Lab_Session_E_Commerce_May23

July 10, 2024

1 Lab Session

1.1 Table Of Contents

- Problem Statement
- Load required libraries
- Connect to DB using mysql-connector-python package
- Create database named `e_commerce`
- Create tables and insert data into tables as specified in the question
- Read all the questions and write sql queries to meet the objective

1.2 Problem Statement

1.2.1 An E-commerce website manages its data in the form of various tables.

You need to create a Database called `e_commerce` and various tables in it. The tables needed and attributes which need to be in every table are given before hand. All you have to do is create tables with data in it and answer some of the questions that follows.

1.2.2 Load Required Libraries

[5]:

```python
import mysql.connector
from mysql.connector import Error
import pandas as pd
import numpy as np
```

1.2.3 Connect to DB using Mysql-connector-python package

[7]:

```python
try:
    # Establish the connection
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password=''
    )

    if connection.is_connected():
        print("Connected to MySQL Server")
        # Create the 'e_commerce' database
        cursor = connection.cursor()
        cursor.execute("CREATE DATABASE e_commerce")
        print("Database 'e_commerce' created successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Connected to MySQL Server
Database 'e_commerce' created successfully
MySQL connection is closed
```

1.2.4 You are required to create a database named 'e_commerce'

[ ]:

```
#  already done above
```

1.2.5 Q1. Create tables for supplier, customer, category, product, productDetails, order, rating to store the data for the E-commerce with the schema definition given below.

- `supplier`(SUPP_ID int primary key, SUPP_NAME varchar(50), SUPP_CITY varchar(50), SUPP_PHONE varchar(10))
- `customer` (CUS_ID INT NOT NULL, CUS_NAME VARCHAR(20) NULL DEFAULT NULL, CUS_PHONE VARCHAR(10), CUS_CITY varchar(30) ,CUS_GENDER CHAR,PRIMARY KEY (CUS_ID))
- `category` (CAT_ID INT NOT NULL, CAT_NAME VARCHAR(20) NULL DEFAULT NULL,PRIMARY KEY (CAT_ID))
- `product` (PRO_ID INT NOT NULL, PRO_NAME VARCHAR(20) NULL DEFAULT NULL, PRO_DESC VARCHAR(60) NULL DEFAULT NULL, CAT_ID INT NOT NULL,PRIMARY KEY (PRO_ID),FOREIGN KEY (CAT_ID) REFERENCES CATEGORY (CAT_ID))
- `product_details` (PROD_ID INT NOT NULL, PRO_ID INT NOT NULL, SUPP_ID INT NOT NULL, PROD_PRICE INT NOT NULL, PRIMARY KEY (PROD_ID),FOREIGN KEY (PRO_ID) REFERENCES PRODUCT (PRO_ID), FOREIGN KEY (SUPP_ID) REFERENCES SUPPLIER(SUPP_ID))
- `order` (ORD_ID INT NOT NULL, ORD_AMOUNT INT NOT NULL, ORD_DATE DATE, CUS_ID INT NOT NULL, PROD_ID INT NOT NULL,PRIMARY KEY (ORD_ID),FOREIGN KEY (CUS_ID) REFERENCES CUSTOMER(CUS_ID),FOREIGN KEY (PROD_ID) REFERENCES PRODUCT_DETAILS(PROD_ID))

- **rating** (RAT_ID INT NOT NULL, CUS_ID INT NOT NULL, SUPP_ID INT NOT NULL, RAT_RATSTARS INT NOT NULL,PRIMARY KEY (RAT_ID),FOREIGN KEY (SUPP_ID) REFERENCES SUPPLIER (SUPP_ID),FOREIGN KEY (CUS_ID) REFERENCES CUSTOMER(CUS_ID))

