VIETNAM NATIONAL UNIVERSITY HCM CITY
UNIVERSITY OF ECONOMICS AND LAW
FACULTY OF INFORMATION SYSTEMS

MID-TERM EXAM
MACHINE LEARNING IN BUSINESS ANALYTICS
HCM City, ...

Code: **ML01**

MID-TERM EXAM

Course: MACHINE LEARNING IN BUSINESS ANALYTICS

Time: ...... minutes

(All documents and devices are allowed)

General Requirement:

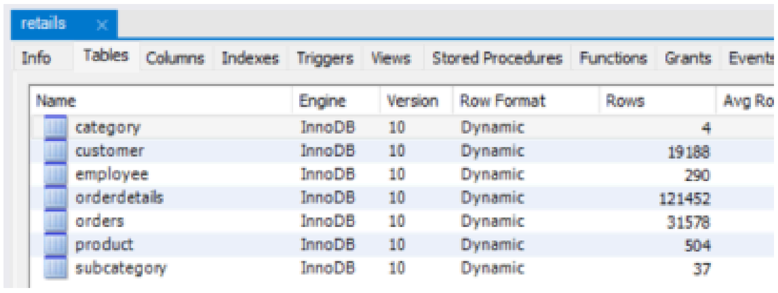
✓ Students create a folder to save their work in drive **D:\** with the name Fullname_StudentID_Class (For example: Full name is **Tran Thanh Cong**, Student ID is **113**, class **DL114**, then create a folder named **TranThanhCong_113_DL114**).

✓ Students are required to save the exam every 10 minutes to avoid losing the exam.

✓ All other requirements Students must follow the instructions of the exam proctor.

PROBLEMS

BaBa company wants to build a sales management system. And company wants the system to have the following functions:



Name	Engine	Version	Row Format	Rows	Avg Ro
category	InnoDB	10	Dynamic	4	
customer	InnoDB	10	Dynamic	19188	
employee	InnoDB	10	Dynamic	290	
orderdetails	InnoDB	10	Dynamic	121452	
orders	InnoDB	10	Dynamic	31578	
product	InnoDB	10	Dynamic	504	
subcategory	InnoDB	10	Dynamic	37	

Link Database (MySQL Script):

https://tranduythanh.com/datasets/Retails_Mysql.rar

Students have to do the tasks:

1) Import MySQL Script to Database in local machine with database name "**Retails**", all data and structures are required
2) Write Python coding statistic:
    a. Statistics on total sales of items purchased by customers
    b. Statistics of total revenue by each category
    c. Statistics of total revenue by Year
    d. Statistics of total revenue by Category and Year
3) Write Python coding for query:
    a. Write a function to return customer detail information if CustomerID is requested.

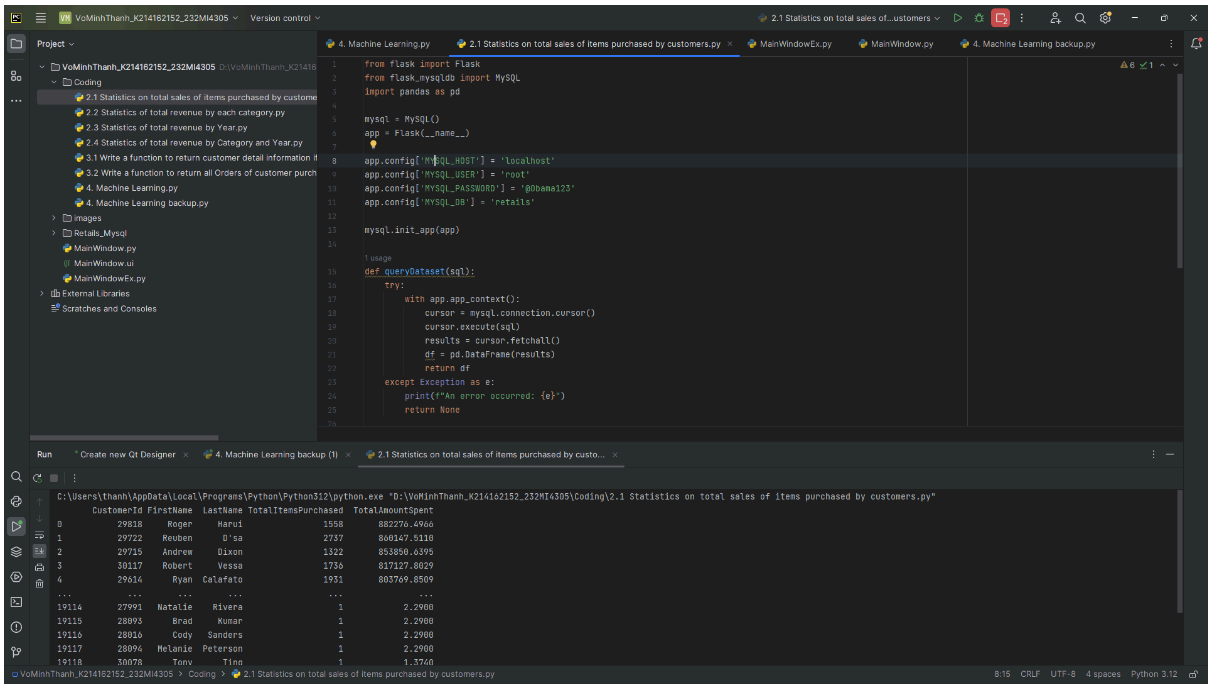
   b. Write a function to return all Orders of customer purchased if CustomerID is requested
4) According to Orders history, Using **Linear Regression** to train a **Machine learning model** to predict trend of customer sales behavior. Students think about idea by yourself to do this task (not limit the way to define the customer sales behavior or price prediction, so each Student will have different solutions).
5) Design a nice PyQt GUI for user interaction all of the above tasks.

