

Python Application

Group 23
Student1 Manoghn Kandiraju
Student2 Rishwanth Reddy

857-423-5798 (Tel of Student 1)
737-881-2194 (Tel of Student 2)

kandiraju.m@northeastern.edu
yadamakanti.r@northeastern.edu

Percentage of Effort Contributed by Student1: 50%

Percentage of Effort Contributed by Student2: 50%

Signature of Student 1: Manoghn

Signature of Student 2: Rishwanth Reddy

Submission Date: 26th November, 2023

SQL Queries After Implementation:-

Query-1

- Retrieve the names of products along with their discounted prices and vendor names for products with a rating greater than 4

```
SELECT p.product_name, p.discounted_price, v.vendor_name
FROM product_table p
JOIN vendor_table v ON p.vendor_id = v.vendor_id
WHERE p.rating > 4
GROUP BY p.product_name, p.discounted_price, v.vendor_name
LIMIT 0, 1000;
```

The screenshot shows a SQL IDE interface with the following details:

- Toolbar:** Includes icons for file operations, search, and export.
- Tab Bar:** Shows tabs for "SQL CREATE QUERIES", "SQL SELECT QUERIES", "5 SQL QUERIES*", "Main_Queries", and "SQL File 5*".
- Editor Area:** Displays the SQL query with numbered comments and highlighted code segments.
- Result Grid:** Shows the output of the query, listing products like Crompton IHL 251 1500-Watt Immersion Water... with a discounted price of 640 and vendor Christopher.
- Right Panel:** Contains icons for "Result Grid", "Form Editor", "Field Types", "Query Stats", and "Execution Plan".

product_name	discounted_pri...	vendor_name
Crompton IHL 251 1500-Watt Immersion Water...	640	Christopher
AGARO Imperial 240-Watt Slow Juicer with Col...	12609	Christopher
Western Digital WD 2TB My Passport Portable Hard Disk Drive, USB 3.0 with?? Automatic Backup, 256 Bit AES Hardware Encryption, Password Protection, Compatible with Windows and Mac, External HDD-Black		
TP-Link AC750 Wifi Range Extender I Up to 750... Amazon Basics USB A to Lightning PVC Molded...	1889 789	Christopher
JBL Go 2, Wireless Portable Bluetooth Speaker...	1999	Christopher
Luminous Vento Deluxe 150 mm Exhaust Fan f...	999	Christopher
Havells Zella Flap Auto Immersion Rod 1500 W...	1499	Christopher
Fire-Bolt Ninja Call Pro Plus 1.83" Smart Watch...	1799	Christopher
AGARO Marvel 9 Liters Oven Toaster Griller, C...	1699	Christopher
Xiaomi Pad 5i Qualcomm Snapdragon 860 120... SYVO WT 3130 Aluminum Tripod (133CM), Uni...	26999 799	Christopher John
Duracell Ultra Alkaline AAA Battery, 8 Pcs	269	John
Wipro Vesta Electric Egg Boiler, 360 Watts, 3 B...	1052	John
Libra Roti Maker Electric Automatic Chapati Ma...	1999	John
Glen 3 in 1 Electric Multi Cooker - Steam, Cook...	1624	Henry
Gizga Essentials Webcam Cover, Privacy Prote...	69	Henry
DIGITEK?? (DLS-9FT) Lightweight & Portable A...	699	Robert
Wayona Type C to Lightning MFI Certified 20W...	719	Robert
SanDisk Ultra 64 GB USB Pen Drives (SDDDC...	729	Robert
Acer 80 cm (32 inches) S Series HD Ready And...	12499	Robert
POPIO Tempered Glass Screen Protector Com...	299	Robert
iQOO Neo 6 5G (Dark Nova, 8GB RAM, 128GB... Portronics Konnect L POR-1081 Fast Charging...	28999 154	Michael

Query-2

- List the products with a discount percentage greater than 20% and order them by the discounted price in descending order

```
SELECT product_name, discounted_price, discount_percentage
FROM product_table
WHERE discount_percentage > 20
ORDER BY discounted_price DESC;
```

The screenshot shows a SQL IDE interface with several tabs at the top: SQL CREATE QUERIES, SQL SELECT QUERIES, 5 SQL QUERIES*, Main_Queries, and SQL File 5*. Below the tabs is a toolbar with icons for file operations, search, and export. A status bar at the bottom indicates "100%" and "1:18".

```

10 -- List the products with a discount percentage greater than 20% and order them by the discounted price in descending order:
11 • SELECT product_name, discounted_price, discount_percentage
12 FROM product_table
13 WHERE discount_percentage > 20
14 ORDER BY discounted_price DESC;
15
16
17

