**Sales Performance Analysis at RevoGrocers SQL Query**

-- Identify the product category that generates the highest revenue.

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(SUM(sales.Quantity \* (1-sales.Discount) \* products.price),2) AS revenue

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

ORDER BY 3 DESC;

-- Assess the correlation between revenue and total units sold for each product category.

WITH categories\_sales AS(

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(SUM(sales.Quantity \* (1-sales.Discount) \* products.price),2) AS revenue

,SUM(sales.Quantity) AS total\_units\_sold

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

ORDER BY 3 DESC, 4 DESC

)

SELECT

ROUND(CORR (revenue,total\_units\_sold ),2) AS correlation\_sales\_units\_sold

FROM categories\_sales;

-- Find the correlation between revenue and the number of unique customers for each product category.

WITH categories\_sales\_2 AS(

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(SUM(sales.Quantity \* (1-sales.Discount) \* products.price),2) AS revenue

,COUNT(DISTINCT sales.CustomerID) AS number\_of\_customer

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

ORDER BY 2 DESC, 1 DESC

)

SELECT

ROUND(CORR(revenue, number\_of\_customer),2) AS correlation\_rev\_cust

FROM categories\_sales\_2;

--Calculate the average price per unit for each product category.

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(AVG(products.price),2) AS avg\_price

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

ORDER BY 3 DESC;

-- Evaluate the correlation between the average price per unit and the number of buyers (unique customers) per category.

WITH avg\_price\_count\_cust AS(

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(AVG(products.price),2) AS avg\_price

,COUNT(DISTINCT sales.CustomerID) AS number\_of\_customer

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

)

SELECT

ROUND(CORR(avg\_price,number\_of\_customer),2) AS corr\_avg\_price\_count\_cust

FROM avg\_price\_count\_cust;

-- Which categories contribute the most to overall revenue (percentage-wise)?

WITH contribute\_revenue AS(

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,ROUND(100 \* SUM(sales.Quantity \* (1 - sales.Discount) \* products.price)

/SUM(SUM(sales.Quantity \* (1 - sales.Discount) \* products.price))OVER(), 2) AS revenue\_cont\_percentage

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

)

SELECT

category\_id

,category\_name

,revenue\_cont\_percentage

,RANK () OVER (ORDER BY revenue\_cont\_percentage DESC) AS rank

FROM contribute\_revenue

ORDER BY 4 ASC;

-- COUNT repeat cust id

-- Which product categories have the highest repeat purchase rate?

WITH repeat\_purchase AS (

SELECT

categories.categoryid AS category\_id

,categories.categoryname AS category\_name

,sales.CustomerID AS customer\_id

,COUNT(sales.CustomerID) AS repeat\_order

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories`AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2,3

HAVING COUNT(sales.CustomerID) > 1

ORDER BY 1

),

customer\_repeat\_order AS(

SELECT

category\_id

,category\_name

,COUNT(repeat\_order) AS number\_of\_customer\_repeat\_order

FROM repeat\_purchase

GROUP BY 1,2

ORDER BY number\_of\_customer\_repeat\_order DESC

),

total\_customers\_per\_category AS (

SELECT

categories.categoryid AS category\_id,

categories.categoryname AS category\_name,

COUNT(DISTINCT sales.CustomerID) AS total\_customers

FROM `fsda-sql-01.grocery\_dataset.sales` AS sales

INNER JOIN `fsda-sql-01.grocery\_dataset.products` AS products

ON sales.ProductID = products.productid

INNER JOIN `fsda-sql-01.grocery\_dataset.categories` AS categories

ON products.categoryid = categories.categoryid

GROUP BY 1,2

)

SELECT

total\_cust\_per\_category.category\_id

,total\_cust\_per\_category.category\_name

,cust\_repeat\_ord.number\_of\_customer\_repeat\_order

,total\_cust\_per\_category.total\_customers

,ROUND((100\*number\_of\_customer\_repeat\_order/ total\_customers),2) AS repeat\_purchase\_rate

FROM customer\_repeat\_order AS cust\_repeat\_ord

INNER JOIN total\_customers\_per\_category AS total\_cust\_per\_category

ON cust\_repeat\_ord.category\_id = total\_cust\_per\_category.category\_id

INNER JOIN `fsda-sql-01.grocery\_dataset.categories` AS categories

ON categories.categoryid = cust\_repeat\_ord.category\_id

ORDER BY 5 DESC;

-- find null sales

select \*

from fsda-sql-01.grocery\_dataset.sales

WHERE CustomerID is null;

-- find null products

select \*

from fsda-sql-01.grocery\_dataset.products

WHERE categoryid is null;

-- Which product categories have the highest repeat purchase rate?

-- count cust id who order > 1 / count unique cust id \* 100 per category

WITH orders AS(

SELECT

products.categoryid AS category\_id

,categories.categoryname AS category\_name

,sales.CustomerID AS cust\_id

FROM fsda-sql-01.grocery\_dataset.sales AS sales

INNER JOIN fsda-sql-01.grocery\_dataset.products AS products

ON sales.ProductID = products.productid

INNER JOIN fsda-sql-01.grocery\_dataset.categories AS categories

ON products.categoryid = categories.categoryid

),

repeat\_order AS (

SELECT

category\_id

,category\_name

,cust\_id

,COUNT(cust\_id) AS number\_repeat\_order

FROM orders

GROUP BY 1,2,3

HAVING COUNT(cust\_id) > 1

),

repeat\_customer AS (

SELECT

category\_id

,category\_name

,count (cust\_id) AS number\_repeat\_customer

FROM repeat\_order

GROUP BY 1,2

)

SELECT

orders.category\_id

,orders.category\_name

,repeat\_customer.number\_repeat\_customer

,COUNT (DISTINCT orders.cust\_id) AS total\_unique\_customer

,ROUND(100\* repeat\_customer.number\_repeat\_customer/COUNT (DISTINCT orders.cust\_id),2) AS repeat\_purchase\_rate

FROM orders

INNER JOIN repeat\_customer

ON orders.category\_id = repeat\_customer.category\_id

GROUP BY 1,2,3

ORDER BY 5 DESC;