



Integration Between SQL and Python

BY Shrouk Hassan



Using Northwind Database and visual studio code as Jupyter Notebook.

Import Python library(`pyodbc`) to connect to databases .

```
import pyodbc

# إعداد الاتصال
conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=DESKTOP-3SLUOLI\SQLEXPRESS;' # اسم الميرفر # localhost
    'DATABASE=Northwind;' # اسم قاعدة البيانات
    'Trusted_Connection=yes;' # استخدام Windows Authentication
)

# اختبار الاتصال
cursor = conn.cursor()
cursor.execute("SELECT @@VERSION") # يعرف إصدار SQL Server
for row in cursor:
    print(row)

# إغلاق الاتصال بعد الانتهاء
conn.close()

✓ 0.0s
```

Importing Libraries, Connecting to the Database and Listing All Tables and Printing the Tables.
The northwind database is commonly used for sample data, often containing information like orders, customers, products, and other business-related data.

- ✓

```
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
# Connect to the Northwind database
conn = sqlite3.connect('northwind.db')

# List all tables in the database
tables = pd.read_sql_query("SELECT name FROM sqlite_master WHERE type='table';", conn)
print(tables)
```

✓ 0.0s

Order Analysis by Year and Month

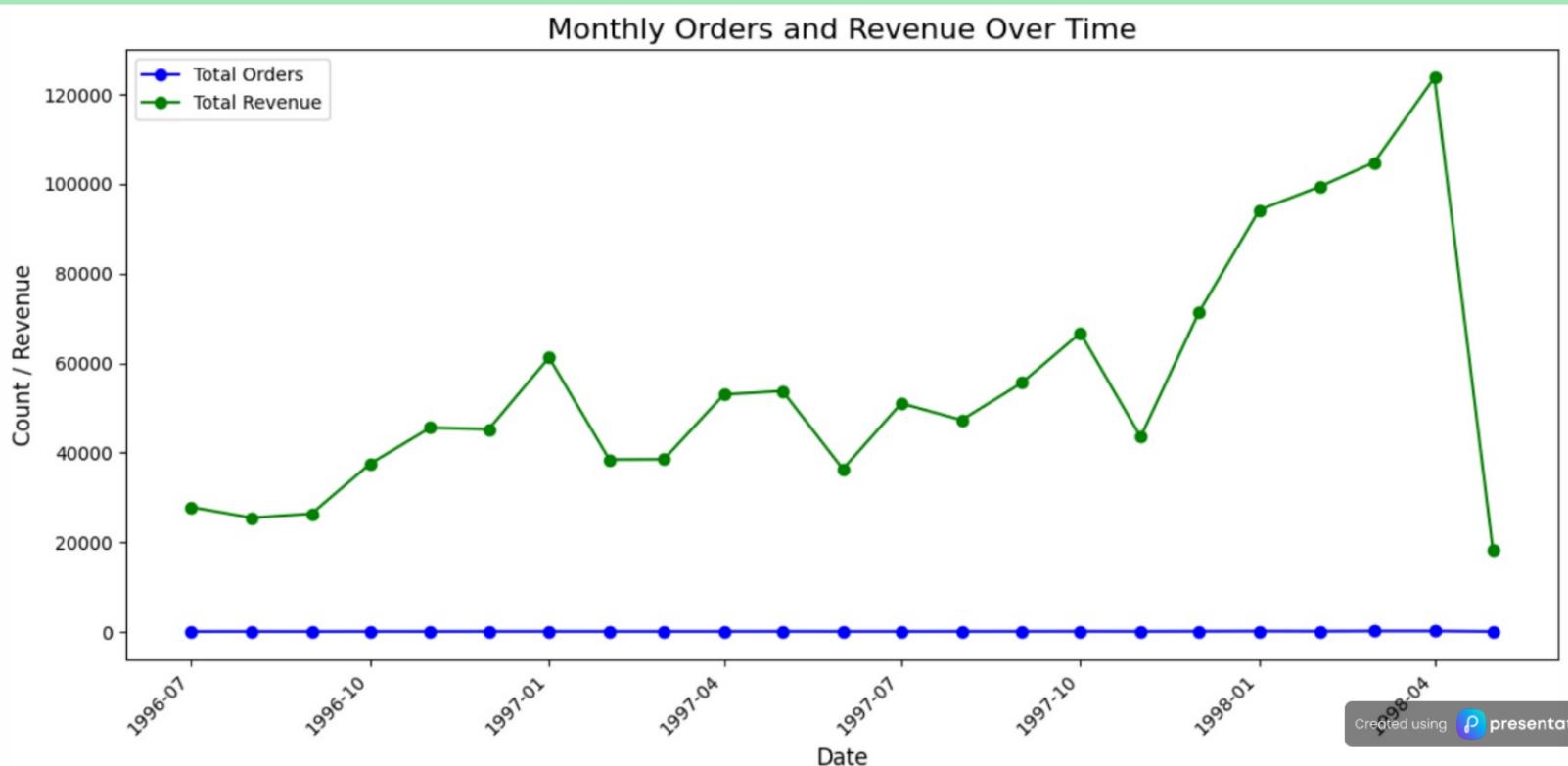
```
#1
conn = connect_to_sql_server()
query = """
SELECT YEAR(OrderDate) AS OrderYear, MONTH(OrderDate) AS OrderMonth,
COUNT(o.OrderID) AS TotalOrders, SUM(OD.UnitPrice * OD.Quantity * (1 - OD.Discount)) AS TotalRevenue
FROM Orders O
JOIN [Order Details] OD ON O.OrderID = OD.OrderID
GROUP BY YEAR(OrderDate), MONTH(OrderDate)
ORDER BY OrderYear, OrderMonth;
"""
# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
    # عرض أول 5 سجلات
else:
    print("لم يتم العثور على بيانات")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
✓ 0.0s
```

OrderYear	OrderMonth	TotalOrders	TotalRevenue
1996	7	59	27861.894974
1996	8	69	25485.274970
1996	9	57	26381.399973
1996	10	73	37515.725197
1996	11	66	45600.044844

```
if 'YearMonth' not in df.columns:
    df['YearMonth'] = pd.to_datetime(df['OrderYear'].astype(str) + '-' + df['OrderMonth'].astype(str), errors='coerce')
plt.figure(figsize=(12, 6))
plt.plot(df['YearMonth'], df['TotalOrders'], label='Total Orders', marker='o', linestyle='-', color='blue')
plt.plot(df['YearMonth'], df['TotalRevenue'], label='Total Revenue', marker='o', linestyle='-', color='green')
plt.title('Monthly Orders and Revenue Over Time', fontsize=16)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Count / Revenue', fontsize=12)
plt.legend(fontsize=10)
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
✓ 0.6s
```



visualization





Active Customer Analysis and output

```
COUNT(O.OrderID) AS TotalOrders  
FROM  
    Customers C  
JOIN  
    Orders O ON C.CustomerID = O.CustomerID  
WHERE  
    O.OrderDate >= DATEADD(YEAR, -1, GETDATE())  
GROUP BY  
    C.CustomerID, C.CompanyName  
HAVING  
    COUNT(O.OrderID) > 10  
ORDER BY  
    TotalOrders DESC;  
***  
  
# قراءة البيانات من SQL Server  
df = read_data_from_sql(conn, query)  
if df is not None:  
    # عرض البيانات  
    print(df.head(5)) # لبيانات 5 أول سجل
```

Empty DataFrame

Columns: [CustomerID, CompanyName, TotalOrders]

Index: []

تم إغلاق الاتصال بنجاح.