[31]:

```python
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # Create supplier table
        cursor.execute("""
        CREATE TABLE supplier (
            SUPP_ID INT PRIMARY KEY,
            SUPP_NAME VARCHAR(50),
            SUPP_CITY VARCHAR(50),
            SUPP_PHONE VARCHAR(10)
        )
        """)

        # Create customer table
        cursor.execute("""
        CREATE TABLE customer (
            CUS_ID INT NOT NULL,
            CUS_NAME VARCHAR(20) NULL DEFAULT NULL,
            CUS_PHONE VARCHAR(10),
            CUS_CITY VARCHAR(30),
            CUS_GENDER CHAR,
            PRIMARY KEY (CUS_ID)
        )
        """)

        # Create category table
        cursor.execute("""
        CREATE TABLE category (
            CAT_ID INT NOT NULL,
            CAT_NAME VARCHAR(20) NULL DEFAULT NULL,
            PRIMARY KEY (CAT_ID)
        )
        """)
        # Create product table
        cursor.execute("""
        CREATE TABLE product (
            PRO_ID INT NOT NULL,
            PRO_NAME VARCHAR(20) NULL DEFAULT NULL,
            PRO_DESC VARCHAR(60) NULL DEFAULT NULL,
            CAT_ID INT NOT NULL,
            PRIMARY KEY (PRO_ID),
            FOREIGN KEY (CAT_ID) REFERENCES category(CAT_ID)
        )
        """)

        # Create product_details table
        cursor.execute("""
        CREATE TABLE product_details (
            PROD_ID INT NOT NULL,
            PRO_ID INT NOT NULL,
            SUPP_ID INT NOT NULL,
            PROD_PRICE INT NOT NULL,
            PRIMARY KEY (PROD_ID),
            FOREIGN KEY (PRO_ID) REFERENCES product(PRO_ID),
            FOREIGN KEY (SUPP_ID) REFERENCES supplier(SUPP_ID)
        )
        """)

        # Create order table
        cursor.execute("""
        CREATE TABLE `orders` (
            ORD_ID INT NOT NULL,
            ORD_AMOUNT INT NOT NULL,
            ORD_DATE DATE,
            CUS_ID INT NOT NULL,
            PROD_ID INT NOT NULL,
            PRIMARY KEY (ORD_ID),
            FOREIGN KEY (CUS_ID) REFERENCES customer(CUS_ID),
            FOREIGN KEY (PROD_ID) REFERENCES product_details(PROD_ID)
        )
        """)

        # Create rating table
        cursor.execute("""
        CREATE TABLE rating (
            RAT_ID INT NOT NULL,
            CUS_ID INT NOT NULL,
            SUPP_ID INT NOT NULL,
            RAT_RATSTARS INT NOT NULL,
            PRIMARY KEY (RAT_ID),
            FOREIGN KEY (SUPP_ID) REFERENCES supplier(SUPP_ID),
            FOREIGN KEY (CUS_ID) REFERENCES customer(CUS_ID)
        )
        """)

        print("All tables created successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Error: '1050 (42S01): Table 'supplier' already exists'
MySQL connection is closed
```

1.2.6 Q2. Insert the following data in the table created above

Note: If you are getting any error while inserting the data into tables, Kindly close the connection and reconnect

Table: supplier

SUPP_ID	SUPP_NAME	SUPP_CITY	SUPP_PHONE
1	Rajesh Retails	Delhi	1234567890
2	Appario Ltd.	Mumbai	258963147032
3	Knome products	Bangalore	9785462315
4	Bansal Retails	Kochi	8975463285
5	Mittal Ltd.	Lucknow	7898456532

[11]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into supplier table
        insert_supplier_query = """
        INSERT INTO supplier (SUPP_ID, SUPP_NAME, SUPP_CITY, SUPP_PHONE)
        VALUES (%s, %s, %s, %s)
        """

        # Data to be inserted
        suppliers = [
            (1, 'Rajesh Retails', 'Delhi', '1234567890'),
            (2, 'Appario Ltd.', 'Mumbai', '2589631470'),
            (3, 'Knome products', 'Bangalore', '9785462315'),
            (4, 'Bansal Retails', 'Kochi', '8975463285'),
            (5, 'Mittal Ltd.', 'Lucknow', '7898456532')
        ]