Questions	Code CLO
1	CLO3
2	CLO 4, CLO 5
3	CLO 4, CLO 5
4	CLO 1, CLO 2, CLO 3
5	CLO 4, CLO 5

Head of Department **Lecturer**

Tran Duy Thanh, PhD. **Tran Duy Thanh, PhD.**

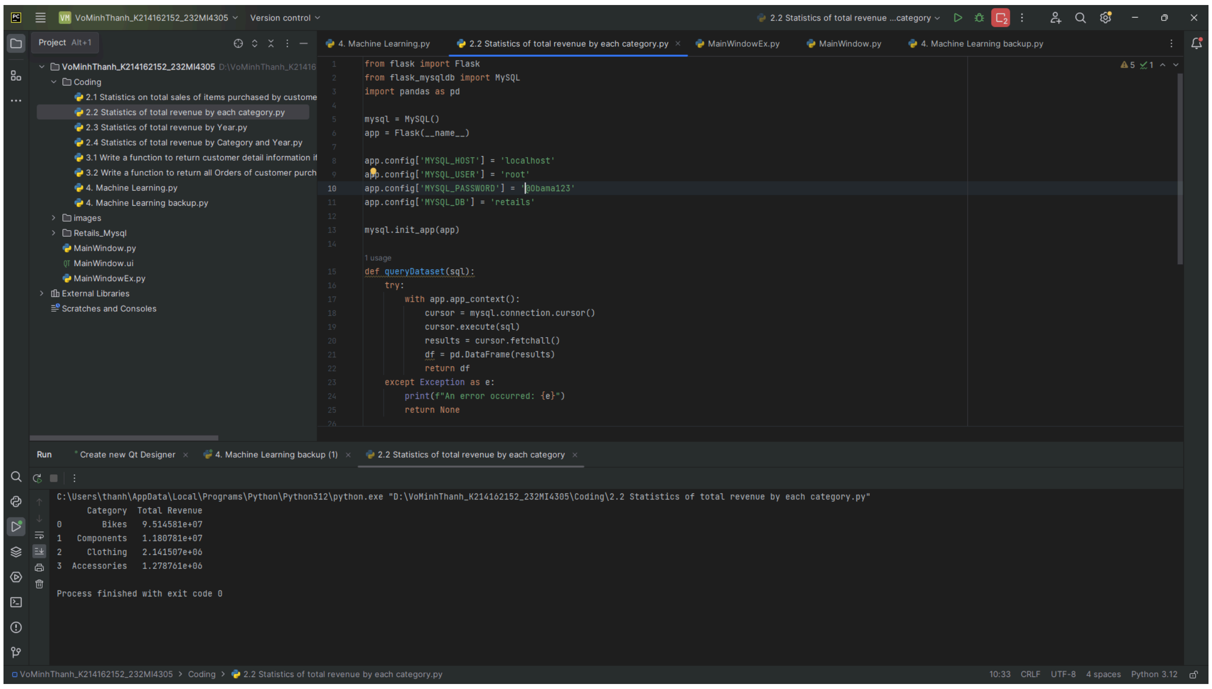
2.1



```python
from flask import Flask
from flask_mysqldb import MySQL
import pandas as pd

mysql = MySQL()
app = Flask(__name__)

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '@Obama123'
app.config['MYSQL_DB'] = 'retails'

mysql.init_app(app)

def queryDataset(sql):
    try:
        with app.app_context():
            cursor = mysql.connection.cursor()
            cursor.execute(sql)
            results = cursor.fetchall()
            df = pd.DataFrame(results)
            return df
    except Exception as e:
        print(f"An error occurred: {e}")
        return None
```

```
C:\Users\thanh\AppData\Local\Programs\Python\Python312\python.exe "D:\VoMinhThanh_K214162152_232MI4305\Coding\2.1 Statistics on total sales of items purchased by customers.py"
      CustomerId FirstName  LastName TotalItemsPurchased  TotalAmountSpent
0          29818     Roger     Harui                1558       882276.4966
1          29722    Reuben      D'sa                2737       860147.5110
2          29715    Andrew     Dixon                1322       853850.6395
3          30117    Robert     Vessa                1736       817127.8029
4          29614      Ryan  Calafato                1931       803769.8509
...          ...       ...       ...                 ...               ...
19114      27991   Natalie    Rivera                   1            2.2900
19115      28093      Brad     Kumar                   1            2.2900
19116      28016      Cody   Sanders                   1            2.2900
19117      28094   Melanie  Peterson                   1            2.2900
19118      30078      Tony      Ting                   1            1.3740
```

