-- Errol Ian Ave Acosta

-- Data Analysis | SQL

-- CGPT

-- February 16, 2025

-- Note:

-- This template provides SQL queries for data visualization purposes.

-- Adjust table and column names according to your database schema.

-- - Adjust date functions like DATE\_TRUNC based on your SQL dialect (e.g., SQL Server uses DATEPART, PostgreSQL uses DATE\_TRUNC).

-- - Replace 'your\_table\_name', 'category\_column', 'numeric\_column', 'date\_column', etc., with your actual table and column names.

-- - Ensure your SQL environment supports all functions used here, or adjust accordingly.

-- - For complex visualizations, you might need to combine multiple queries or use a BI tool on top of this data.

-- Basic Query for Data Overview

SELECT \*

FROM your\_table\_name

LIMIT 10; -- Show first 10 rows to understand data structure

-- Count of Records by Category

-- Useful for bar charts or pie charts to visualize category distribution

SELECT category\_column, COUNT(\*) as count

FROM your\_table\_name

GROUP BY category\_column

ORDER BY count DESC;

-- Aggregates for Numerical Data

-- For creating histograms or box plots

SELECT

AVG(numeric\_column) as avg\_value,

MIN(numeric\_column) as min\_value,

MAX(numeric\_column) as max\_value,

STDDEV(numeric\_column) as std\_dev

FROM your\_table\_name;

-- Time Series Data

-- For line charts showing trends over time

SELECT

DATE\_TRUNC('day', date\_column) as day,

SUM(numeric\_column) as daily\_sum

FROM your\_table\_name

GROUP BY day

ORDER BY day;

-- Correlation Between Two Numerical Columns

-- Helps in creating scatter plots to visualize relationships

SELECT

column1,

column2

FROM your\_table\_name

WHERE column1 IS NOT NULL AND column2 IS NOT NULL; -- Ensure no NULL values for accurate correlation

-- Distribution of Data Over Time for Categorical Variables

-- Suitable for stacked bar or area charts

SELECT

DATE\_TRUNC('month', date\_column) as month,

category\_column,

COUNT(\*) as monthly\_count

FROM your\_table\_name

GROUP BY month, category\_column

ORDER BY month, monthly\_count DESC;

-- Top N Analysis

-- For creating bar charts to show top contributors

SELECT

entity\_column,

SUM(numeric\_column) as total

FROM your\_table\_name

GROUP BY entity\_column

ORDER BY total DESC

LIMIT 10; -- Adjust '10' to change the number of top entities shown

-- Percentile Analysis

-- Useful for box plots or to understand data spread

WITH percentiles AS (

SELECT

PERCENTILE\_CONT(0.25) WITHIN GROUP (ORDER BY numeric\_column) AS Q1,

PERCENTILE\_CONT(0.5) WITHIN GROUP (ORDER BY numeric\_column) AS Median,

PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY numeric\_column) AS Q3

FROM your\_table\_name

)

SELECT \* FROM percentiles;

-- Cohort Analysis

-- For visualizing user behavior over time

WITH cohort AS (

SELECT

user\_id,

DATE\_TRUNC('month', first\_purchase\_date) as cohort\_month,

DATE\_TRUNC('month', purchase\_date) as purchase\_month,

COUNT(\*) as purchase\_count

FROM your\_table\_name

GROUP BY user\_id, cohort\_month, purchase\_month

)

SELECT

cohort\_month,

DATE\_DIFF(purchase\_month, cohort\_month, MONTH) as months\_since\_cohort,

COUNT(DISTINCT user\_id) as active\_users

FROM cohort

GROUP BY cohort\_month, months\_since\_cohort

ORDER BY cohort\_month, months\_since\_cohort;