        # Insert data
        cursor.executemany(insert_supplier_query, suppliers)

        # Commit the transaction
        connection.commit()

        print("Supplier data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Supplier data inserted successfully
MySQL connection is closed
```

Table: customer

CUS_ID	CUS_NAME	SUPP_PHONE	CUS_CITY	CUS_GENDER
1	AAKASH	9999999999	DELHI	M
2	AMAN	9785463215	NOIDA	M
3	NEHA	9999999998	MUMBAI	F
4	MEGHA	9994562399	KOLKATA	F
5	PULKIT	7895999999	LUCKNOW	M

[12]:

```
# insert into "customer" table
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
```

```python
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into customer table
        insert_customer_query = """
        INSERT INTO customer (CUS_ID, CUS_NAME, CUS_PHONE, CUS_CITY, CUS_GENDER)
        VALUES (%s, %s, %s, %s, %s)
        """

        # Data to be inserted
        customers = [
            (1, 'AAKASH', '9999999999', 'DELHI', 'M'),
            (2, 'AMAN', '9785463215', 'NOIDA', 'M'),
            (3, 'NEHA', '9999999998', 'MUMBAI', 'F'),
            (4, 'MEGHA', '9994562399', 'KOLKATA', 'F'),
            (5, 'PULKIT', '7895999999', 'LUCKNOW', 'M')
        ]

        # Insert data
        cursor.executemany(insert_customer_query, customers)

        # Commit the transaction
        connection.commit()

        print("Customer data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Customer data inserted successfully
MySQL connection is closed
```

Table: category

CAT_ID	CAT_NAME
1	BOOKS
2	GAMES
3	GROCERIES

CAT_ID	CAT_NAME
4	ELECTRONICS
5	CLOTHES

[14]:

```python
# insert into "categoty" table
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into category table
        insert_category_query = """
        INSERT INTO category (CAT_ID, CAT_NAME)
        VALUES (%s, %s)
        """

        # Data to be inserted
        categories = [
            (1, 'BOOKS'),
            (2, 'GAMES'),
            (3, 'GROCERIES'),
            (4, 'ELECTRONICS'),
            (5, 'CLOTHES')
        ]

        # Insert data
        cursor.executemany(insert_category_query, categories)

        # Commit the transaction
        connection.commit()

        print("Category data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Category data inserted successfully
MySQL connection is closed
```

Table: product

PRO_ID	PRO_NAME	PRO_DESC	CAT_ID
1	GTA V	DFJDJFDJFDJFDJFJF	2
2	TSHIRT	DFDFJDFJDKFD	5
3	ROG LAPTOP	DFNTTNTNTERND	4
4	OATS	REURENTBTOTH	3
5	HARRY POTTER	NBEMCTHTJTH	1

[15]:

```
# insert into "product" table
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into product table
        insert_product_query = """
        INSERT INTO product (PRO_ID, PRO_NAME, PRO_DESC, CAT_ID)
        VALUES (%s, %s, %s, %s)
        """

        # Data to be inserted
        products = [
            (1, 'GTA V', 'DFJDJFDJFDJFDJFJF', 2),
            (2, 'TSHIRT', 'DFDFJDFJDKFD', 5),
            (3, 'ROG LAPTOP', 'DFNTTNTNTERND', 4),
            (4, 'OATS', 'REURENTBTOTH', 3),
            (5, 'HARRY POTTER', 'NBEMCTHTJTH', 1)
        ]

        # Insert data
        cursor.executemany(insert_product_query, products)

        # Commit the transaction
        connection.commit()
```

```python
        print("Product data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Product data inserted successfully
MySQL connection is closed
```

Table: product_details

PROD_ID	PRO_ID	SUPP_ID	PROD_PRICE
1	1	2	1500
2	3	5	30000
3	5	1	3000
4	2	3	2500
5	4	1	1000

[16]:

```python
# insert into "product_details" table
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into product_details table
        insert_product_details_query = """
        INSERT INTO product_details (PROD_ID, PRO_ID, SUPP_ID, PROD_PRICE)
        VALUES (%s, %s, %s, %s)
        """

        # Data to be inserted
        product_details = [
            (1, 1, 2, 1500),
            (2, 3, 5, 30000),
            (3, 5, 1, 3000),
            (4, 2, 3, 2500),
            (5, 4, 1, 1000)
        ]

        # Insert data
        cursor.executemany(insert_product_details_query, product_details)

        # Commit the transaction
        connection.commit()

        print("Product details data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Product details data inserted successfully
MySQL connection is closed
```

Table: orders

ORD_ID	ORD_AMOUNT	ORD_DATE	CUS_ID	PROD_ID
20	1500	2021-10-12	3	5
25	30500	2021-09-16	5	2
26	2000	2021-10-05	1	1
30	3500	2021-08-16	4	3
50	2000	2021-10-06	2	1

[33]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into orders table
        insert_orders_query = """
        INSERT INTO orders (ORD_ID, ORD_AMOUNT, ORD_DATE, CUS_ID, PROD_ID)
        VALUES (%s, %s, %s, %s, %s)
        """

        # Data to be inserted
        orders = [
            (20, 1500, '2021-10-12', 3, 5),
            (25, 30500, '2021-09-16', 5, 2),
            (26, 2000, '2021-10-05', 1, 1),
            (30, 3500, '2021-08-16', 4, 3),
            (50, 2000, '2021-10-06', 2, 1)
        ]

        # Insert data
        cursor.executemany(insert_orders_query, orders)

        # Commit the transaction
        connection.commit()

        print("Orders data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Orders data inserted successfully
MySQL connection is closed
```

Table: rating

RAT_ID	CUS_ID	SUPP_ID	RAT_RATSTARS
1	2	2	4
2	3	4	3
3	5	1	5
4	1	3	2
5	4	5	4

[21]:
```
# insert into "rating" table
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to insert data into rating table
        insert_rating_query = """
        INSERT INTO rating (RAT_ID, CUS_ID, SUPP_ID, RAT_RATSTARS)
        VALUES (%s, %s, %s, %s)
        """

        # Data to be inserted
        ratings = [
            (1, 2, 2, 4),
            (2, 3, 4, 3),
            (3, 5, 1, 5),
            (4, 1, 3, 2),
            (5, 4, 5, 4)
        ]

        # Insert data
        cursor.executemany(insert_rating_query, ratings)

        # Commit the transaction
        connection.commit()

        print("Ratings data inserted successfully")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Ratings data inserted successfully
MySQL connection is closed
```

1.2.7 Q3) Display the number of the customer group by their genders who have placed any order of amount greater than or equal to Rs.3000.

[41]:

```python
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get the number of customers by gender who placed orders >= 3000
        query = """
        SELECT c.CUS_GENDER, COUNT(*) AS NumberOfCustomers
        FROM orders o
        JOIN customer c ON o.CUS_ID = c.CUS_ID
        WHERE o.ORD_AMOUNT >= 3000
        GROUP BY c.CUS_GENDER;
        """

        # Execute the query
        cursor.execute(query)

        # Fetch the results
        results = cursor.fetchall()

        # Print the results
        for row in results:
            print(f"Gender: {row[0]}, Number of Customers: {row[1]}")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Gender: M, Number of Customers: 1
Gender: F, Number of Customers: 1
MySQL connection is closed
```