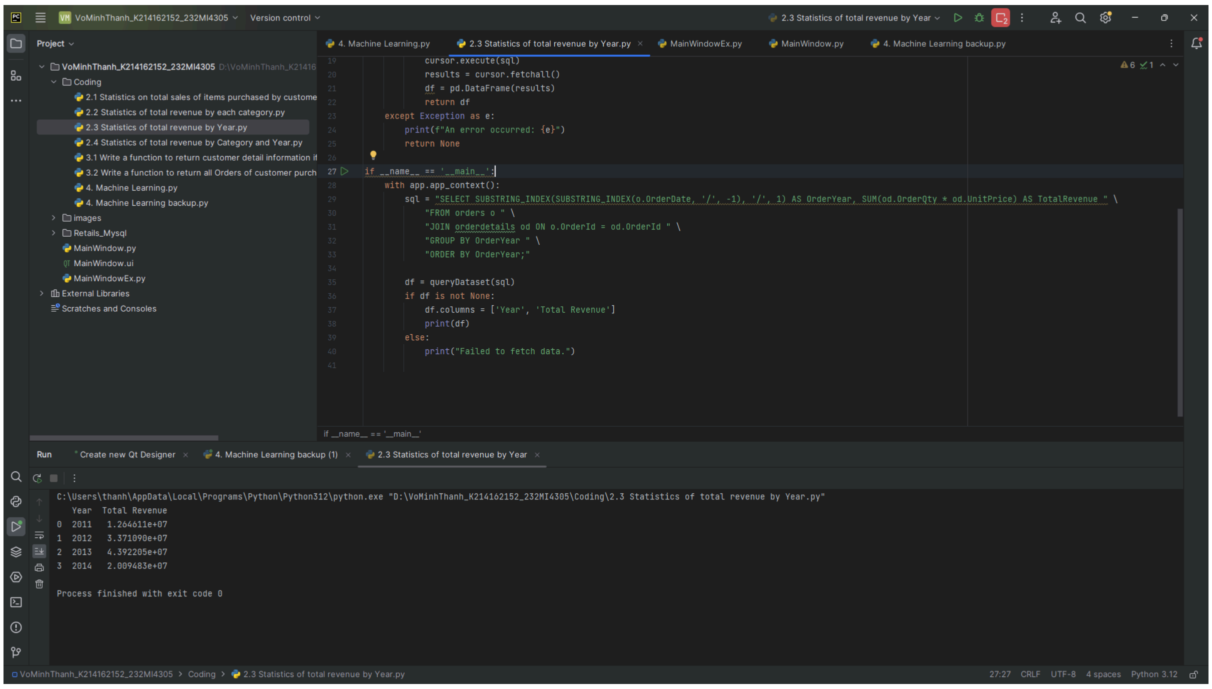
2.2



```
from flask import Flask
from flask_mysqldb import MySQL
import pandas as pd

mysql = MySQL()
app = Flask(__name__)

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '@Obama123'
app.config['MYSQL_DB'] = 'retails'

mysql.init_app(app)

def queryDataset(sql):
    try:
        with app.app_context():
            cursor = mysql.connection.cursor()
            cursor.execute(sql)
            results = cursor.fetchall()
            df = pd.DataFrame(results)
            return df
    except Exception as e:
        print(f"An error occurred: {e}")
        return None
```

```
C:\Users\thanh\AppData\Local\Programs\Python\Python312\python.exe "D:\VoMinhThanh_K214162152_232MI4305\Coding\2.2 Statistics of total revenue by each category.py"
      Category  Total Revenue
0        Bikes   9.514581e+07
1   Components   1.180781e+07
2     Clothing   2.141507e+06
3  Accessories   1.278761e+06

Process finished with exit code 0
```

