**-GROUP 7**

-Group Members :

1.Tatiana Tan

2.Arun Sharma

-Date: 07/21/2024

-Assignment: Module – 11 Milestone #3

-GitHub link: <https://github.com/SharmaArun017/CSD310.git>

= Arun Sharma

Based on a typical case study for a winery, here are three potential reports that could be useful for business decisions:

= Report 1: Wine Inventory and Price Report

Description: This report shows the current inventory of wines along with their prices. This helps in tracking wine availability and pricing, which is crucial for pricing strategies and stock management.

Sql Query :

# Wine Inventory and Price Report Query

wine\_inventory\_query = """ SELECT Wine.WineID, Wine.WineName, Wine.Type, Wine.Price, IFNULL(SUM(OrderDetails.Quantity), 0) AS TotalSold

FROM Wine

LEFT JOIN OrderDetails ON Wine.WineID = OrderDetails.WineID

GROUP BY Wine.WineID, Wine.WineName, Wine.Type, Wine.Price

ORDER BY Wine.WineID; """

Pyhton script :

import mysql.connector

# Connect to the MySQL database

connection = mysql.connector.connect(

host='localhost',

user='root',

password='password',

database='BacchusWinery'

)

cursor = connection.cursor()

# Wine Inventory and Price Report Query

wine\_inventory\_query = """

SELECT Wine.WineID, Wine.WineName, Wine.Type, Wine.Price, IFNULL(SUM(OrderDetails.Quantity), 0) AS TotalSold

FROM Wine

LEFT JOIN OrderDetails ON Wine.WineID = OrderDetails.WineID

GROUP BY Wine.WineID, Wine.WineName, Wine.Type, Wine.Price

ORDER BY Wine.WineID;

"""

cursor.execute(wine\_inventory\_query)

rows = cursor.fetchall()

headers = [i[0] for i in cursor.description]

# Display the result

print("\nWine Inventory and Price Report\n")

print(" | ".join(headers))

print("-" \* (len(headers) \* 15))

for row in rows:

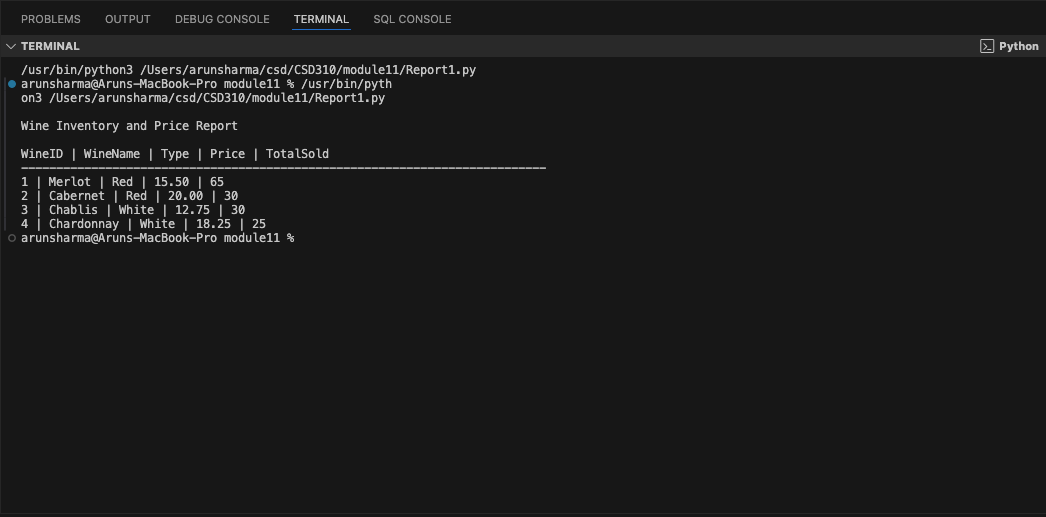
print(" | ".join(str(cell) for cell in row))

# Close the connection

cursor.close()

connection.close()

Screenshot Result :



= Report 2: Supplier Supply Summary Report

Description: This report provides a summary of supplies from each supplier, including supply types and quantities. It helps in understanding supplier contributions and managing supplier relationships.

SQL Query:

# Supplier Supply Summary Report Query

supplier\_supply\_query = """

SELECT Supplier.SupplierID, Supplier.SupplierName, Supply.SupplyType, SUM(Supply.Quantity) AS TotalQuantity

FROM Supplier

JOIN Supply ON Supplier.SupplierID = Supply.SupplierID

GROUP BY Supplier.SupplierID, Supplier.SupplierName, Supply.SupplyType

ORDER BY Supplier.SupplierID; """

Python Script:

import mysql.connector

# Connect to the MySQL database

connection = mysql.connector.connect(

host='localhost',

user='root',

password='password',

database='BacchusWinery'

)

cursor = connection.cursor()

# Supplier Supply Summary Report Query

supplier\_supply\_query = """

SELECT Supplier.SupplierID, Supplier.SupplierName, Supply.SupplyType, SUM(Supply.Quantity) AS TotalQuantity