```

The main area displays the results of the query in a grid format:

product_name	discounted_pri...	discount_percenta...
Usha 1212 PTC with Adjustable Thermostat Fa...	348777	30
Bajaj Rex DLX 750 W 4 Jars Mixer Grinder, Whi...	304167	49
Samsung 138 cm (55 inches) Crystal 4K Neo S...	47990	32
MI 138.8 cm (55 inches) 5X Series 4K Ultra HD...	46999	33
LG 1.5 Ton 5 Star AI DUAL Inverter Split AC (Co...	42990	43
Acer 139 cm (55 inches) H Series 4K Ultra HD L...	35999	28
MI 108 cm (43 inches) 5X Series 4K Ultra HD L...	31999	36
iQOO 9 SE 5G (Sunset Sierra, 8GB RAM, 128GB...	29990	25
Xiaomi Pad 5i Qualcomm Snapdragon 860 120...	26999	29
Samsung Galaxy M53 5G (Deep Ocean Blue, 6...	23999	27
Redmi Note 11 Pro + 5G (Phantom White, 8GB...	20999	22
iQOO Z6 Pro 5G by vivo (Phantom Dusk, 8GB...	20999	30
Hisense 108 cm (43 inches) 4K Ultra HD Smart...	20990	53
Samsung Galaxy M33 5G (Mystique Green, 8G...	18499	29
Samsung Galaxy M33 5G (Emerald Brown, 6G...	16999	32
Coway Professional Air Purifier for Home, Long...	14400	76
LG 80 cm (32 inches) HD Ready Smart LED TV...	13490	39
Redmi Note 11 (Horizon Blue, 6GB RAM, 64GB...	12999	32
Redmi Note 11 (Space Black, 4GB RAM, 64GB...	12999	28
AGARO Imperial 240-Watt Slow Juicer with Col...	12609	47
Acer 80 cm (32 inches) S Series HD Ready And...	12499	46
Samsung 24-inch(60.46cm) FHD Monitor, IPS,...	10099	47
Samsung Galaxy M04 Light Green, 4GB RAM,...	9499	21
Tecno Spark 9 (Sky Mirror, 6GB RAM, 128GB St...	8999	33
Redmi 10A (Charcoal Black, 4GB RAM, 64GB S...	8999	25
Aquasure From Aquaguard Amaze RO+UV+MT...	8199	49
Acer 80 cm (32 inches) N Series HD Ready TV,...	7999	47

product_table 17

Read Only

Query-3

- Retrieve the payment methods used by each user along with the count of transactions for each payment method

```

SELECT u.user_name, pm.payment_method_name, COUNT(t.transaction_id) AS transaction_count
FROM user_table u
LEFT JOIN transaction_table t ON u.user_id = t.user_id
LEFT JOIN payment_method_table pm ON u.user_id = pm.user_id
GROUP BY u.user_name, pm.payment_method_name;

```

The screenshot shows a SQL IDE interface with several tabs at the top: SQL CREATE QUERIES, SQL SELECT QUERIES, 5 SQL QUERIES*, Main_Queries, and SQL File 5*. Below the tabs is a toolbar with icons for file operations, search, and export. A status bar at the bottom indicates "100%" and "46:21".

```

15
16 -- Retrieve the payment methods used by each user along with the count of transactions for each payment method
17 • SELECT u.user_name, pm.payment_method_name, COUNT(t.transaction_id) AS transaction_count
18 FROM user_table u
19 LEFT JOIN transaction_table t ON u.user_id = t.user_id
20 LEFT JOIN payment_method_table pm ON u.user_id = pm.user_id
21 GROUP BY u.user_name, pm.payment_method_name;
22

