**REFERENCES OF SQL QUERIES AND EXPLANATION OF LOGIC USED**

Description of the database schema and tables used.

The table in the database schema represents distinct features of the data, and it contains comprehensive information on different smartphone models. The table is intended to store and manage smartphone data.

Table Name: smartphone\_cleaned

Below is a description of the table and its columns:

COLUMNS DESCRIBED:• **brand\_name (varchar)** : The brand name of the smartphone.(e.g Apple, Samsung)

• **model (varchar)**: The model name or number of the smartphone.

• **price (int)**: The price of the smartphone in RUPEES.

• **rating (float)**: The rating of the smartphone out of 100.

• **has\_5g (Bit)** : Indicates whether the smartphone supports 5G connectivity (TRUE/FALSE).

• **has\_nfc (Bit)**: Indicates whether the smartphone has NFC capability (TRUE/FALSE).

• **has\_ir\_blaster (Bit)**: Indicates whether the smartphone has an IR blaster (TRUE/FALSE).

• **processor\_name (varchar)**: The name of the processor used in the smartphone.

• **processor\_brand (varchar)**: The brand of the processor used in the smartphone.

• **num\_cores (varchar)**: The number of cores in the processor.

• **processor\_speed (float)**: The speed of the processor in GHz.

• **battery\_capacity (int)**: The battery capacity of the smartphone in mAh.

• **fast\_charging (int)**: The fast-charging capability of the smartphone in watts.

• **ram\_capacity (int)**: The RAM capacity of the smartphone in GB.

• **internal\_memory (int)**: The internal storage capacity of the smartphone in GB.

• **refresh\_rate (int)**: The refresh rate of the smartphone displays in Hz.

• **resolution (int)**: The resolution of the smartphone displays.

• **num\_rear\_cameras (int)**: The number of rear cameras on the smartphone.

• **num\_front\_cameras (int)**: The number of front cameras on the smartphone.

•**os (varchar)**: The operating system used in the smartphone.

**primary\_camera\_rear (varchar)**: The resolution of the primary rear camera in megapixels.

• **primary\_camera\_front (varchar)**: The resolution of the primary front camera in megapixels.

• **extended\_memory (bit)**: The capacity of extended memory (e.g., microSD card support) in the smartphone, if applicable

**EXPLANATION OF SQL QUERY LOGIC AND ANY DATA TRANSFORMATIONS PERFORMED.**

**DATA CLEANING**

With thorough information, the data depicts unique characteristics. Excel was used to clean the data efficiently, addressing missing numbers, detecting anomalies, fixing inconsistent data, and duplicates.

**IMPORTATION OF THE DATA**

**Database Selection:**

This command sets the context for subsequent SQL queries by selecting the products database as the active database. It ensures that all subsequent operations are performed within this specific database, avoiding ambiguity when dealing with similarly named tables across different databases.

USE products;

**Displaying Tables:**

This command lists all the tables available in the currently selected database (products).

It is useful for getting an overview of the data structure within the database, especially when exploring the contents of a new or unfamiliar database.

SHOW TABLES;

This workflow is a common approach in data exploration and preparation, ensuring that users are working with the correct data and focusing on the most relevant parts of it.

SELECT \* FROM smartphone\_cleaned\_v21 LIMIT 10;

select brand\_name, model, price, rating, fast\_charging

from smartphone\_cleaned\_v21;

**DATA ANALYSIS**

The purpose of the query is to present a ranked list of cellphones from well-known manufacturers, taking into account both price and rating. The query effectively filters out brands that are not of interest by using the ‘IN’ clause, concentrating on particular well-known brands.

SELECT brand\_name, model, price, rating

FROM smartphone\_cleaned\_v21

WHERE brand\_name IN ('oneplus', 'samsung', 'motorola', 'realme', 'apple')

ORDER BY price DESC, rating DESC;

#Checking the total number of smartphones (or rows) in the dataset

It's a basic way to check the size of a dataset or the completeness of a specific column.

select count(brand\_name) as count\_name from smartphone\_cleaned\_v21;

# Checking the average value of the price column in the table.

This metric provides a central tendency measure, giving an insight into the typical price of smartphones in the dataset.