FROM Supplier

JOIN Supply ON Supplier.SupplierID = Supply.SupplierID

GROUP BY Supplier.SupplierID, Supplier.SupplierName, Supply.SupplyType

ORDER BY Supplier.SupplierID;

"""

cursor.execute(supplier\_supply\_query)

rows = cursor.fetchall()

headers = [i[0] for i in cursor.description]

# Display the result

print("\nSupplier Supply Summary Report\n")

print(" | ".join(headers))

print("-" \* (len(headers) \* 15))

for row in rows:

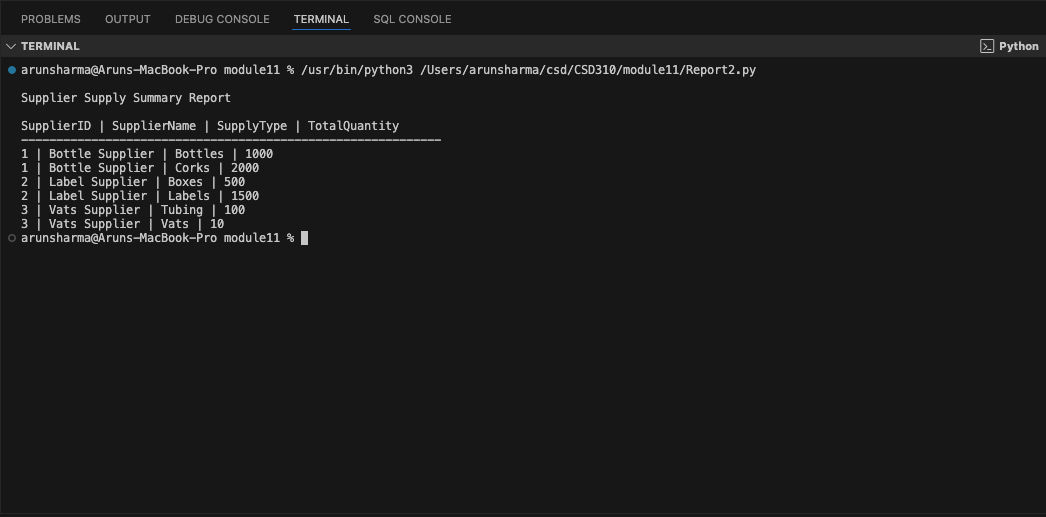
print(" | ".join(str(cell) for cell in row))

# Close the connection

cursor.close()

connection.close()

Screenshot Result :



Report 3: Order Details Report

Description: This report details all orders, including the distributor, order date, wine, and quantity. It helps in tracking orders and analyzing sales trends.

SQL Query:

# Order Details Report Query

order\_details\_query = """

SELECT Orders.OrderID, Orders.OrderDate, Distributor.DistributorName, Wine.WineName, OrderDetails.Quantity

FROM Orders

JOIN Distributor ON Orders.DistributorID = Distributor.DistributorID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Wine ON OrderDetails.WineID = Wine.WineID

ORDER BY Orders.OrderID, OrderDetails.OrderDetailsID; """

Pyhton Script:

import mysql.connector

# Connect to the MySQL database

connection = mysql.connector.connect(

host='localhost',

user='root',

password='password',

database='BacchusWinery'

)

cursor = connection.cursor()

# Order Details Report Query

order\_details\_query = """

SELECT Orders.OrderID, Orders.OrderDate, Distributor.DistributorName, Wine.WineName, OrderDetails.Quantity

FROM Orders

JOIN Distributor ON Orders.DistributorID = Distributor.DistributorID

JOIN OrderDetails ON Orders.OrderID = OrderDetails.OrderID

JOIN Wine ON OrderDetails.WineID = Wine.WineID

ORDER BY Orders.OrderID, OrderDetails.OrderDetailsID;

"""

cursor.execute(order\_details\_query)

rows = cursor.fetchall()

headers = [i[0] for i in cursor.description]

# Display the result

print("\nOrder Details Report\n")

print(" | ".join(headers))

print("-" \* (len(headers) \* 15))

for row in rows:

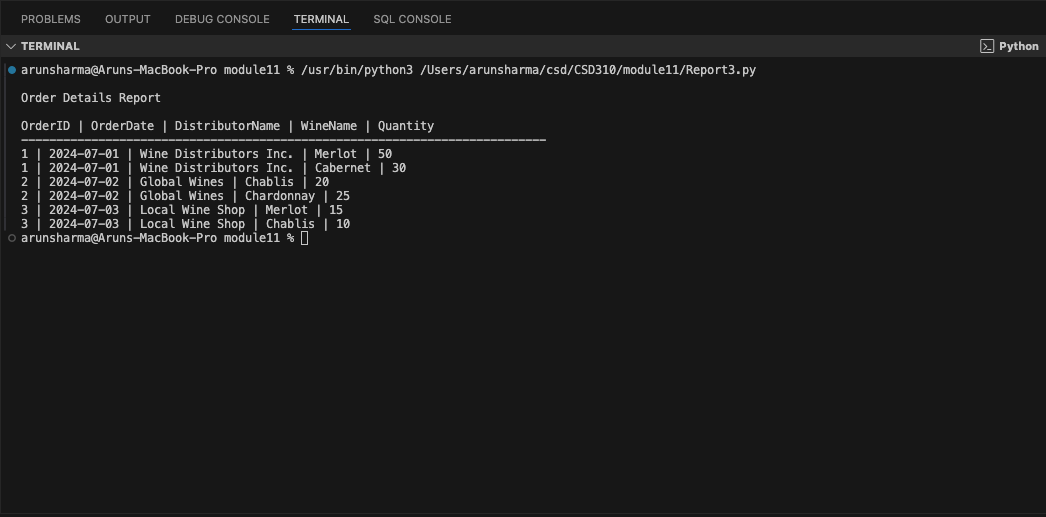
print(" | ".join(str(cell) for cell in row))