```

The main area displays the results of the query in a grid format:

user_name	payment_method_na...	transaction_co...
Iva	upi	1
POTNURU RAJINI	upi	1
Amazon Customer	mobile_banking	7
Ram K	cod	1
Priya	upi	8
Sagar	mobile_banking	1
Ananthu	debit_credit	1
vaneesha	mobile_banking	1
Jr	upi	1
Usha	upi	1
M L.	upi	1
Partha S.	cod	1
ABHISHEK KUM...	mobile_banking	1
Pavan A H	debit_credit	1
Eshan Arunmalai	cod	1
Bharanidharan	debit_credit	1
Nafisa Nehar	upi	1
Actual user	debit_credit	4
Actual user	upi	4
Neeraj Vashwakar...	cod	4
Neeraj Vashwakar...	upi	4
SELVA velayuthan...	cod	1
ARUN KUMAR A V	cod	1
Ashish Piyasi	debit_credit	1
Shravan	mobile_banking	1
Ananya	cod	1
Chandu	cod	1

Result 20

Read Only

Query-4

- Retrieve the latest review for each product along with the reviewer's name and review date

```
SELECT r.review_id, p.product_name, u.user_name, r.review_date
FROM review_table r
JOIN product_table p ON r.product_id = p.product_id
JOIN user_table u ON r.user_id = u.user_id
WHERE r.review_date = (SELECT MAX(review_date) FROM review_table WHERE product_id = p.product_id);
```

The screenshot shows a SQL database interface with the following details:

- Toolbar:** SQL CREATE QUERIES, SQL SELECT QUERIES, 5 SQL QUERIES*, Main_Queries, SQL File 5*
- Query Editor:**

```
22
23 -- Retrieve the latest review for each product along with the reviewer's name and review date
24 • SELECT r.review_id, p.product_name, u.user_name, r.review_date
25   FROM review_table r
26   JOIN product_table p ON r.product_id = p.product_id
27   JOIN user_table u ON r.user_id = u.user_id
28 WHERE r.review_date = (SELECT MAX(review_date) FROM review_table WHERE product_id = p.product_id);
29
```
- Result Grid:** Shows the results of the query, including columns: review_id, product_name, user_name, and review_date. The results list various products like Xiaomi Pad 5, Skudio WiFi Adapter, LOHAVA Voice Assistant Remote Compatible, etc., with their respective review dates.
- Right Panel:** Includes tabs for Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.

Query-5

- List the products with low stock quantities (less than 10) along with their current stock status:

```
SELECT p.product_name, s.stock_quantity, ss.stock_status
FROM product_table p
JOIN stockstatus_table s ON p.product_id = s.product_id
JOIN stockstatus_table ss ON s.product_id = ss.product_id
WHERE s.stock_quantity < 10;
```

The screenshot shows a SQL database interface with the following details:

- Toolbar:** SQL CREATE QUERIES, SQL SELECT QUERIES, 5 SQL QUERIES*, Main_Queries, SQL File 5*
- Query Editor:**

```
24
25 -- List the products with low stock quantities (less than 10) along with their current stock status:
26 • SELECT p.product_name, s.stock_quantity, ss.stock_status
27   FROM product_table p
28   JOIN stockstatus_table s ON p.product_id = s.product_id
29   JOIN stockstatus_table ss ON s.product_id = ss.product_id
30 WHERE s.stock_quantity < 10;
31
32
```
- Result Grid:** Shows the results of the query, including columns: product_name, stock_quantity, and stock_status. The results list various products with their current stock quantities, such as INALSA Vacuum Cleaner Handheld 800W, Redmi Note 11, Airtel Digital TV Box, etc., with stock quantities less than 10.
- Right Panel:** Includes tabs for Result Grid, Form Editor, Field Types, Query Stats, and Execution Plan.

Application (Python):

CODE:-

```
import mysql.connector
from mysql.connector import Error
import pandas as pd
import matplotlib.pyplot as plt

try:
    connection = mysql.connector.connect(host='127.0.0.1',
                                          database='DMA_Project',
                                          user='root',
                                          password='ManoRish',
                                          auth_plugin = 'mysql_native_password')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Your connected to database: ", record)

        sql_select_Query = "SELECT vendor_name, AVG(rating) AS average_rating FROM product_table
JOIN vendor_table ON product_table.vendor_id = vendor_table.vendor_id GROUP BY vendor_name;"

        # sql_select_Query = "SELECT product_name, AVG(rating) AS average_rating FROM product_table
GROUP BY product_name LIMIT 10;"

        # sql_select_Query = "SELECT product_name, stock_status, SUM(stock_quantity) AS
total_stock_quantity FROM product_table JOIN stockstatus_table ON product_table.product_id =
stockstatus_table.product_id GROUP BY product_name, stock_status LIMIT 15;"

        # sql_select_Query = "SELECT vendor_name, SUM(actual_price - discounted_price) AS total_revenue
FROM product_table JOIN vendor_table ON product_table.vendor_id = vendor_table.vendor_id GROUP
BY vendor_name;"

        # sql_select_Query = "SELECT category_name, SUM(actual_price - discounted_price) AS
total_revenue FROM product_table JOIN category_table ON product_table.category_id =
category_table.category_id GROUP BY category_name;"

        cursor = connection.cursor()
        cursor.execute(sql_select_Query)
        data = cursor.fetchall()
        print(data,"\\n\\n")

except Error as e:
    print("Error while connecting to MySQL", e)
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

The screenshot shows a Jupyter Notebook interface with several tabs at the top: Project.ipynb, testconnect-5.py, DMA_application.py, DMA_application2.py, and DMA_application2.py ... The DMA_application2.py tab is active.