C:\Users\HP\AppData\Local\Temp\ipykernel_9960\26

```
df = pd.read_sql(query, conn)
```



Low Stock Analysis

```
conn = connect_to_sql_server()
query = """
    SELECT
        ProductID,
        ProductName,
        UnitsInStock,
        ReorderLevel
    FROM
        Products
    WHERE
        UnitsInStock < (SELECT AVG(ReorderLevel) FROM Products);
"""
# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)

if df is not None:
    # عرض البيانات
    print(df.head())
    # عرض أول 5 سجلات
else:
    # لم يتم العثور على بيانات
    print("لم يتم العثور على بيانات")
    # إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

✓ 0.0s

ProductID	ProductName	UnitsInStock	ReorderLevel
5	Chef Anton's Gumbo Mix	0	0
8	Northwoods Cranberry Sauce	6	0
17	Alice Mutton	0	0
21	Sir Rodney's Scones	3	5
29	Thüringer Rostbratwurst	0	0

تم إغلاق الاتصال بنجاح.

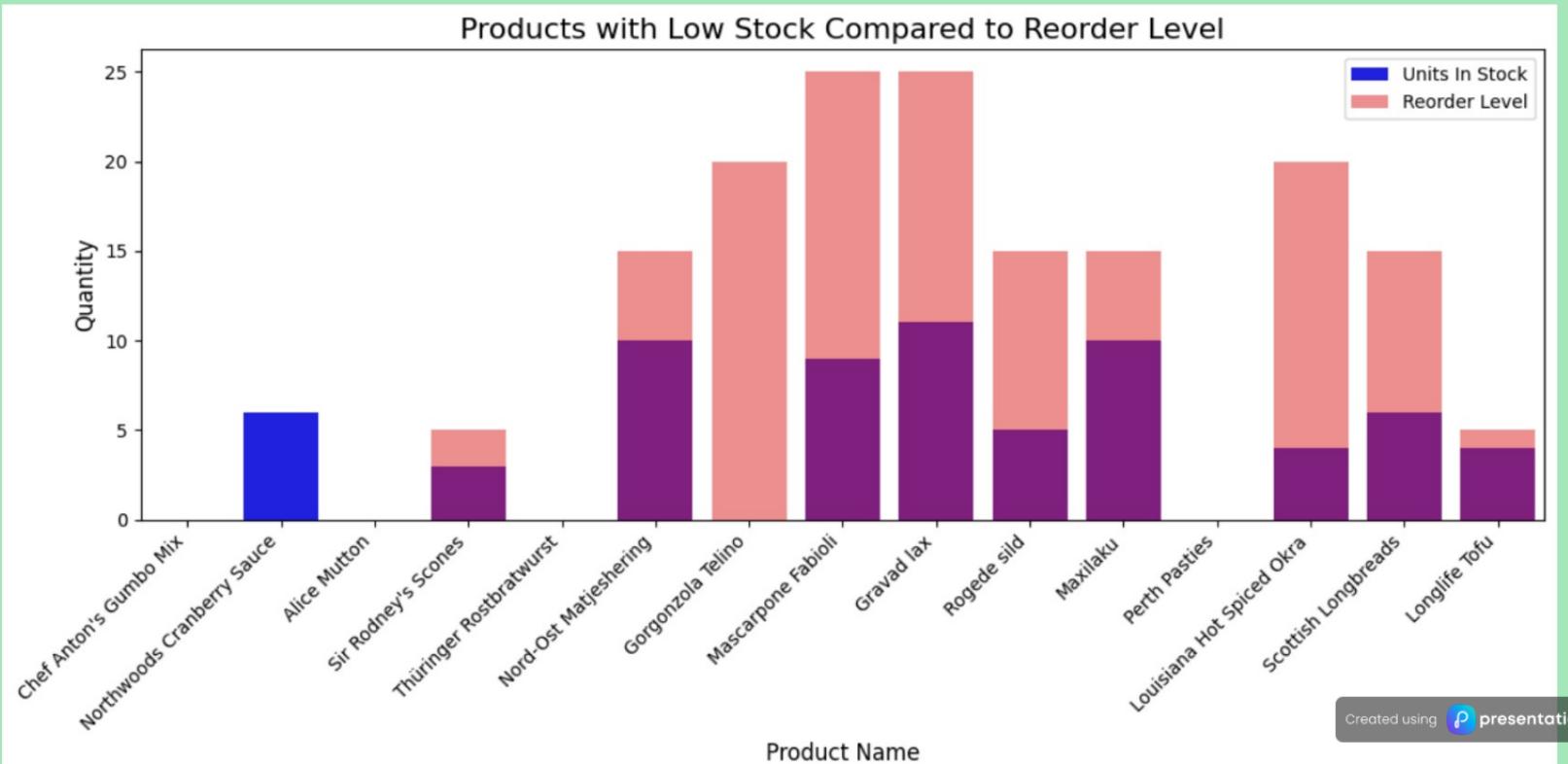
```
# Create a bar plot for UnitsInStock vs. ReorderLevel
plt.figure(figsize=(12, 6))
sns.barplot(data=df, x='ProductName', y='UnitsInStock', color='blue', label='Units In Stock')
sns.barplot(data=df, x='ProductName', y='ReorderLevel', color='red', label='Reorder Level', alpha=0.5)

# Customize the plot
plt.title('Products with Low Stock Compared to Reorder Level', fontsize=16)
plt.xlabel('Product Name', fontsize=12)
plt.ylabel('Quantity', fontsize=12)
plt.xticks(rotation=45, ha='right')
plt.legend()

