Basic Introduction to Dataset: SQL Queries

USE SALES;

-- CUSTOMERS table queries --

-- To view the structure of CUSTOMERS table and the first record

SELECT \* FROM SALES.CUSTOMERS LIMIT 1;

-- Count total number of customers based on CUSTOMER\_CODE

SELECT COUNT(CUSTOMER\_CODE) AS TOTAL\_CUSTOMERS FROM SALES.CUSTOMERS;

-- Count total number of rows in CUSTOMERS table

SELECT COUNT(\*) AS TOTAL\_NUM\_ROWS FROM SALES.CUSTOMERS;

-- Count distinct customer names in CUSTOMERS table

SELECT COUNT(DISTINCT CUSTMER\_NAME) AS DISTINCT\_NAME\_COUNT FROM SALES.CUSTOMERS;

-- Count occurrences of each customer type

SELECT CUSTOMER\_TYPE, COUNT(\*) AS TYPE\_COUNT

FROM SALES.CUSTOMERS

GROUP BY CUSTOMER\_TYPE;

-- DATE table queries --

-- To view the structure of DATE table and the first record

SELECT \* FROM SALES.DATE LIMIT 1;

-- List distinct years available in DATE table

SELECT DISTINCT YEAR FROM SALES.DATE;

-- MARKETS table queries --

-- To view the structure of MARKETS table and the first record

SELECT \* FROM SALES.MARKETS LIMIT 1;

-- Count total number of market codes in MARKETS table

SELECT COUNT(MARKET\_CODE) FROM SALES.MARKETS;

-- Count total number of rows in MARKETS table

SELECT COUNT(\*) AS TOTAL\_NUM\_ROWS FROM SALES.MARKETS;

-- Count distinct zones in MARKETS table

SELECT COUNT(DISTINCT ZONE) AS DISTINCT\_ZONE\_COUNT FROM SALES.MARKETS;

-- Count occurrences of each zone in MARKETS table

SELECT ZONE, COUNT(\*) AS TYPE\_COUNT

FROM SALES.MARKETS

GROUP BY ZONE;

-- PRODUCTS table queries --

-- To view the structure of PRODUCTS table and the first record

SELECT \* FROM SALES.PRODUCTS LIMIT 1;

-- Count total number of product codes in PRODUCTS table

SELECT COUNT(PRODUCT\_CODE) FROM SALES.PRODUCTS;

-- Count distinct product codes in PRODUCTS table

SELECT COUNT(DISTINCT PRODUCT\_CODE) FROM SALES.PRODUCTS;

-- Count occurrences of each product type in PRODUCTS table

SELECT PRODUCT\_TYPE, COUNT(\*) AS TYPE\_COUNT

FROM SALES.PRODUCTS

GROUP BY PRODUCT\_TYPE;

-- TRANSACTIONS table queries --

-- To view the structure of TRANSACTIONS table and the first record

SELECT \* FROM SALES.TRANSACTIONS LIMIT 1;

-- Count total number of rows in TRANSACTIONS table

SELECT COUNT(\*) FROM SALES.TRANSACTIONS;

-- Missing values analysis for critical columns in TRANSACTIONS table

SELECT 'product\_code' AS ColumnName,

SUM(CASE WHEN product\_code IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN product\_code = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'customer\_code' AS ColumnName,

SUM(CASE WHEN customer\_code IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN customer\_code = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'market\_code' AS ColumnName,

SUM(CASE WHEN market\_code IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN market\_code = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'order\_date' AS ColumnName,

SUM(CASE WHEN order\_date IS NULL THEN 1 ELSE 0 END) AS NullCount,

0 AS EmptyStringCount -- Counting empty strings as 0

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'sales\_qty' AS ColumnName,

SUM(CASE WHEN sales\_qty IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN sales\_qty = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'sales\_amount' AS ColumnName,

SUM(CASE WHEN sales\_amount IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN sales\_amount = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'currency' AS ColumnName,

SUM(CASE WHEN currency IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN currency = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'profit\_margin\_percentage' AS ColumnName,

SUM(CASE WHEN profit\_margin\_percentage IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN profit\_margin\_percentage = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'profit\_margin' AS ColumnName,

SUM(CASE WHEN profit\_margin IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN profit\_margin = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS

UNION ALL

SELECT 'cost\_price' AS ColumnName,

SUM(CASE WHEN cost\_price IS NULL THEN 1 ELSE 0 END) AS NullCount,

SUM(CASE WHEN cost\_price = '' THEN 1 ELSE 0 END) AS EmptyStringCount

FROM SALES.TRANSACTIONS;