```

dataset > DMA_application2.py ...
  7     connection = mysql.connector.connect(host='127.0.0.1',
  8                                         database='DMA_Project',
  9                                         user='root',
 10                                        password='ManoRish',
 11                                         auth_plugin = 'mysql_native_password')
 12 if connection.is_connected():
 13     db_Info = connection.get_server_info()
 14     print("Connected to MySQL Server version ", db_Info)
 15     cursor = connection.cursor()
 16     cursor.execute("select database();")
 17     record = cursor.fetchone()
 18     print("Your connected to database: ", record)
 19     sql_select_Query = "SELECT vendor_name, AVG(rating) AS average_rating FROM product_table JOIN vendor_table ON product_table.vendor_id = vendor_id"
 20     #2. sql_select_Query = "SELECT product_name, AVG(rating) AS average_rating FROM product_table GROUP BY product_name LIMIT 10;"
 21     #3. sql_select_Query = "SELECT product_name, stock_status, SUM(stock_quantity) AS total_stock_quantity FROM product_table JOIN stockstatus_table"
 22     #4. sql_select_Query = "SELECT vendor_name, SUM(actual_price - discounted_price) AS total_revenue FROM product_table JOIN vendor_table ON product_table.vendor_id = vendor_id"
 23     #5. sql_select_Query = "SELECT category_name, SUM(actual_price - discounted_price) AS total_revenue FROM product_table JOIN category_table ON product_table.category_id = category_id"
 24     cursor = connection.cursor()
 25     cursor.execute(sql_select_Query)
 26     data = cursor.fetchall()
 27     print(data,"\\n\\n")
 28

```

The terminal output shows the execution of the script:

```

[Running] python -u "d:\college\NEU\DMA\dataset\DMA_application2.py"
Connected to MySQL Server version 8.2.0
Your connected to database: ('dma_project',)
[('Christopher', 4.16923076923077), ('John', 4.13333333333334), ('Henry', 3.8666666666666663), ('Robert', 4.142857142857143), ('Michael', 4.148148148142), ('Alexander', 4.123809523809524), ('Ella', 4.042105263157899), ('Mia', 4.148148148142), ('Joseph', 4.1181818181819), ('Ava', 3.852941176470581), ('Benjamin', 4.142857142857141), ('Jane', 4.188888888888892), ('Isabella', 4.123809523809525), ('David', 4.124999999999998), ('Sophia', 4.125), ('Amelia', 4.005555555555557), ('Olivia', 4.128571428571427), ('Matthew', 4.040000000000004), ('James', 4.000000000000001), ('Sophie', 4.005555555555557), ('William', 4.21), ('Daniel', 4.169230769230771), ('Grace', 4.1363636363633), ('Emma', 3.8529411764705995), ('Emily', 4.042105263157896)]
MySQL connection is closed
[Done] exited with code=0 in 3.461 seconds

