```python
import mysql.connector

mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    password = "pass123"
)

print(mydb)
```

```
<mysql.connector.connection.MySQLConnection object at
0x000002793F6FF470>
```

```python
mycursor = mydb.cursor()

mycursor.execute("CREATE DATABASE DMart")
mycursor.execute ("USE DMart")

print(" Database created succesfully")
```

```
 Database created succesfully
```

```python
create_table = """
    CREATE TABLE DMartSales (
    Order_ID VARCHAR(50),
    Order_Date DATE,
    Customer_name VARCHAR(100),
    Country VARCHAR(50),
    State VARCHAR(100),
    City VARCHAR(100),
    Region VARCHAR(50),
    Segment VARCHAR(50),
    Ship_mode VARCHAR(50),
    Category VARCHAR(50),
    Sub_category VARCHAR(50),
    Product_name VARCHAR(100),
    Discount DECIMAL(10,2),
    Sales DECIMAL(10,2),
    Profit DECIMAL(10,2),
    Quantity INT,
    Feedback VARCHAR(20)
)
"""
mycursor.execute(create_table)

print ("Table Created Successfully")
```

```
Table Created Successfully
```

```python
create_sp = """
CREATE PROCEDURE sp_InsertSalesData (
    IN Order_ID VARCHAR(50),
    IN Order_Date DATE,
    IN Customer_name VARCHAR(100),
    IN Country VARCHAR(50),
    IN State VARCHAR(100),
    IN City VARCHAR(100),
    IN Region VARCHAR(50),
    IN Segment VARCHAR(50),
    IN Ship_mode VARCHAR(50),
    IN Category VARCHAR(50),
    IN Sub_category VARCHAR(50),
    IN Product_name VARCHAR(100),
    IN Discount DECIMAL(10,2),
    IN Sales DECIMAL(10,2),
    IN Profit DECIMAL(10,2),
    IN Quantity INT,
    IN Feedback VARCHAR(20)
)
BEGIN
    INSERT INTO DMartSales (Order_ID, Order_Date, Customer_name,
Country, State, City, Region, Segment, Ship_mode, Category,
Sub_category, Product_name, Discount, Sales, Profit, Quantity,
Feedback)
    VALUES (Order_ID, Order_Date, Customer_name, Country, State, City,
Region, Segment, Ship_mode, Category, Sub_category, Product_name,
Discount, Sales, Profit, Quantity, Feedback);
END;
"""
mycursor.execute(create_sp)

print("Procedure created successfully")

----------------------------------------------------------------------
-----
ProgrammingError                          Traceback (most recent call
last)
Cell In[39], line 26
      1 create_sp = """
      2 CREATE PROCEDURE sp_InsertSalesData (
      3     IN Order_ID VARCHAR(50),
   (...)
     24 END;
     25 """
---> 26 mycursor.execute(create_sp)
     28 print("Procedure created successfully")

File ~\AppData\Roaming\Python\Python312\site-packages\mysql\connector\
cursor.py:537, in MySQLCursor.execute(self, operation, params, multi)
    534     return
self._execute_iter(self._connection.cmd_query_iter(stmt))
```

```
    536 try:
--> 537     self._handle_result(self._connection.cmd_query(stmt))
    538 except InterfaceError as err:
    539     if self._connection.have_next_result:

File ~\AppData\Roaming\Python\Python312\site-packages\mysql\connector\
opentelemetry\context_propagation.py:97, in
with_context_propagation.<locals>.wrapper(cnx, *args, **kwargs)
     95 # pylint: disable=possibly-used-before-assignment
     96 if not OTEL_ENABLED or not cnx.otel_context_propagation:
---> 97     return method(cnx, *args, **kwargs)
     99 current_span = trace.get_current_span()
    100 tp_header = None

File ~\AppData\Roaming\Python\Python312\site-packages\mysql\connector\
connection.py:872, in MySQLConnection.cmd_query(self, query, raw,
buffered, raw_as_string)
    870 query = bytes(packet)
    871 try:
--> 872     result =
self._handle_result(self._send_cmd(ServerCmd.QUERY, query))
    873 except ProgrammingError as err:
    874     if err.errno == 3948 and "Loading local data is disabled"
in err.msg:

File ~\AppData\Roaming\Python\Python312\site-packages\mysql\connector\
connection.py:648, in MySQLConnection._handle_result(self, packet)
    646     return self._handle_eof(packet)
    647 if packet[4] == 255:
--> 648     raise get_exception(packet)
    650 # We have a text result set
    651 column_count = self._protocol.parse_column_count(packet)

ProgrammingError: 1304 (42000): PROCEDURE sp_InsertSalesData already
exists

pip install pandas sqlalchemy

Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: pandas in d:\anaconda\lib\site-packages
(2.2.2)
Requirement already satisfied: sqlalchemy in d:\anaconda\lib\site-
packages (2.0.30)
Requirement already satisfied: numpy>=1.26.0 in d:\anaconda\lib\site-
packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in d:\anaconda\
lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in d:\anaconda\lib\site-
packages (from pandas) (2024.1)
```

```
Requirement already satisfied: tzdata>=2022.7 in d:\anaconda\lib\site-
packages (from pandas) (2023.3)
Requirement already satisfied: typing-extensions>=4.6.0 in d:\
anaconda\lib\site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in d:\anaconda\lib\
site-packages (from sqlalchemy) (3.0.1)
Requirement already satisfied: six>=1.5 in d:\anaconda\lib\site-
packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
pip install pymysql
```

```
Defaulting to user installation because normal site-packages is not
writeable
Requirement already satisfied: pymysql in c:\users\kiit\appdata\
roaming\python\python312\site-packages (1.1.1)
Note: you may need to restart the kernel to use updated packages.