# Show the plot
plt.tight_layout()
plt.show()
```



visualization





Supplier Analysis by Location

```
conn = connect_to_sql_server()
    كتابة الاستعلام
query = """
SELECT
    s.SupplierID,
    s.CompanyName,
    s.City,
    s.Country,
    COUNT(p.ProductID) AS ProductCount
FROM
    Suppliers s
LEFT JOIN
    Products p ON s.SupplierID = p.SupplierID
GROUP BY
    s.SupplierID, s.CompanyName, s.City, s.Country
ORDER BY
    s.Country, s.City;
"""

df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head()) # عرض أول 5 سجلات
else:
    print("لم يتم العثور على بيانات")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
/
0.0s
```

	SupplierID	CompanyName	City	Country
0	7	Pavlova, Ltd.	Melbourne	Australia
1	24	G'day, Mate	Sydney	Australia
2	10	Refrescos Americanas LTDA	Sao Paulo	Brazil
3	25	Ma Maison	Montréal	Canada
4	29	Forêts d'éables	Ste-Hyacinthe	Canada
	ProductCount			
0	5			
1	3			
2	1			
3	2			
4	2			

تم إغلاق الاتصال بنجاح.



visualization





Delays and output

```
conn = connect_to_sql_server()
    كتابة الاستعلام
query = """
SELECT
    o.OrderID,
    o.OrderDate,
    o.RequiredDate,
    o.ShippedDate,
    s.CompanyName AS ShipperName,
    DATEDIFF(DAY, o.OrderDate, o.ShippedDate) AS DelayInDays
FROM
    Orders o
INNER JOIN
    Shippers s ON o.ShipVia = s.ShipperID
WHERE
    DATEDIFF(DAY, o.OrderDate, o.ShippedDate) > 5;
"""

    تنفيذ الاستعلام
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
        عرض أول 5 سجلات # إغلاق الاتصال بعد الانتهاء
else:
    print("لم يتم العثور على بيانات")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

	OrderID	OrderDate	RequiredDate	ShippedDate	ShipperName	DelayInDays
0	10248	1996-07-04	1996-08-01	1996-07-16	Federal Shipping	12
1	10251	1996-07-08	1996-08-05	1996-07-15	Speedy Express	7
2	10253	1996-07-10	1996-07-24	1996-07-16	United Package	6
3	10254	1996-07-11	1996-08-08	1996-07-23	United Package	12
4	10257	1996-07-16	1996-08-13	1996-07-22	Federal Shipping	6

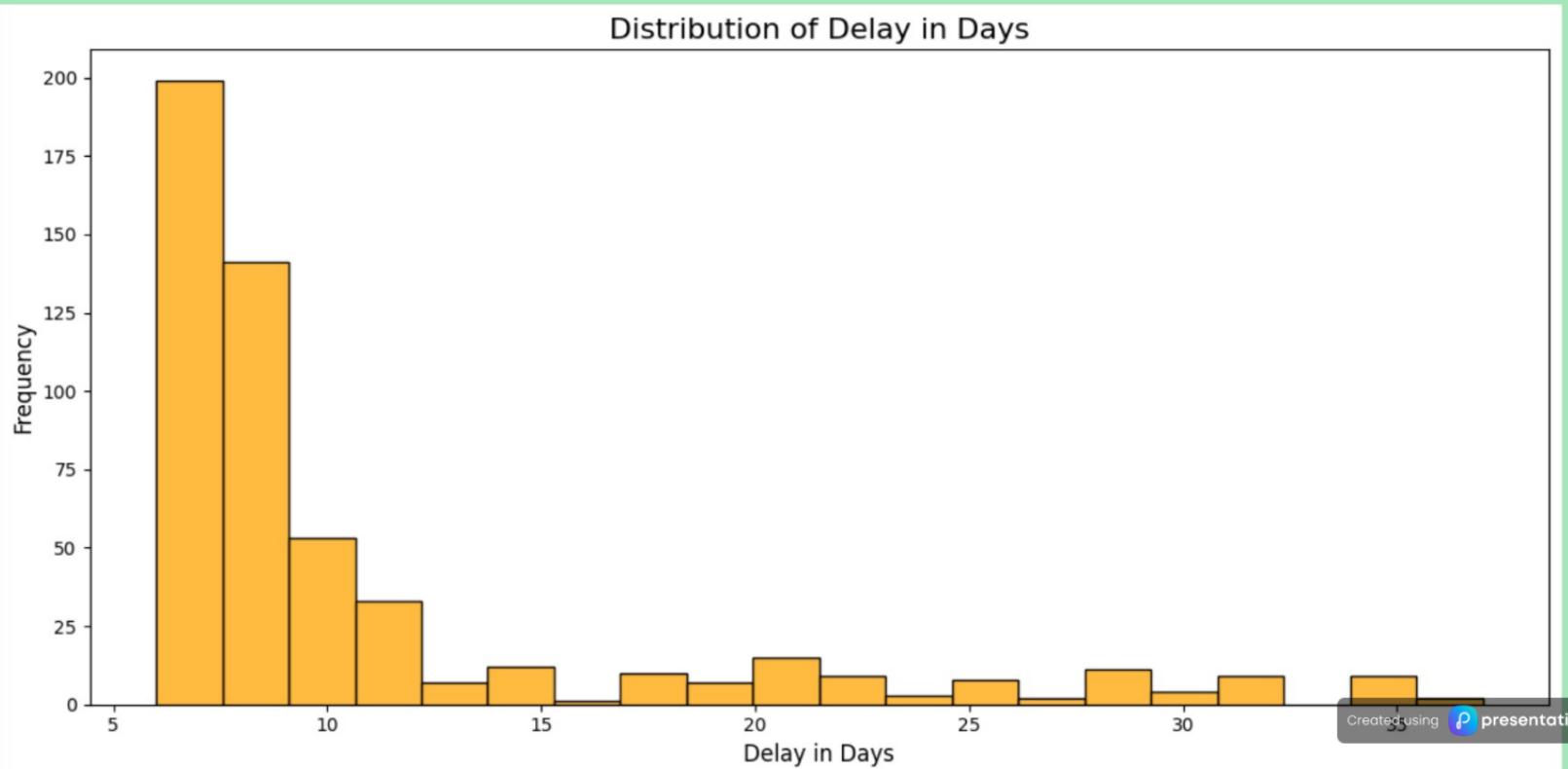
تم إغلاق الاتصال بنجاح.