SELECT AVG(price) AS avg\_price FROM smartphone\_cleaned\_v21;

#This query computes the total sum of all values in the price column of the table. The total price is used in estimating the total market value of the products in the dataset or for inventory valuation.

SELECT SUM(price) AS total\_price FROM smartphone\_cleaned\_v21;

# how each brand performs in terms of average rating and price.

SELECT

brand\_name,

AVG(rating) AS avg\_rating,

AVG(price) AS avg\_price

FROM

smartphone\_cleaned\_v21

GROUP BY

brand\_name;

SELECT: brand of the smartphone

The AVG function calculates the average of the rating column for each group (brand).

For the AVG func(price) calculates the average of the price column for each brand group

FROM: the table from which to retrieve the data

GROUP BY: computes the averages (AVG) within each group.

#Checking the top fast fast brands with the fastest charging rate. The query is aimed at producing a summary report that shows the number of smartphone models available for each brand (numofmodels), sorted by the maximum fast charging capability of each brand's models.

SELECT brand\_name, COUNT(model) AS numofmodels

FROM smartphone\_cleaned\_v21

GROUP BY brand\_name

ORDER BY MAX(fast\_charging) DESC

LIMIT 0, 1000;

SELECT: specifies the columns to be returned in the result set.

FROM: to retrieve the data.

GROUP BY: one row for each brand, showing the total number of models for that brand.

ORDER BY: determines the order in which the rows are returned.

# the query provides a comprehensive overview of the smartphone market, segmented by brand, with a focus on high-quality products. The query is designed to provide a summary of smartphone brands, focusing on their average rating and price, as well as the total ratings and prices.

SELECT

brand\_name,

AVG(rating) AS avg\_rating,

AVG(price) AS avg\_price,

SUM(rating) AS total\_rating,

SUM(price) AS total\_price

FROM

smartphone\_cleaned\_v21

GROUP BY

brand\_name

HAVING

AVG(rating) > 4

LIMIT 0, 1000;

SELECT : used to group and differentiate the data in the results.

AVG(rating) AS avg\_rating: Calculates the average rating for each brand.

AVG(price) AS avg\_price: Computes the average price of smartphones for each brand.

SUM(rating) AS total\_rating: Sums up the ratings for all products under each brand.

SUM(price) AS total\_price: Sums up the prices of all products under each brand.

FROM : Indicates the table from which the data is being retrieved.

GROUP BY: all calculations (averages and sums) are done separately for each brand.

HAVING: Filters the grouped results to include only those brands with an average rating greater than 4.

# For checking a detailed and categorized view of smartphone brands,

SELECT

brand\_name,

AVG(rating) AS avg\_rating,

AVG(price) AS avg\_price,

SUM(rating) AS total\_rating,

SUM(price) AS total\_price,

CASE

WHEN AVG(rating) >= 4.5 THEN 'High'

WHEN AVG(rating) >= 3.0 THEN 'Medium'

ELSE 'Low'

END AS rating\_category

FROM

smartphone\_cleaned\_v21

GROUP BY

brand\_name

HAVING

AVG(rating) > 2

LIMIT 0, 1000;

SELECT: used to group and differentiate the data in the results.

AVG(rating) AS avg\_rating: Calculates the average rating for each brand.

AVG(price) AS avg\_price: Computes the average price of smartphones for each brand.

SUM(rating) AS total\_rating: Sums up the ratings for all products under each brand.

SUM(price) AS total\_price: Sums up the prices of all products under each brand.

CASE: provides a qualitative assessment of the brand's performance based on customer ratings.

FROM: Indicates the table from which the data is being retrieved.

GROUP BY: all calculations (averages and sums) are done separately for each brand.

HAVING: Filters the results to include only those brands with an average rating greater than 2.