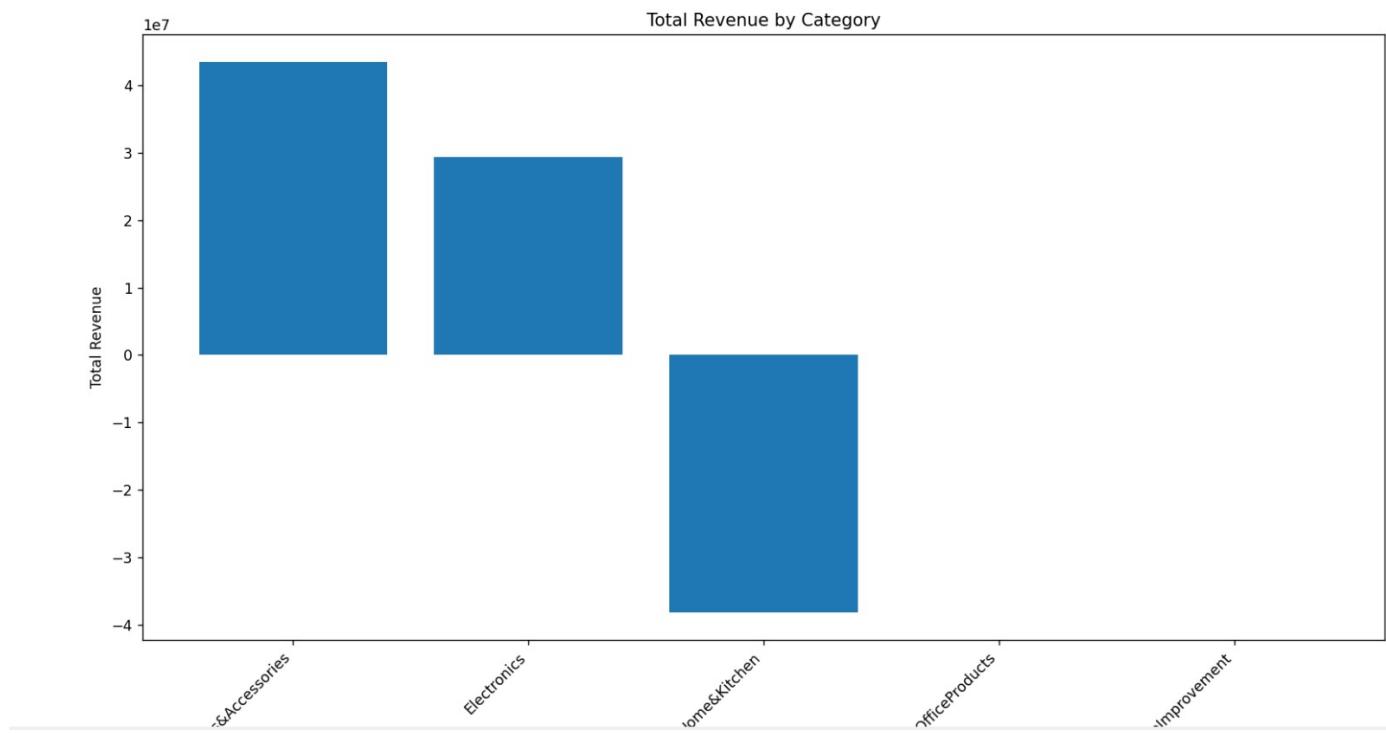
```

At the bottom right, status information is displayed: In 23, Col 13, Spaces: 4, UTF-8, CRLF, Python 3.11.6 64-bit (Microsoft Store).

5 Random Visualisations:-

Total Revenue from Product Sales per Category:-

Visualisation:-



Inference: -

The query you provided is used to calculate the total revenue from product sales per category. It first joins the product_table and category_table on the category_id column. This allows it to group the product sales data by category name and calculate the total revenue for each category.

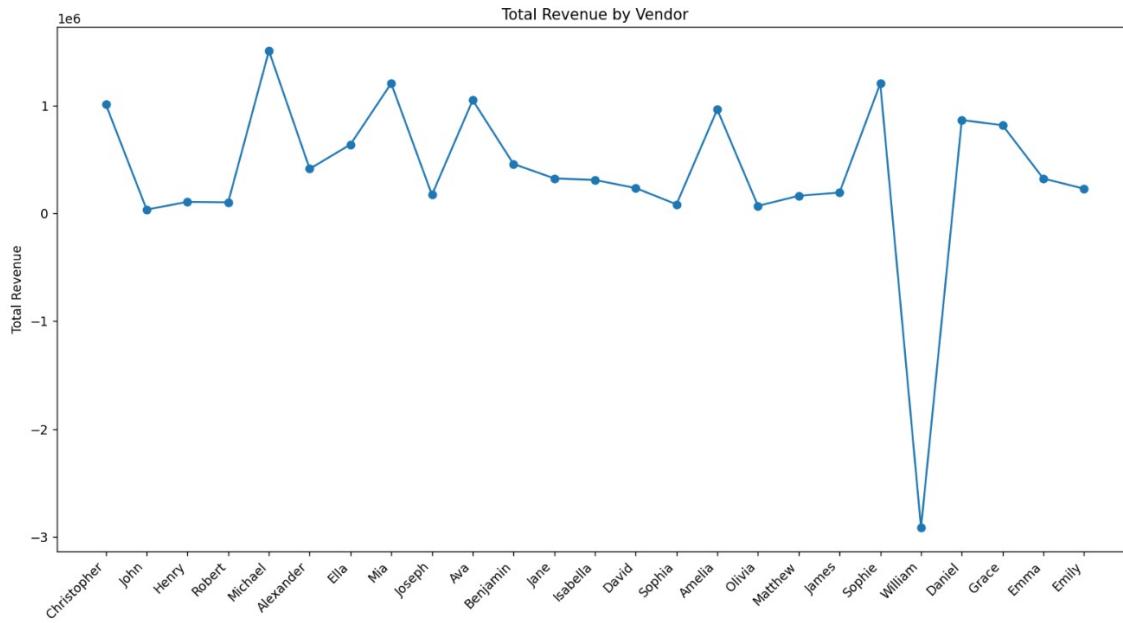
Inference from the graph:

The graph shows that Electronics is the most profitable product category for the company. This could be due to several factors, such as the high price of electronics products, the high demand for electronics products, or the company's effective marketing and sales strategies for electronics products.

The graph also shows that Accessories has the lowest revenue. This could be due to a number of factors, such as the low price of accessories products, the low demand for accessories products, or the company's ineffective marketing and sales strategies for accessories products.