1.2.8 Q4) Display all the order along with product name ordered by a customer having Customer_Id=2;

```
[42]: try:
          connection = mysql.connector.connect(
              host='localhost',
              user='root',
              password='',
              database='e_commerce'
          )

          if connection.is_connected():
              cursor = connection.cursor()

              # SQL query to get all orders along with product name for a customer␣
       ↪with Customer_Id=2
              query = """
              SELECT o.ORD_ID, o.ORD_AMOUNT, o.ORD_DATE, p.PRO_NAME
              FROM orders o
              JOIN product_details pd ON o.PROD_ID = pd.PROD_ID
              JOIN product p ON pd.PRO_ID = p.PRO_ID
              WHERE o.CUS_ID = 2;
              """

              # Execute the query
              cursor.execute(query)

              # Fetch the results
              results = cursor.fetchall()

              # Print the results
              for row in results:
                  print(f"Order ID: {row[0]}, Order Amount: {row[1]}, Order Date:␣
       ↪{row[2]}, Product Name: {row[3]}")

      except Error as e:
          print(f"Error: '{e}'")

      finally:
          if connection.is_connected():
              cursor.close()
              connection.close()
              print("MySQL connection is closed")
```

```
Order ID: 50, Order Amount: 2000, Order Date: 2021-10-06, Product Name: GTA V
MySQL connection is closed
```

1.2.9 Q5) Display the Supplier details who can supply more than one product.

[43]:

```python
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )

    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get suppliers who can supply more than one product
        query = """
        SELECT s.SUPP_ID, s.SUPP_NAME, s.SUPP_CITY, s.SUPP_PHONE
        FROM supplier s
        JOIN product_details pd ON s.SUPP_ID = pd.SUPP_ID
        GROUP BY s.SUPP_ID, s.SUPP_NAME, s.SUPP_CITY, s.SUPP_PHONE
        HAVING COUNT(pd.PROD_ID) > 1;
        """

        cursor.execute(query)
        results = cursor.fetchall()

        for row in results:
            print(f"Supplier ID: {row[0]}, Supplier Name: {row[1]}, Supplier City: {row[2]}, Supplier Phone: {row[3]}")

except Error as e:
    print(f"Error: '{e}'")

finally:
    if connection.is_connected():
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
```

```
Supplier ID: 1, Supplier Name: Rajesh Retails, Supplier City: Delhi, Supplier
Phone: 1234567890
MySQL connection is closed
```

1.2.10 Q6) Find the category of the product whose order amount is minimum.

[47]:

```python
# Continue from the previous connection setup...
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get the category of the product with the minimum order␣
 ↪amount
        query = """
        SELECT c.CAT_NAME
        FROM orders o
        JOIN product_details pd ON o.PROD_ID = pd.PROD_ID
        JOIN product p ON pd.PRO_ID = p.PRO_ID
        JOIN category c ON p.CAT_ID = c.CAT_ID
        ORDER BY o.ORD_AMOUNT ASC
        LIMIT 1;
        """

        cursor.execute(query)
        result = cursor.fetchone()

        print(f"Category of the product with the minimum order amount:␣
 ↪{result[0]}")

except Error as e:
    print(f"Error: '{e}'")
```

```
Category of the product with the minimum order amount: GROCERIES
```

1.2.11 Q7) Display the Id and Name of the Product ordered after "2021-10-05".

[48]:
```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get the Id and Name of the Product ordered after␣
 ↪2021-10-05
        query = """
        SELECT p.PRO_ID, p.PRO_NAME
        FROM orders o
        JOIN product_details pd ON o.PROD_ID = pd.PROD_ID
        JOIN product p ON pd.PRO_ID = p.PRO_ID
        WHERE o.ORD_DATE > '2021-10-05';
        """

        cursor.execute(query)
        results = cursor.fetchall()

        for row in results:
            print(f"Product ID: {row[0]}, Product Name: {row[1]}")

except Error as e:
    print(f"Error: '{e}'")
```

```
Product ID: 4, Product Name: OATS
Product ID: 1, Product Name: GTA V
```

1.2.12 Q8) Print the top 3 supplier name and id and rating on the basis of their rating along with the customer name who has given the rating.