C:\Users\HP\AppData\Local\Temp\ipykernel_9960\2681990858.py:27: UserWarning: pa

```
df = pd.read_sql(query, conn)
```



visualization





Profitability by Category

```
conn = connect_to_sql_server()
query = """
    SELECT
        c.CategoryName,
        AVG(p.UnitPrice) AS AverageProfitability
    FROM
        Categories c
    INNER JOIN
        Products p ON c.CategoryID = p.CategoryID
    GROUP BY
        c.CategoryName
    ORDER BY
        AverageProfitability DESC;
"""

# تنفيذ الاستعلام
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head()) # عرض أول 5 سجلات
else:
    print("لم يتم العثور على بيانات")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

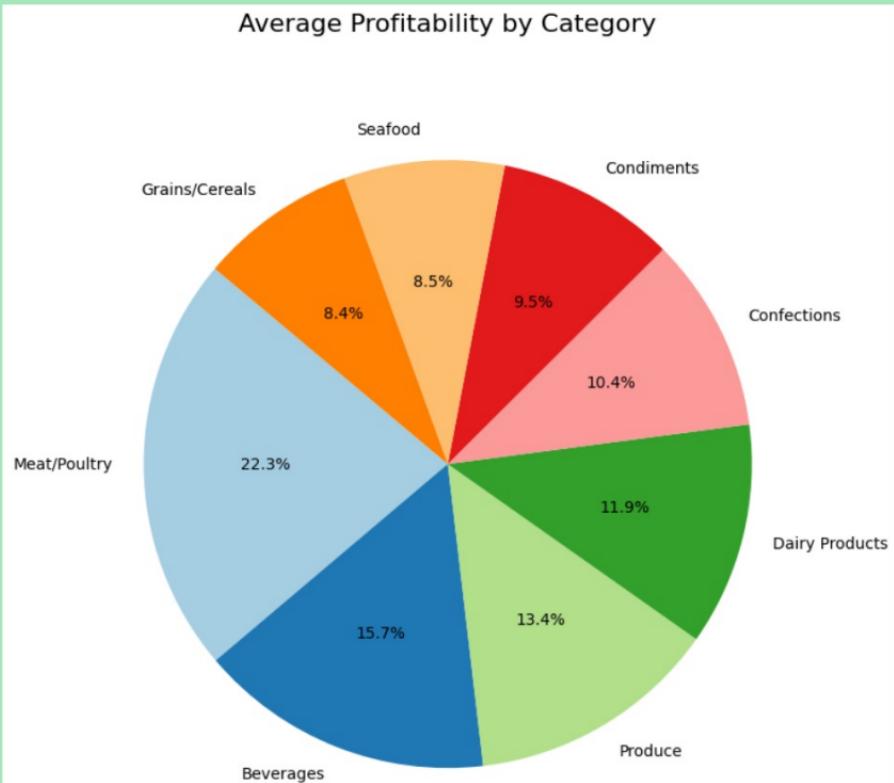
	CategoryName	AverageProfitability
0	Meat/Poultry	54.0066
1	Beverages	37.9791
2	Produce	32.3700
3	Dairy Products	28.7300
4	Confections	25.1600

تم إغلاق الاتصال بنجاح.

C:\Users\HP\OneDrive\Desktop\Learn Python\Week 1\0060126



visualization





Top Customers

```
conn = connect_to_sql_server()
# القراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT TOP 5
C.CustomerID,
C.CompanyName AS CustomerName,
SUM(OD.UnitPrice * OD.Quantity * (1 - OD.Discount)) AS TotalSales
FROM
Customers C
JOIN
Orders O ON C.CustomerID = O.CustomerID
JOIN
[Order Details] OD ON O.OrderID = OD.OrderID
GROUP BY
C.CustomerID, C.CompanyName
ORDER BY
TotalSales DESC;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
else:
    # لم يتم العثور على بيانات
    print("لم يتم العثور على بيانات")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
✓ 0.0s
```

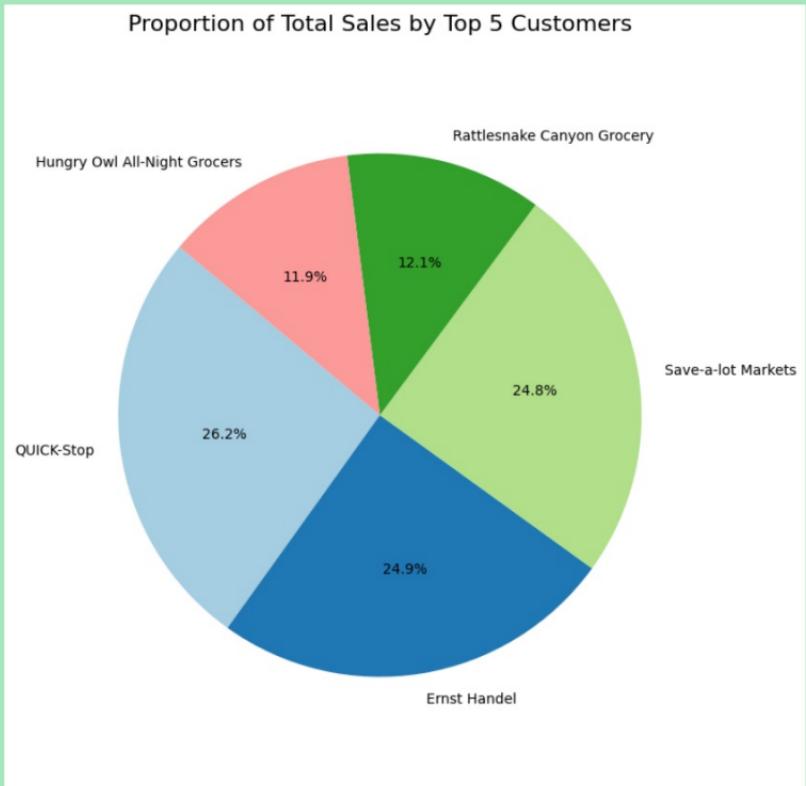
	CustomerID	CustomerName	TotalSales
0	QUICK	QUICK-Stop	110277.304977
1	ERNSH	Ernst Handel	104874.978714
2	SAVEA	Save-a-lot Markets	104361.949921
3	RATTC	Rattlesnake Canyon Grocery	51097.800333
4	HUNGO	Hungry Owl All-Night Grocers	49979.905001

تم إغلاق الاتصال بنجاح

C:\Users\HP\AppData\Local\Temp\ipykernel_9960\2681000858.py:27



visualization



Discount Analysis

```
conn = connect_to_sql_server()
# لقراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    C.CustomerID,
    C.CompanyName AS CustomerName,
    AVG(OD.Discount) * 100 AS AverageDiscountPercentage
FROM
    Customers C
JOIN
    Orders O ON C.CustomerID = O.CustomerID
JOIN
    [Order Details] OD ON O.OrderID = OD.OrderID
GROUP BY
    C.CustomerID, C.CompanyName
HAVING
    AVG(OD.Discount) > 0.1
ORDER BY
    AverageDiscountPercentage DESC;
"""

