and arrange them in ascending order with respect to their product\_id

select product\_id, product\_name from `farmers\_market.product` order by product\_id;

1. Write a query to calculate the salary of all employees after an increment of 20%. Save the newly calculated salary column as 'New\_salary'.

select emp\_id, name, salary, salary+(salary\*0.2) as New\_salary from `farmers\_market.Employee` order by emp\_id;

Day 15 – Assignment 3 – 31 October 2023

1. Write a query to Find the booth assignments for vendor\_id 7 for all dates between April 3, 2019 and May 16, 2019, (including the 2 dates). (farmers\_market dataset)

select \* from `farmers\_market.vendor\_booth\_assignments` where vendor\_id = 7 and market\_date between '2019-04-03' and '2019-05-16';

1. Write a query to get data about a customer you knew as “Jer” but you are not sure if they are listed as Jeremy or Jeremiah or Jerry. (farmers\_market dataset)

select \* from `farmers\_market.customer` where customer\_first\_name like 'Jer%';

1. Print a report of everything the customer\_id 4 has ever purchased at the market, sorted by date. (farmers\_market dataset)

select \* from `farmers\_market.customer\_purchases` where customer\_id = 4 order by market\_date;+

1. Write a query to find all the details of the product that has the third-highest revenue.

select \* from 'Supermarket.Products' order by Revenue offset 2 limit 2;

# Assignmen 8 Question 1

#Write a Query find all the market dates for which the total sales is are more that $150

select market\_date,sum(quantity\*cost\_to\_customer\_per\_qty) as Sales

from `farmers\_market.customer\_purchases`

group by market\_date

having sum(quantity\*cost\_to\_customer\_per\_qty) > 150 order by sales; # Order by is taken for clear understanding

# Assignment 8 Question 2

# Write a Query to get the Vendor ID for which the total sales is between $100 to $150 for every market date

select market\_date,vendor\_id,sum(quantity\*cost\_to\_customer\_per\_qty) as sales

from `farmers\_market.customer\_purchases`

group by market\_date,vendor\_id

having sum(quantity\*cost\_to\_customer\_per\_qty) between 100 and 150;

# Assignment 8 Question 3

# Write a query to find all the products which have a minimum of 20 quantities sold on each market date

select market\_date,product\_id from `farmers\_market.customer\_purchases`

group by market\_date,product\_id

having min(quantity) = 20;

Assignment 8 Errors Debugging

Error 1



When we forget to write the correct name of a column or missed the comma’s.

Error 2



This error causes when we did not write the correct database, tables names which are present in warehouse.

Error 3



This error causes when we mentioned the column which is not grouped or we used aggregations without group by.

Error 4



This error causes when we try to aggregate in where clause.

Error 5



This error causes when we did not follow the order of syntax.

Error 6



This error causes when we did not write the correct datatype of that column.

Error 7



This error causes when we are writing multiple queries in one page, we have to separate them with ‘; ‘

Error 8



This error causes when we did not referred the column name correctly when joining the tables

Error 9



This error occurs when we did not specify how we should join the tables

Error 10

A black and blue text

Description automatically generated

This is not an error but after execution of the query it will throw error.

Error 11



This error occurs when we try to aggregate the non-numeric column

Error 12



This error occurs when we did not close the parenthesis or quotes correctly.

Assignment – 9 – 09 October 2023

# Assignment 9 Question 1

# Here in this query we have write the over after the rank function ( rank() over ) and by is missing after partition which is not the right syntax

SELECT

vendor\_id,

market\_date,

product\_id,

Original\_price,

RANK() OVER (PARTITION by vendor\_id ORDER BY original\_price DESC)

AS price\_rank

FROM `farmers\_market.vendor\_inventory`;

# Assignment 9 Question 2

# In this query there are three error .

# 1. DENSE RANK() does not have the right syntax should DENSE\_RANK().

# 2. The database name is not mentioned before the table name, it should be `farmers\_markets.vendor\_inventory` .

# 3. While selecting the product\_Id from the table, the product\_Id is misspelled it should be product\_id .

# 4. There is no comma between market\_date and product\_id which will throw a error.

SELECT

vendor\_id, market\_date, product\_id, original\_price,

DENSE\_RANK() OVER (PARTITION BY product\_id ORDER BY original\_price

DESC) AS price\_rank

FROM `farmers\_market.vendor\_inventory`;

# Assignment 9 Question 3

# Complete the below query to get the desired result (Total\_amt paid by customer\_id 4 & 5 per market\_date )

SELECT

market\_date, customer\_id,

sum(quantity \* cost\_to\_customer\_per\_qty) AS total\_amt\_spent

FROM `farmers\_market.customer\_purchases`

where customer\_id = 4 or customer\_id = 5

group by market\_date,customer\_id

ORDER BY market\_date;

# Assignment 9 Question 4

# Complete the below query to get the desired result (List down all the product details where product category contains “Fruits” in it.)

SELECT \*

FROM `farmers\_market.product`

WHERE product\_category\_id IN ( select  product\_category\_id

FROM `farmers\_market.product\_category`

WHERE lower(product\_category\_name) like '%fruits%');

# Assignment 9 Question 5

# Write a Query to Rank all the products bases on their Quantities sold per market date (Use all Rank(), Dense\_Rank() & Row numbers() functions in your query)

select

market\_date,

quantity,

rank() over (partition by market\_date order by quantity) as rank,

dense\_rank() over (partition by market\_date order by quantity) as dense\_rank,

row\_number() over (partition by market\_date order by quantity) as row\_number

from `farmers\_market.customer\_purchases`;