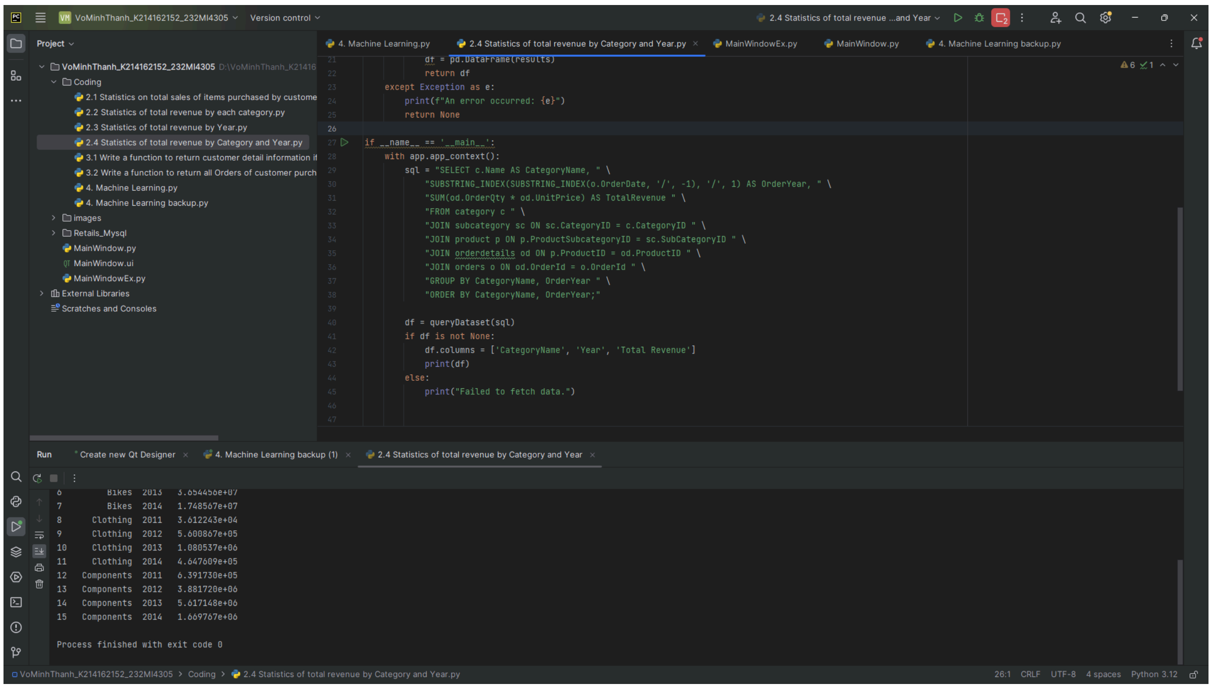
2.3



```
            cursor.execute(sql)
            results = cursor.fetchall()
            df = pd.DataFrame(results)
            return df
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

if __name__ == '__main__':
    with app.app_context():
        sql = "SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(o.OrderDate, '/', -1), '/', 1) AS OrderYear, SUM(od.OrderQty * od.UnitPrice) AS TotalRevenue " \
            "FROM orders o " \
            "JOIN orderdetails od ON o.OrderId = od.OrderId " \
            "GROUP BY OrderYear " \
            "ORDER BY OrderYear;"

        df = queryDataset(sql)
        if df is not None:
            df.columns = ['Year', 'Total Revenue']
            print(df)
        else:
            print("Failed to fetch data.")
```

```
C:\Users\thanh\AppData\Local\Programs\Python\Python312\python.exe "D:\VoMinhThanh_K214162152_232MI4305\Coding\2.3 Statistics of total revenue by Year.py"
   Year  Total Revenue
0  2011   1.264611e+07
1  2012   3.371090e+07
2  2013   4.392205e+07
3  2014   2.009483e+07

Process finished with exit code 0
```