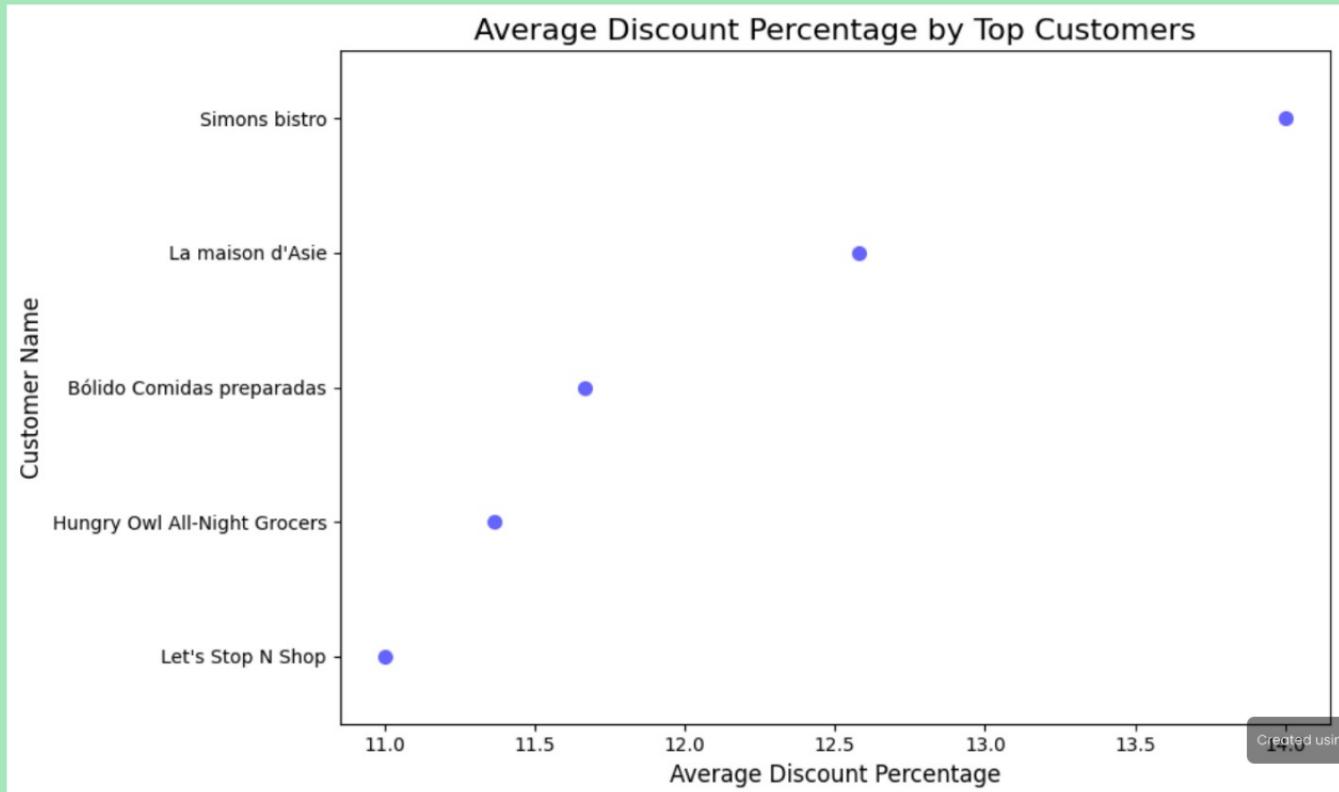
# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head()) # عرض أول 5 سجلات
else:
    # إغلاق الاتصال بعد الانتهاء
    close_connection(conn)
```

	CustomerID	CustomerName	AverageDiscountPercentage
0	SIMOB	Simons bistro	14.000000
1	LAMAI	La maison d'Asie	12.580645
2	BOLID	Bólido Comidas preparadas	11.666667
3	HUNGO	Hungry Owl All-Night Grocers	11.363636
4	LETSS	Let's Stop N Shop	11.000000

تم إغلاق الاتصال بنجاح.



visualization





Order Revenue by Employee

```
conn = connect_to_sql_server()
# قراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    E.EmployeeID,
    E.LastName + ', ' + E.FirstName AS EmployeeName,
    SUM(OD.UnitPrice * OD.Quantity * (1 - OD.Discount)) AS TotalRevenue
FROM
    Employees E
JOIN
    Orders O ON E.EmployeeID = O.EmployeeID
JOIN
    [Order Details] OD ON O.OrderID = OD.OrderID
GROUP BY
    E.EmployeeID, E.LastName, E.FirstName
ORDER BY
    TotalRevenue DESC;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
    # عرض أول 5 سجلات
else:
    print("Data not found")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

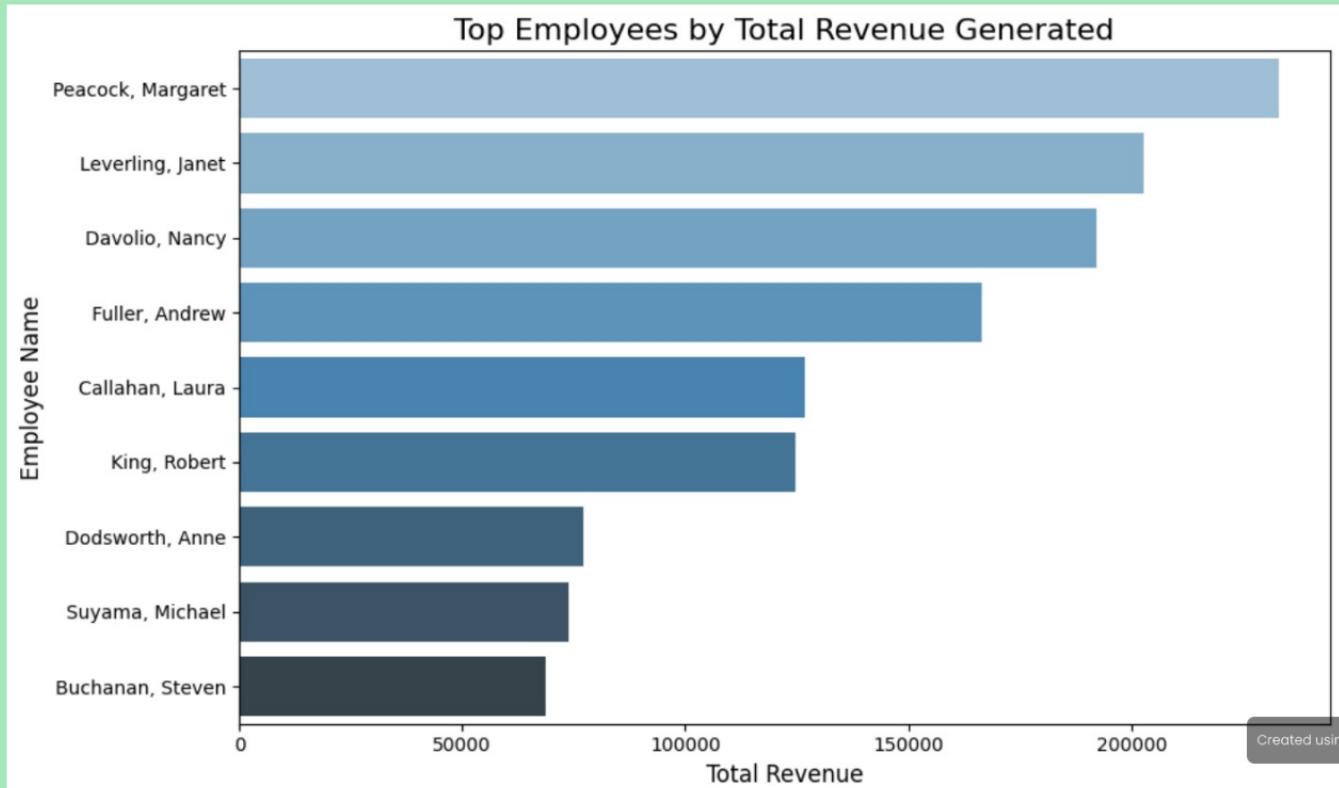
	EmployeeID	EmployeeName	TotalRevenue
0	4	Peacock, Margaret	232890.845947
1	3	Leverling, Janet	202812.842793
2	1	Davolio, Nancy	192107.604321
3	2	Fuller, Andrew	166537.754978
4	8	Callahan, Laura	126862.277704