```

```
from sqlalchemy import create_engine
import pandas as pd
```

```
engine =
create_engine('mysql+pymysql://root:pass123@localhost:3306/DMart')
```

```
df = pd.read_excel("C:/Users/KIIT/Downloads/DMart Data
Store.xlsx",header=0)
```

```
df.head()
```

```
          Order ID Order Date     Customer Name         Country  \
0  BN-2011-7407039 2011-01-01        Ruby Patel          Sweden
1  AZ-2011-9050313 2011-01-03    Summer Hayward  United Kingdom
2  AZ-2011-6674300 2011-01-04  Devin Huddleston          France
3  BN-2011-2819714 2011-01-04       Mary Parker  United Kingdom
4  BN-2011-2819714 2011-01-04       Mary Parker  United Kingdom

                  State        City   Region      Segment     Ship
Mode  \
0             Stockholm   Stockholm    North  Home Office  Economy
Plus
1               England   Southport    North     Consumer
Economy
2  Auvergne-Rhône-Alpes     Valence  Central     Consumer
Economy
3               England  Birmingham    North    Corporate
Economy
4               England  Birmingham    North    Corporate
Economy

          Category Sub-Category                           Product Name
\
0  Office Supplies        Paper            Enermax Note Cards, Premium

1        Furniture    Bookcases     Dania Corner Shelving, Traditional

2  Office Supplies          Art  Binney & Smith Sketch Pad, Easy-Erase

3  Office Supplies          Art             Boston Markers, Easy-Erase

4  Office Supplies      Storage            Eldon Folders, Single Width


   Discount  Sales  Profit  Quantity  Feedback?
0       0.5     45     -26         3      False
1       0.0    854     290         7       True
2       0.0    140      21         3       True
3       0.5     27     -22         2       True
4       0.5     17      -1         2       True
```

```
for _, row in df.iterrows():
        order_date = row['Order Date'].strftime('%Y-%m-%d') if
isinstance(row['Order Date'], pd.Timestamp) else row['Order Date']

        mycursor.callproc("sp_InsertSalesData", [
            row['Order ID'],
            order_date,
            row['Customer Name'],
            row['Country'],
            row['State'],
            row['City'],
            row['Region'],
            row['Segment'],
            row['Ship Mode'],
            row['Category'],
            row['Sub-Category'],
            row['Product Name'],
            row['Discount'],
            row['Sales'],
            row['Profit'],
            row['Quantity'],
            row['Feedback?']
        ])
mydb.commit()

print('Data stored successfully')
```

```
Data stored successfully
```

```
get_proc = """

CREATE PROCEDURE sp_GetSalesData()
BEGIN
```

```
  SELECT * FROM DMartSales;
END;

"""
mycursor.execute(get_proc)
```

```
print("Procedure created successfully")
```

```
Procedure created successfully
```

```
mycursor.callproc("sp_GetSalesData")
```

```
print("Get procedure created successfully")
```

```
Get procedure created successfully
```

```
for result in mycursor.stored_results():
    df = result.fetchall()
    columns = result.column_names
```

```
data = pd.DataFrame(data, columns=columns)
data.to_csv("salesDMart.csv", index=False)
print("Data fetched and saved to 'sales_DMart.csv'.")
```

```
Data fetched and saved to 'sales_DMart.csv'.
```

```
mycursor.close()
mydb.close()
```

```
print("All steps completed successfully.")
```

```
All steps completed successfully.
```