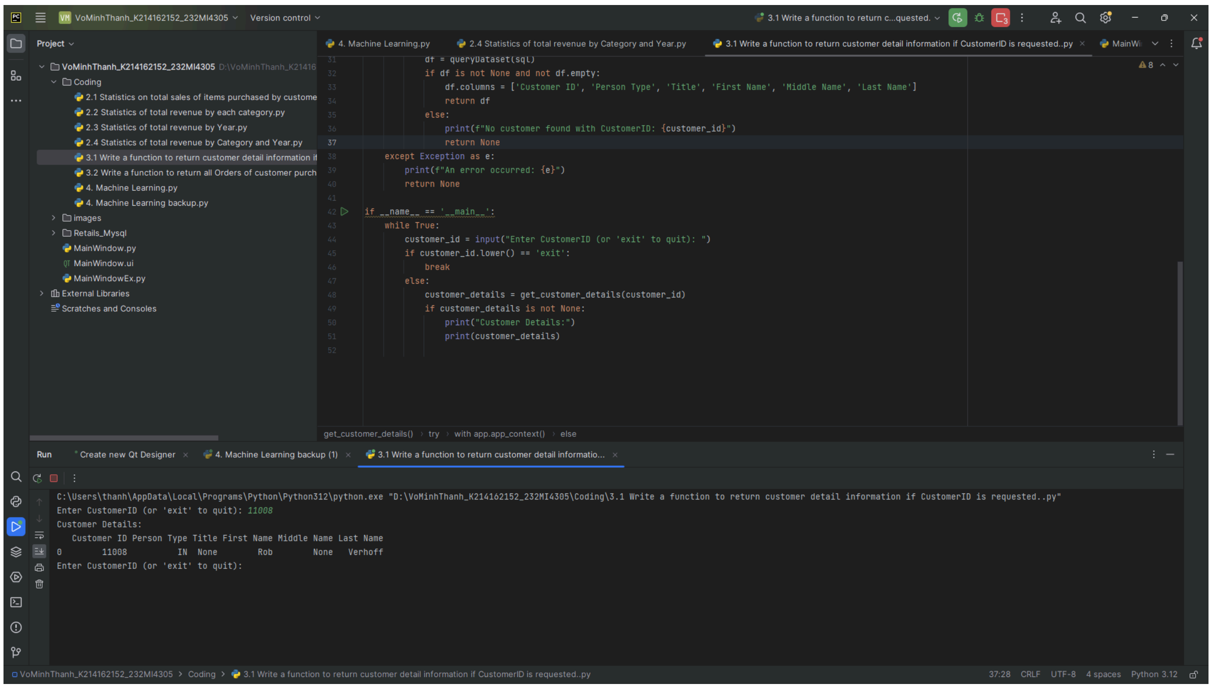
2.4


VoMinhThanh_K214162152_232MI4305
Version control
2.4 Statistics of total revenue ...and Year
Project
4. Machine Learning.py
2.4 Statistics of total revenue by Category and Year.py
MainWindowEx.py
MainWindow.py
4. Machine Learning backup.py
Coding
2.1 Statistics on total sales of items purchased by custome
2.2 Statistics of total revenue by each category.py
2.3 Statistics of total revenue by Year.py
2.4 Statistics of total revenue by Category and Year.py
3.1 Write a function to return customer detail information if
3.2 Write a function to return all Orders of customer purch
4. Machine Learning.py
4. Machine Learning backup.py
images
Retails_Mysql
MainWindow.py
MainWindow.ui
MainWindowEx.py
External Libraries
Scratches and Consoles
Run
Create new Qt Designer
4. Machine Learning backup (1)
2.4 Statistics of total revenue by Category and Year
VoMinhThanh_K214162152_232MI4305 > Coding > 2.4 Statistics of total revenue by Category and Year.py
26:1 CRLF UTF-8 4 spaces Python 3.12

```python
                df = pd.DataFrame(results)
                return df
        except Exception as e:
            print(f"An error occurred: {e}")
            return None

if __name__ == '__main__':
    with app.app_context():
        sql = "SELECT c.Name AS CategoryName, " \
            "SUBSTRING_INDEX(SUBSTRING_INDEX(o.OrderDate, '/', -1), '/', 1) AS OrderYear, " \
            "SUM(od.OrderQty * od.UnitPrice) AS TotalRevenue " \
            "FROM category c " \
            "JOIN subcategory sc ON sc.CategoryID = c.CategoryID " \
            "JOIN product p ON p.ProductSubcategoryID = sc.SubCategoryID " \
            "JOIN orderdetails od ON p.ProductID = od.ProductID " \
            "JOIN orders o ON od.OrderId = o.OrderId " \
            "GROUP BY CategoryName, OrderYear " \
            "ORDER BY CategoryName, OrderYear;"

        df = queryDataset(sql)
        if df is not None:
            df.columns = ['CategoryName', 'Year', 'Total Revenue']
            print(df)
        else:
            print("Failed to fetch data.")
```

```
6        Bikes  2013   3.654456e+07
7        Bikes  2014   1.748567e+07
8     Clothing  2011   3.612243e+04
9     Clothing  2012   5.600867e+05
10    Clothing  2013   1.080537e+06
11    Clothing  2014   4.647609e+05
12  Components  2011   6.391730e+05
13  Components  2012   3.881720e+06
14  Components  2013   5.617148e+06
15  Components  2014   1.669767e+06

Process finished with exit code 0
```