تم إغلاق الاتصال بنجاح.

C:\Users\HP\AppData\Local\Temp\ipykernel_9060\2691009859



visualization





Best-Selling Products

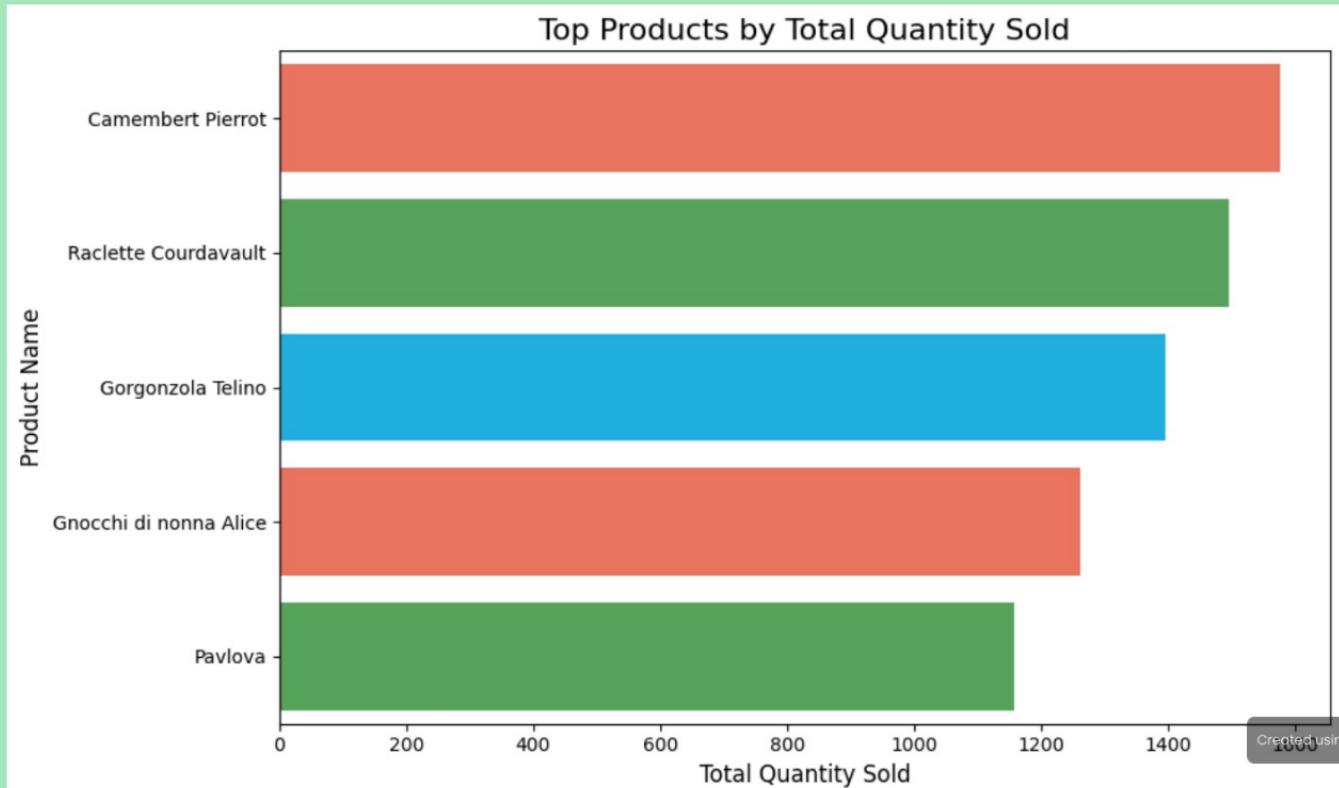
```
conn = connect_to_sql_server()
query = """
SELECT TOP 5
    P.ProductName,
    SUM(OD.Quantity) AS TotalQuantitySold
FROM
    Products P
JOIN
    [Order Details] OD ON P.ProductID = OD.ProductID
GROUP BY
    P.ProductName
ORDER BY
    TotalQuantitySold DESC;
"""
# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head()) # عرض أول 5 سجلات
else:
    # إغلاق الاتصال بعد الانتهاء
    close_connection(conn)
✓ 0.0s
```

	ProductName	TotalQuantitySold
0	Camembert Pierrot	1577
1	Raclette Courdavault	1496
2	Gorgonzola Telino	1397
3	Gnocchi di nonna Alice	1263
4	Pavlova	1158

تم إغلاق الاتصال بنجاح.