[49]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get the top 3 suppliers by rating along with customer␣
 ↪name
        query = """
        SELECT s.SUPP_ID, s.SUPP_NAME, r.RAT_RATSTARS, c.CUS_NAME
        FROM rating r
        JOIN supplier s ON r.SUPP_ID = s.SUPP_ID
        JOIN customer c ON r.CUS_ID = c.CUS_ID
        ORDER BY r.RAT_RATSTARS DESC
        LIMIT 3;
        """

        cursor.execute(query)
        results = cursor.fetchall()

        for row in results:
            print(f"Supplier ID: {row[0]}, Supplier Name: {row[1]}, Rating:␣
 ↪{row[2]}, Customer Name: {row[3]}")
```

```
except Error as e:
    print(f"Error: '{e}'")
```

```
Supplier ID: 1, Supplier Name: Rajesh Retails, Rating: 5, Customer Name: PULKIT
Supplier ID: 2, Supplier Name: Appario Ltd., Rating: 4, Customer Name: AMAN
Supplier ID: 5, Supplier Name: Mittal Ltd., Rating: 4, Customer Name: MEGHA
```

1.2.13 Q9) Display customer name and gender whose names start or end with character 'A'.

[50]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get customer name and gender whose names start or end␣
  ↪with 'A'
        query = """
        SELECT CUS_NAME, CUS_GENDER
        FROM customer
        WHERE CUS_NAME LIKE 'A%' OR CUS_NAME LIKE '%A';
        """

        cursor.execute(query)
        results = cursor.fetchall()

        for row in results:
            print(f"Customer Name: {row[0]}, Customer Gender: {row[1]}")

except Error as e:
    print(f"Error: '{e}'")
```

```
Customer Name: AAKASH, Customer Gender: M
Customer Name: AMAN, Customer Gender: M
Customer Name: NEHA, Customer Gender: F
Customer Name: MEGHA, Customer Gender: F
```

1.2.14 Q10) Display the total order amount of the male customers.

[51]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get the total order amount of male customers
        query = """
        SELECT SUM(o.ORD_AMOUNT) AS TotalOrderAmount
        FROM orders o
        JOIN customer c ON o.CUS_ID = c.CUS_ID
        WHERE c.CUS_GENDER = 'M';
        """

        cursor.execute(query)
        result = cursor.fetchone()

        print(f"Total Order Amount of Male Customers: {result[0]}")

except Error as e:
    print(f"Error: '{e}'")
```

```
Total Order Amount of Male Customers: 34500
```

1.2.15 Q11) Display all the Customers left outer join with the orders

[52]:

```
try:
    connection = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='e_commerce'
    )
    if connection.is_connected():
        cursor = connection.cursor()

        # SQL query to get all customers with left outer join to orders
        query = """
        SELECT c.*, o.ORD_ID, o.ORD_AMOUNT, o.ORD_DATE, o.PROD_ID
        FROM customer c
        LEFT JOIN orders o ON c.CUS_ID = o.CUS_ID;
        """
        cursor.execute(query)
        results = cursor.fetchall()

        for row in results:
            print(row)

except Error as e:
    print(f"Error: '{e}'")
```

```
(1, 'AAKASH', '9999999999', 'DELHI', 'M', 26, 2000, datetime.date(2021, 10, 5),
1)
(2, 'AMAN', '9785463215', 'NOIDA', 'M', 50, 2000, datetime.date(2021, 10, 6), 1)
(3, 'NEHA', '9999999998', 'MUMBAI', 'F', 20, 1500, datetime.date(2021, 10, 12),
5)
(4, 'MEGHA', '9994562399', 'KOLKATA', 'F', 30, 3500, datetime.date(2021, 8, 16),
3)
(5, 'PULKIT', '7895999999', 'LUCKNOW', 'M', 25, 30500, datetime.date(2021, 9,
16), 2)
```

NOTE: Always close an open connection once you are done with the database operations

1.3 Happy Learning:)