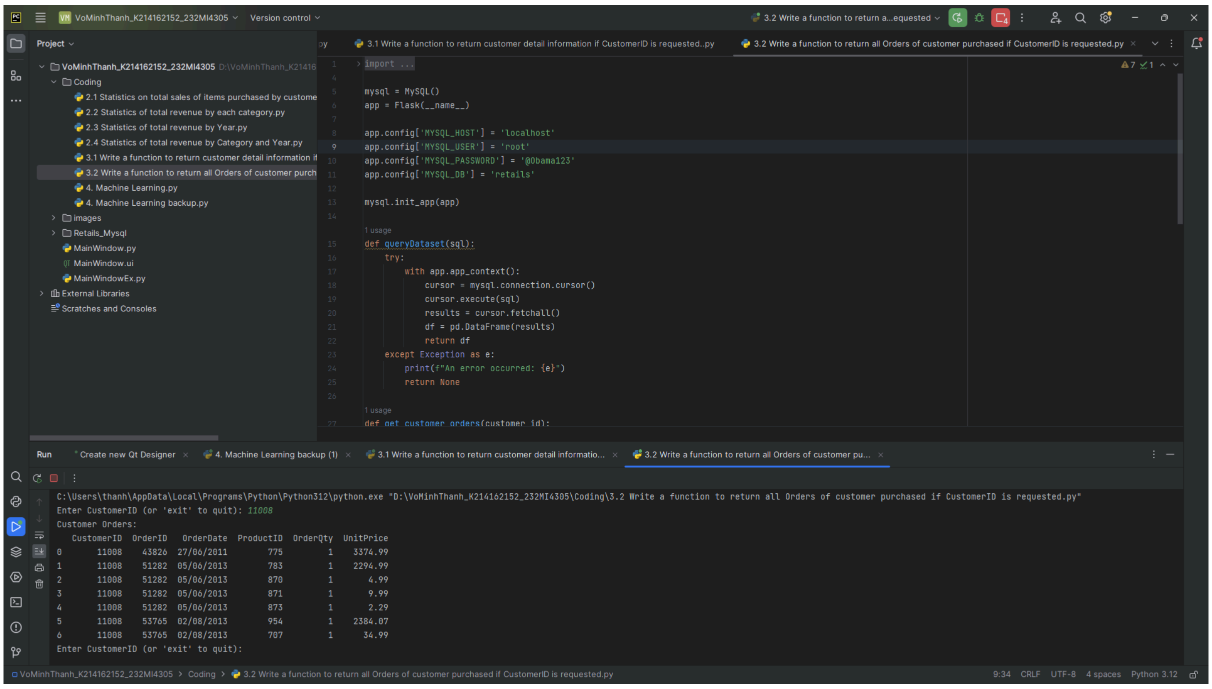
3.1


VoMinhThanh_K214162152_232MI4305
Version control
3.1 Write a function to return c...quested.
Project
4. Machine Learning.py
2.4 Statistics of total revenue by Category and Year.py
3.1 Write a function to return customer detail information if CustomerID is requested..py
Coding
2.1 Statistics on total sales of items purchased by custome
2.2 Statistics of total revenue by each category.py
2.3 Statistics of total revenue by Year.py
2.4 Statistics of total revenue by Category and Year.py
3.1 Write a function to return customer detail information if
3.2 Write a function to return all Orders of customer purch
4. Machine Learning.py
4. Machine Learning backup.py
images
Retails_Mysql
MainWindow.py
MainWindow.ui
MainWindowEx.py
External Libraries
Scratches and Consoles
get_customer_details() > try > with app.app_context() > else
Run
Create new Qt Designer
4. Machine Learning backup (1)
3.1 Write a function to return customer detail informatio...
VoMinhThanh_K214162152_232MI4305 > Coding > 3.1 Write a function to return customer detail information if CustomerID is requested..py
37:28 CRLF UTF-8 4 spaces Python 3.12

```python
            df = queryDataset(sql)
            if df is not None and not df.empty:
                df.columns = ['Customer ID', 'Person Type', 'Title', 'First Name', 'Middle Name', 'Last Name']
                return df
            else:
                print(f"No customer found with CustomerID: {customer_id}")
                return None
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

if __name__ == '__main__':
    while True:
        customer_id = input("Enter CustomerID (or 'exit' to quit): ")
        if customer_id.lower() == 'exit':
            break
        else:
            customer_details = get_customer_details(customer_id)
            if customer_details is not None:
                print("Customer Details:")
                print(customer_details)
```

```
C:\Users\thanh\AppData\Local\Programs\Python\Python312\python.exe "D:\VoMinhThanh_K214162152_232MI4305\Coding\3.1 Write a function to return customer detail information if CustomerID is requested..py"
Enter CustomerID (or 'exit' to quit): 11008
Customer Details:
   Customer ID Person Type Title First Name Middle Name Last Name
0        11008          IN  None        Rob        None   Verhoff
Enter CustomerID (or 'exit' to quit):
```