visualization





Unshipped Orders

```
conn = connect_to_sql_server()
# القراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    C.CompanyName AS CustomerName,
    O.OrderDate
FROM
    Orders O
JOIN
    Customers C ON O.CustomerID = C.CustomerID
WHERE
    O.ShippedDate IS NULL
ORDER BY
    O.OrderDate DESC;
"""

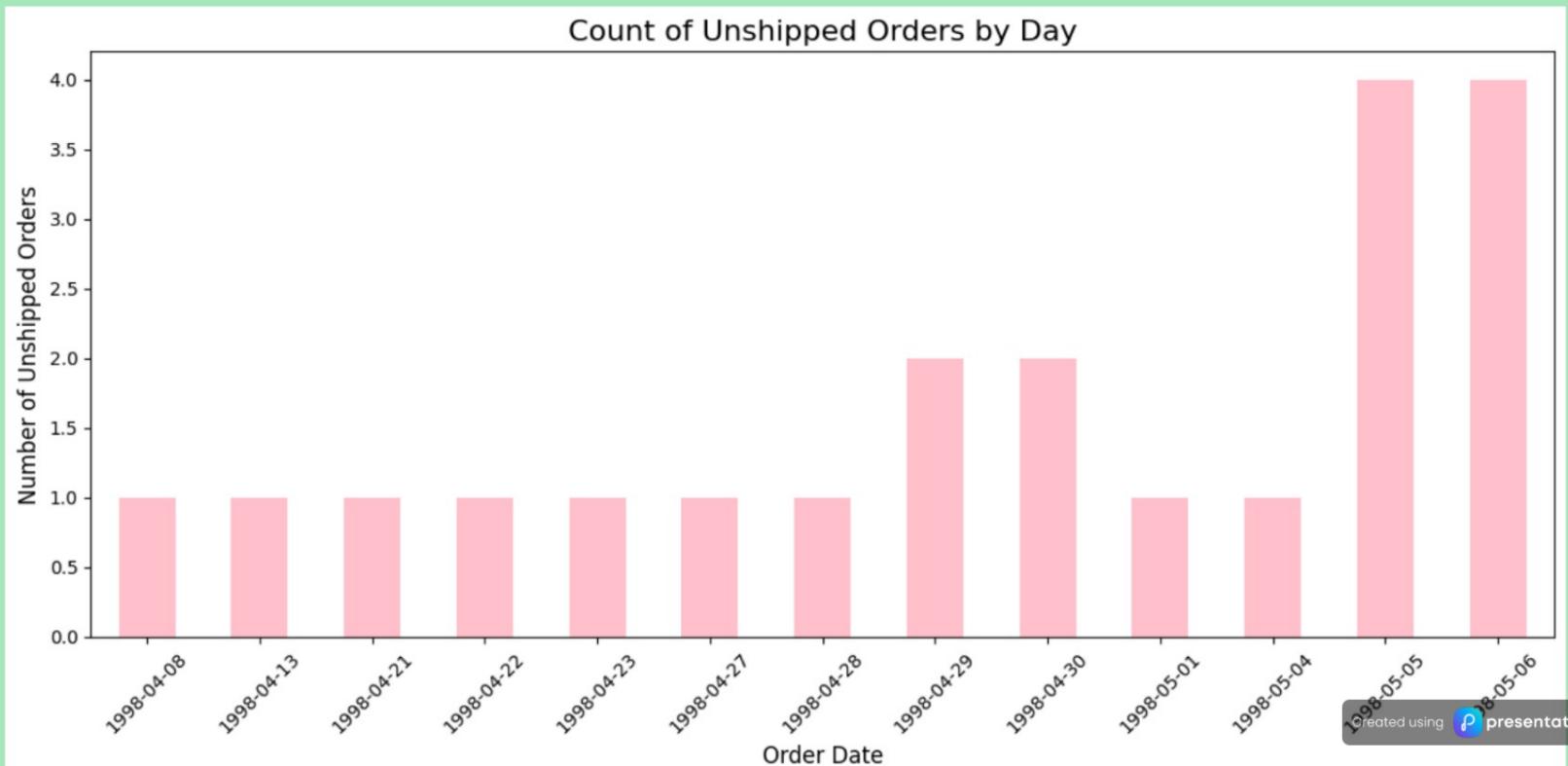
# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)

if df is not None:
    # عرض البيانات
    print(df.head())
else:
    print("Data not found")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

	CustomerName	OrderDate
0	Simons bistro	1998-05-06
1	Richter Supermarkt	1998-05-06
2	Bon app'	1998-05-06
3	Rattlesnake Canyon Grocery	1998-05-06
4	Lehmanns Marktstand	1998-05-05

تم إغلاق الاتصال بنجاح.

visualization



Sales by Region

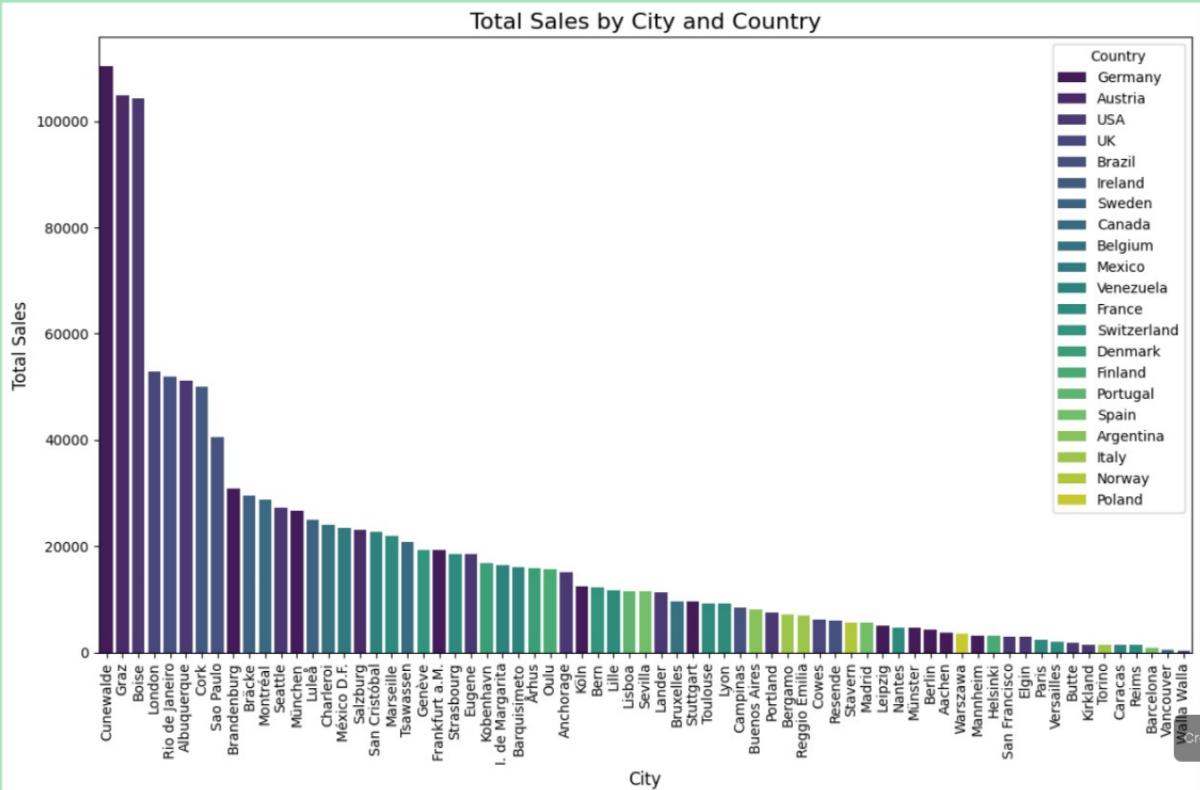
```
conn = connect_to_sql_server()
# لقراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    C.Country,
    C.City,
    SUM(OD.UnitPrice * OD.Quantity * (1 - OD.Discount)) AS TotalSales
FROM
    Orders O
JOIN
    Customers C ON O.CustomerID = C.CustomerID
JOIN
    [Order Details] OD ON O.OrderID = OD.OrderID
GROUP BY
    C.Country, C.City
ORDER BY
    TotalSales DESC;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
else:
    # إغلاق الاتصال بعد الانتهاء
    close_connection(conn)
```

	Country	City	TotalSales
0	Germany	Cunewalde	110277.304977
1	Austria	Graz	104874.978714
2	USA	Boise	104361.949921
3	UK	London	52825.010068
4	Brazil	Rio de Janeiro	51956.979933

تم إغلاق الاتصال بنجاح.

visualization





Subqueries for Supplier Analysis

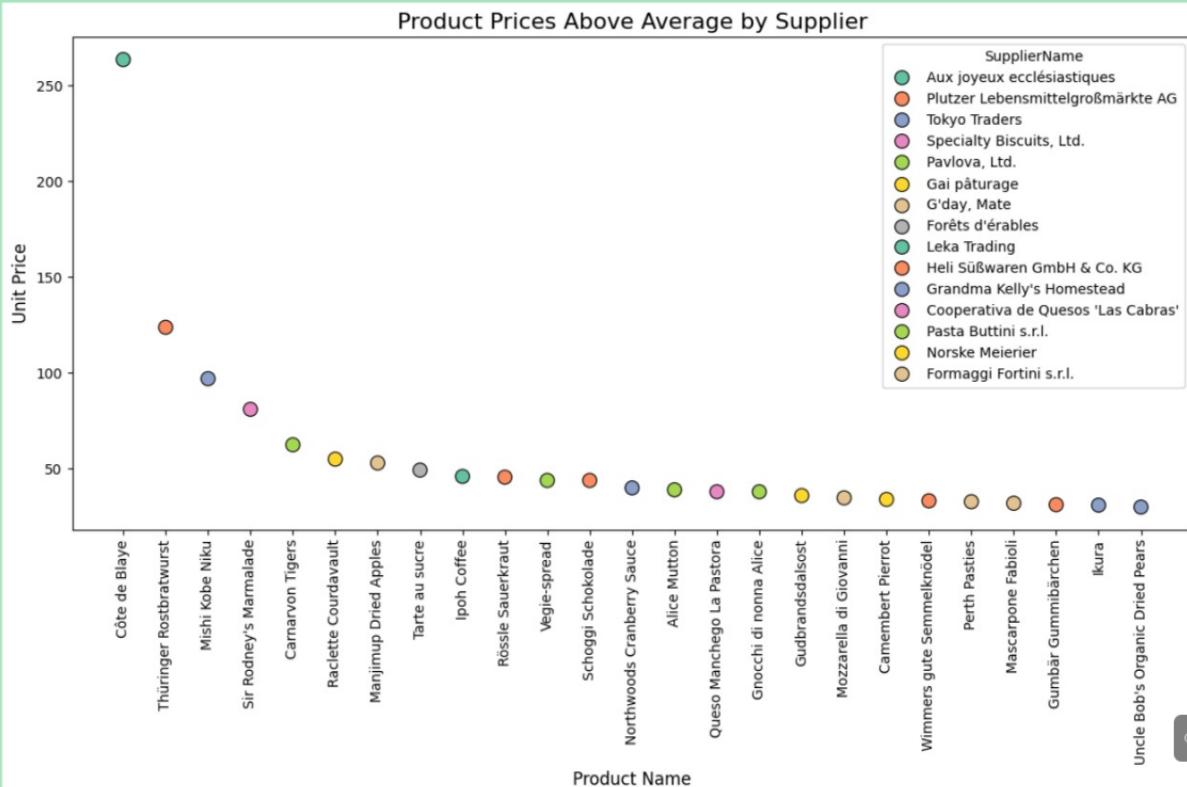
```
conn = connect_to_sql_server()
# القراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    S.SupplierID,
    S.CompanyName AS SupplierName,
    P.ProductName,
    P.UnitPrice
FROM
    Suppliers S
JOIN
    Products P ON S.SupplierID = P.SupplierID
WHERE
    P.UnitPrice > (SELECT AVG(UnitPrice) FROM Products)
ORDER BY
    P.UnitPrice DESC;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head()) # عرض أول 5 سجلات
else:
    print("Data not found")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