# Close the connection

cursor.close()

connection.close()

Screenshot Result :



= Tatiana Tan

### **Report 1: Wine Inventory and Price Report**

**Description:** This report shows the current inventory of wines along with their prices. This helps in tracking wine availability and pricing, which is crucial for pricing strategies and stock management.

**SQL Query:**

SELECT WineName, Type, Price  
FROM Wine;

**Python Script:**

python

Copy code

import mysql.connector  
from tabulate import tabulate  
  
# Connect to the MySQL database  
connection = mysql.connector.connect(  
 host='localhost',  
 user='root',  
 password='password',  
 database='BacchusWinery'  
)  
  
cursor = connection.cursor()  
  
# Function to display Wine Inventory Report  
def display\_wine\_inventory():  
 cursor.execute("SELECT WineName, Type, Price FROM Wine")  
 rows = cursor.fetchall()  
 headers = [i[0] for i in cursor.description]  
 print("\nWine Inventory and Price Report")  
 print(tabulate(rows, headers, tablefmt='psql'))  
  
# Display Wine Inventory Report

display\_wine\_inventory()  
  
# Close the connection  
cursor.close()  
connection.close()

### **Report 2: Supplier Supply Summary Report**

**Description:** This report provides a summary of supplies from each supplier, including supply types and quantities. It helps in understanding supplier contributions and managing supplier relationships.

**SQL Query:**

SELECT s.SupplierName, sp.SupplyType, sp.Quantity  
FROM Supplier s  
JOIN Supply sp ON s.SupplierID = sp.SupplierID;

**Python Script:**

python

Copy code

import mysql.connector  
from tabulate import tabulate  
  
# Connect to the MySQL database  
connection = mysql.connector.connect(  
 host='localhost',  
 user='root',  
 password='password',  
 database='BacchusWinery'  
)  
  
cursor = connection.cursor()  
  
# Function to display Supplier Supply Summary Report  
def display\_supplier\_supply\_summary():  
 cursor.execute("""

SELECT s.SupplierName, sp.SupplyType, sp.Quantity  
 FROM Supplier s  
 JOIN Supply sp ON s.SupplierID = sp.SupplierID  
 """)  
 rows = cursor.fetchall()  
 headers = [i[0] for i in cursor.description]  
 print("\nSupplier Supply Summary Report")  
 print(tabulate(rows, headers, tablefmt='psql'))  
  
# Display Supplier Supply Summary Report  
display\_supplier\_supply\_summary()  
  
# Close the connection  
cursor.close()  
connection.close()

### **Report 3: Order Details Report**

**Description:** This report details all orders, including the distributor, order date, wine, and quantity. It helps in tracking orders and analyzing sales trends.

**SQL Query:**

SELECT o.OrderID, o.OrderDate, d.DistributorName, w.WineName, od.Quantity  
FROM Orders o  
JOIN Distributor d ON o.DistributorID = d.DistributorID  
JOIN OrderDetails od ON o.OrderID = od.OrderID  
JOIN Wine w ON od.WineID = w.WineID;

**Python Script:**

python

Copy code

import mysql.connector  
from tabulate import tabulate  
  
# Connect to the MySQL database  
connection = mysql.connector.connect(

host='localhost',  
 user='root',  
 password='password',  
 database='BacchusWinery'  
)  
  
cursor = connection.cursor()  
  
# Function to display Order Details Report  
def display\_order\_details():  
 cursor.execute("""  
 SELECT o.OrderID, o.OrderDate, d.DistributorName, w.WineName, od.Quantity  
 FROM Orders o  
 JOIN Distributor d ON o.DistributorID = d.DistributorID  
 JOIN OrderDetails od ON o.OrderID = od.OrderID  
 JOIN Wine w ON od.WineID = w.WineID  
 """)  
 rows = cursor.fetchall()  
 headers = [i[0] for i in cursor.description]  
 print("\nOrder Details Report")  
 print(tabulate(rows, headers, tablefmt='psql'))  
  
# Display Order Details Report  
display\_order\_details()  
  
# Close the connection  
cursor.close()  
connection.close()

Results :

1.

A black screen with blue lines

Description automatically generated  
2.

A screenshot of a computer

Description automatically generated  
3.

A screenshot of a computer

Description automatically generated