**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](mailto:kandiraju.m@northeastern.edu)

[yadamakanti.r@northeastern.edu](mailto: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;

A screenshot of a computer

Description automatically generated

***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;

A screenshot of a computer

Description automatically generated

***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;

A screenshot of a computer

Description automatically generated

***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);

A screenshot of a computer

Description automatically generated

***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;

A screenshot of a computer

Description automatically generated

**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")

**A screen shot of a computer

Description automatically generated**

**5 Random Visualisations:-**

**Total Revenue from Product Sales per Category:-**

Visualisation:-

**A graph showing a bar graph

Description automatically generated with medium confidence**

**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:-

**A graph with blue lines and dots

Description automatically generated**

**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:-

**A pie chart with text and numbers

Description automatically generated**

**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:-

**A graph of blue bars

Description automatically generated**

**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:-

**A graph of blue bars with white text

Description automatically generated**

**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 line 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.