-- Data types in TRANSACTIONS table --

SELECT COLUMN\_NAME, DATA\_TYPE

FROM INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_SCHEMA = 'SALES'

AND TABLE\_NAME = 'TRANSACTIONS';

-- Count distinct currencies and their occurrences in TRANSACTIONS table

SELECT COUNT(DISTINCT CURRENCY) AS DISTINCT\_CURRENCY\_COUNT FROM SALES.TRANSACTIONS;

-- Show occurrences of each currency type in TRANSACTIONS table

SELECT CURRENCY, COUNT(\*) AS TYPE\_COUNT

FROM SALES.TRANSACTIONS

GROUP BY CURRENCY;

-- Advanced Queries for Rechecking Dashboard Details --

-- Calculate Total Sales Amount per Customer

SELECT t.customer\_code, c.custmer\_name, SUM(t.sales\_amount) AS total\_sales\_amount

FROM sales.transactions t

INNER JOIN sales.customers c ON t.customer\_code = c.customer\_code

GROUP BY t.customer\_code, c.custmer\_name;

-- Aggregate Sales by Market and Year

SELECT m.markets\_name, d.year, SUM(t.sales\_amount) AS total\_sales\_amount

FROM sales.transactions t

INNER JOIN sales.markets m ON t.market\_code = m.markets\_code

INNER JOIN sales.date d ON t.order\_date = d.date

GROUP BY m.markets\_name, d.year;

-- Create a Report of Total Sales by Year and Month

SELECT d.year, d.month\_name, SUM(t.sales\_amount) AS total\_sales\_amount

FROM sales.transactions t

INNER JOIN sales.date d ON t.order\_date = d.date

GROUP BY d.year, d.month\_name

ORDER BY d.year, d.month\_name;

-- Show transactions for Chennai market (market code for Chennai is 'Mark001')

SELECT \* FROM sales.transactions WHERE market\_code = 'Mark001';

-- Show distinct product codes that were sold in Chennai

SELECT DISTINCT product\_code FROM sales.transactions WHERE market\_code = 'Mark001';

-- Show transactions where currency is US dollars

SELECT \* FROM sales.transactions WHERE currency = 'USD';

-- Show transactions in the year 2020 joined by date table

SELECT transactions.\*, date.\*

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020;

-- Show total revenue in the year 2020

SELECT SUM(transactions.sales\_amount)

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020 AND (transactions.currency = 'INR' OR transactions.currency = 'USD');

-- Show total revenue in January 2020

SELECT SUM(transactions.sales\_amount)

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020 AND date.month\_name = 'January' AND (transactions.currency = 'INR' OR transactions.currency = 'USD');

-- Show total revenue in 2020 in Chennai (market code 'Mark001')

SELECT SUM(transactions.sales\_amount)

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020 AND transactions.market\_code = 'Mark001';

-- Show average sales amount per transaction

SELECT AVG(sales\_amount) AS average\_sales

FROM sales.transactions;

-- Show the top 10 customers by total sales amount

SELECT customer\_code, SUM(sales\_amount) AS total\_sales

FROM sales.transactions

GROUP BY customer\_code

ORDER BY total\_sales DESC

LIMIT 10;

-- Show the product with the highest sales amount in 2020

SELECT product\_code, SUM(sales\_amount) AS total\_sales

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY product\_code

ORDER BY total\_sales DESC

LIMIT 1;

-- Show the market with the highest total revenue in 2020

SELECT market\_code, SUM(sales\_amount) AS total\_sales

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY market\_code

ORDER BY total\_sales DESC

LIMIT 1;

-- Show the number of transactions per customer in 2020

SELECT customer\_code, COUNT(\*) AS transaction\_count

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY customer\_code;

-- Show the total sales amount per market in 2020

SELECT market\_code, SUM(sales\_amount) AS total\_sales

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY market\_code;

-- Show the total number of products sold in each market in 2020

SELECT market\_code, COUNT(DISTINCT product\_code) AS product\_count

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY market\_code;

-- Show the total sales amount for each product category in 2020

SELECT products.category, SUM(transactions.sales\_amount) AS total\_sales

FROM sales.transactions

INNER JOIN sales.products ON transactions.product\_code = products.product\_code

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY products.category;

-- Show the average sales amount per transaction in 2020

SELECT AVG(transactions.sales\_amount) AS average\_sales

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020;

-- Show the total sales amount in each currency for 2020

SELECT transactions.currency, SUM(transactions.sales\_amount) AS total\_sales

FROM sales.transactions

INNER JOIN sales.date ON transactions.order\_date = date.date

WHERE date.year = 2020

GROUP BY transactions.currency;