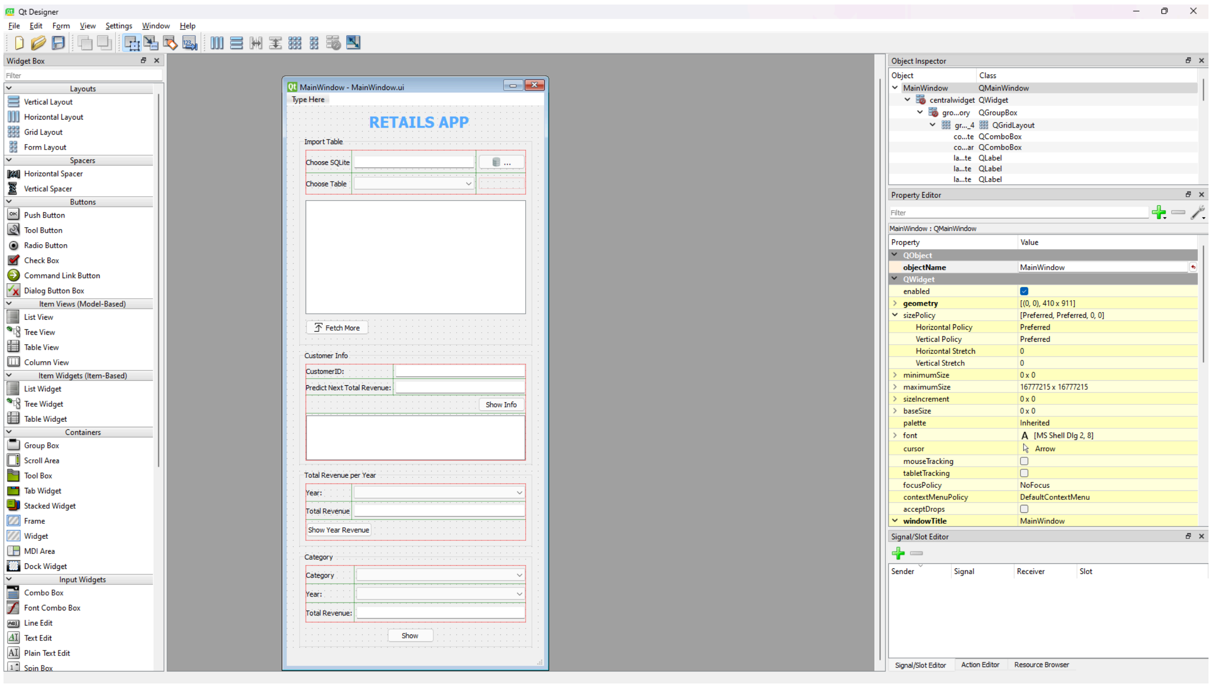
3.2


VoMinhThanh_K214162152_232MI4305
Version control
3.2 Write a function to return a...equested
Project
Coding
2.1 Statistics on total sales of items purchased by custome
2.2 Statistics of total revenue by each category.py
2.3 Statistics of total revenue by Year.py
2.4 Statistics of total revenue by Category and Year.py
3.1 Write a function to return customer detail information if
3.2 Write a function to return all Orders of customer purch
4. Machine Learning.py
4. Machine Learning backup.py
images
Retails_Mysql
MainWindow.py
MainWindow.ui
MainWindowEx.py
External Libraries
Scratches and Consoles
3.1 Write a function to return customer detail information if CustomerID is requested..py
3.2 Write a function to return all Orders of customer purchased if CustomerID is requested.py
Run
Create new Qt Designer
4. Machine Learning backup (1)
3.1 Write a function to return customer detail informatio...
3.2 Write a function to return all Orders of customer pu...
VoMinhThanh_K214162152_232MI4305 > Coding > 3.2 Write a function to return all Orders of customer purchased if CustomerID is requested.py
9:34 CRLF UTF-8 4 spaces Python 3.12

```python
import ...

mysql = MySQL()
app = Flask(__name__)

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '@Obama123'
app.config['MYSQL_DB'] = 'retails'

mysql.init_app(app)

1 usage
def queryDataset(sql):
    try:
        with app.app_context():
            cursor = mysql.connection.cursor()
            cursor.execute(sql)
            results = cursor.fetchall()
            df = pd.DataFrame(results)