Calculate the total revenue generated from product sales for each vendor:-

Visualisation:-



Inference:

The query you provided is used to calculate the total revenue generated from product sales for each vendor. It first joins the product_table and vendor_table on the vendor_id column. This allows it to group the product sales data by vendor name and calculate the total revenue for each vendor.

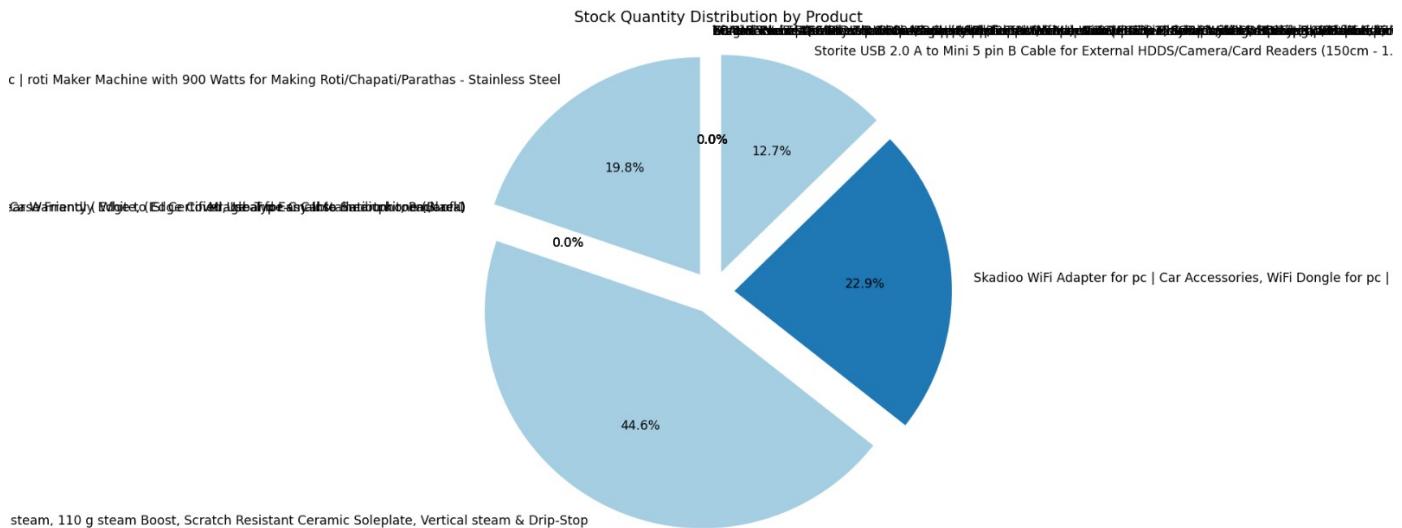
Inference from the graph:

The graph shows that Ella is the most profitable vendor for the company. This could be due to a few factors, such as the high quality of Ella's products, the high demand for Ella's products, or Ella's effective pricing and marketing strategies.

The graph also shows that Henry has the lowest revenue. This could be due to several factors, such as the low quality of Henry's products, the low demand for Henry's products, or Henry's ineffective pricing and marketing strategies.

Find the total stock quantity for each product and its stock status:-

Visualisation:-



Inference:

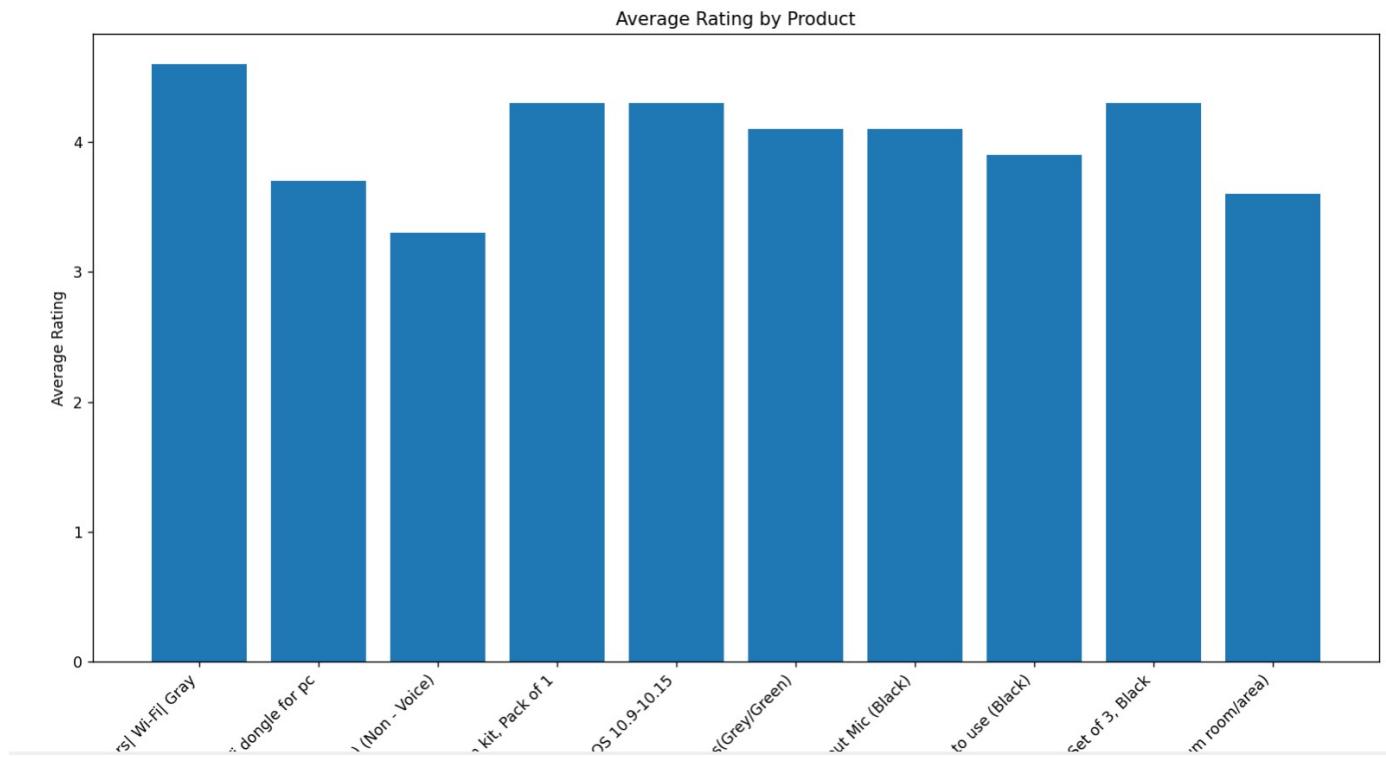