	SupplierID	SupplierName	ProductName \
	UnitPrice		
0	18	Aux joyeux ecclésiastiques	Côte de Blaye
1	12	Plutzer Lebensmittelgroßmärkte AG	Thüringer Rostbratwurst
2	4	Tokyo Traders	Mishi Kobe Niku
3	8	Specialty Biscuits, Ltd.	Sir Rodney's Marmalade
4	7	Pavlova, Ltd.	Carnarvon Tigers

تم إغلاق الاتصال بنجاح.

visualization





Unsold Products

```
conn = connect_to_sql_server()
# القراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    P.ProductName
FROM
    Products P
LEFT JOIN
    [Order Details] OD ON P.ProductID = OD.ProductID
WHERE
    OD.OrderID IS NULL;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())  # عرض أول 5 سجلات
else:
    print("Data not found")
# إغلاق الاتصال بعد الانتهاء
close_connection(conn)
```

Empty DataFrame
Columns: [ProductName]
Index: []
تم إغلاق الاتصال بنجاح.



Orders by Product Category

```
conn = connect_to_sql_server()
# القراءة أول 10 سجلات من جدول SQL استعلام
query = """
SELECT
    C.CategoryName,
    COUNT(DISTINCT O.OrderID) AS NumberOfOrders,
    SUM(OD.UnitPrice * OD.Quantity * (1 - OD.Discount)) AS TotalRevenue
FROM
    Categories C
JOIN
    Products P ON C.CategoryID = P.CategoryID
JOIN
    [Order Details] OD ON P.ProductID = OD.ProductID
JOIN
    Orders O ON OD.OrderID = O.OrderID
GROUP BY
    C.CategoryName
ORDER BY
    TotalRevenue DESC;
"""

# قراءة البيانات من SQL Server
df = read_data_from_sql(conn, query)
if df is not None:
    # عرض البيانات
    print(df.head())
else:
    # عرض أول 5 سجلات
    print("Data not found")
# إغلاق الاتصال بعد الاتصال
close_connection(conn)
```

	CategoryName	NumberOfOrders	TotalRevenue
0	Beverages	354	267868.180523
1	Dairy Products	303	234507.285217
2	Confections	295	167357.224832
3	Meat/Poultry	161	163022.359089
4	Seafood	291	131261.737425

تم إغلاق الاتصال بنجاح.



visualization



Thank you.