            return df
    except Exception as e:
        print(f"An error occurred: {e}")
        return None

1 usage
def get_customer_orders(customer_id):
```

```
C:\Users\thanh\AppData\Local\Programs\Python\Python312\python.exe "D:\VoMinhThanh_K214162152_232MI4305\Coding\3.2 Write a function to return all Orders of customer purchased if CustomerID is requested.py"
Enter CustomerID (or 'exit' to quit): 11008
Customer Orders:
   CustomerID  OrderID  OrderDate  ProductID  OrderQty  UnitPrice
0       11008    43826  27/06/2011        775         1    3374.99
1       11008    51282  05/06/2013        783         1    2294.99
2       11008    51282  05/06/2013        870         1       4.99
3       11008    51282  05/06/2013        871         1       9.99
4       11008    51282  05/06/2013        873         1       2.29
5       11008    53765  02/08/2013        954         1    2384.07
6       11008    53765  02/08/2013        707         1      34.99
Enter CustomerID (or 'exit' to quit):
```

5.



Object Inspector

Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
gro...ory	QGroupBox
gr..._4	QGridLayout
co...te	QComboBox
co...ar	QComboBox
la...te	QLabel
la...te	QLabel
la...te	QLabel

Property Editor

MainWindow : QMainWindow

Property	Value
QObject	
objectName	MainWindow
QWidget	
enabled	✓
geometry	[(0, 0), 410 x 911]
sizePolicy	[Preferred, Preferred, 0, 0]
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Inherited
font	[MS Shell Dlg 2, 8]
cursor	Arrow
mouseTracking	☐
tabletTracking	☐
focusPolicy	NoFocus
contextMenuPolicy	DefaultContextMenu
acceptDrops	☐
windowTitle	MainWindow