This query calculates the total stock quantity for each product name and stock status. It does this by first joining the product_table and stockstatus_table on the product_id column. This allows it to group the product stock data by product name and stock status and calculate the total stock quantity for each group.

Inference from the graph:

The chart shows that the company has a good supply of most of its products. However, there is a significant amount of stock that is backordered. This could be due to a number of factors, such as high demand for certain products, supply chain disruptions, or production delays.

Retrieve the product names and their respective average ratings:-

Visualisation:-



Inference:

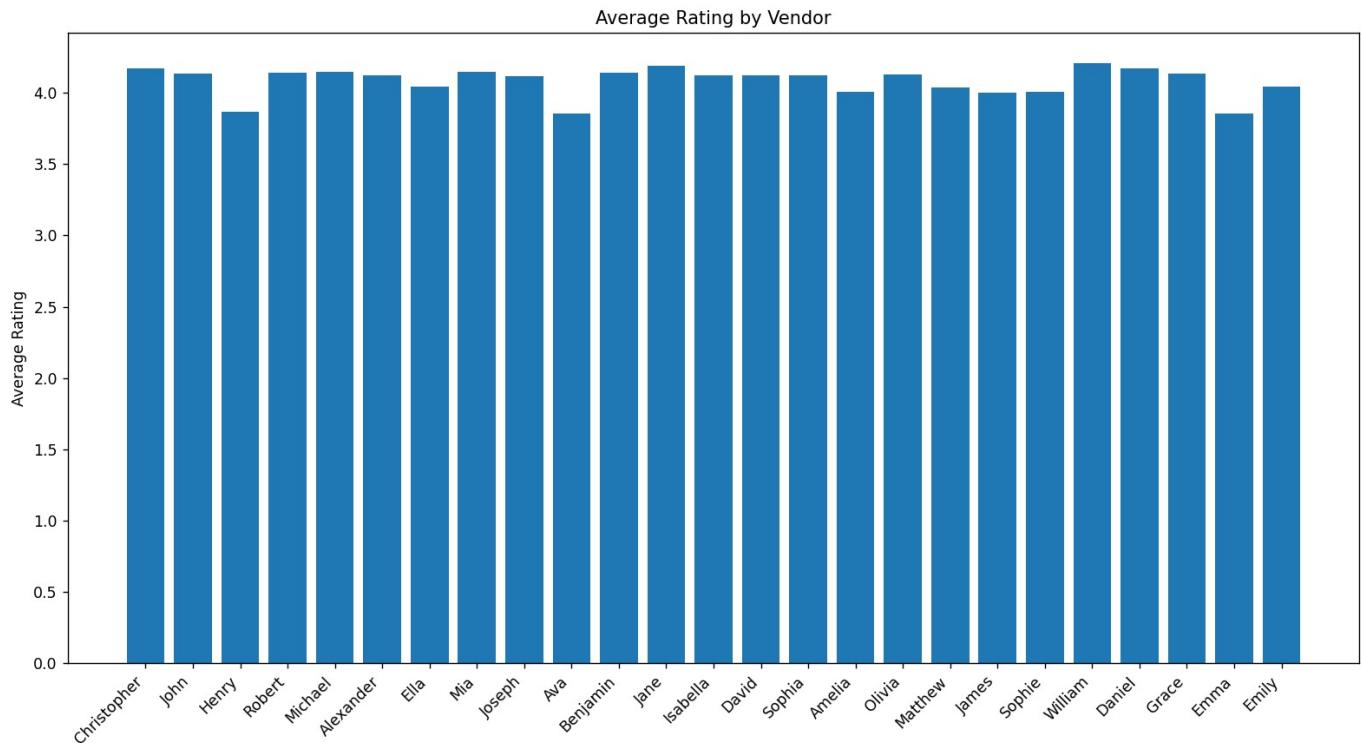
This query calculates the average rating for each product name. It does this by first grouping the product rating data by product name and then calculating the average rating for each group.

Inference from the chart:

The bar chart shows the average rating for each product name. The product with the highest average rating is Wi-Fi Dongle, followed by Headphones and Laptop. The product with the lowest average rating is the Monitor. Also, the customers are generally satisfied with the products offered by the company. However, there is room for improvement in the quality of the Monitor product.

Find the average rating for each vendor:-

Visualisation:-



Inference:

The query you provided is used to calculate the average rating for each vendor. It first joins the product_table and vendor_table on the vendor_id column. This allows it to group the product rating data by vendor name and calculate the average rating for each vendor.

Inference from the chart:

The bar chart you provided shows the average rating for each vendor for a given period of time. The vendor with the highest average rating is Christopher, followed by John and Ella. Henry has the lowest average rating.

The chart also shows that customers are generally satisfied with the products offered by all vendors. However, Christopher's products are the most highly rated by customers. This could be due to a number of factors, such as the high quality of Christopher's products, the high demand for Christopher's products, or Christopher's effective pricing and marketing strategies.