Data Analytics Project

Python - Visualisation

Sama Nasseredine

Table des matières

What are the regional sales in the best performing country?

What we need to find: /Objectives

- The best performing country
- Then, if the country has regions, the best performing region.

- Identify the top-performing country based on relevant performance metrics.
- For countries with multiple regions, determine the highest-performing region within that country.

Required SQL tables and columns:

- Sales.SalesTerritory : TerritoryID, Name, CountryRegionCode
- Sales.SalesOrderHeader: TotalDue, TerritoryID
- Person.CountryRegion : Name, CountryRegionCode

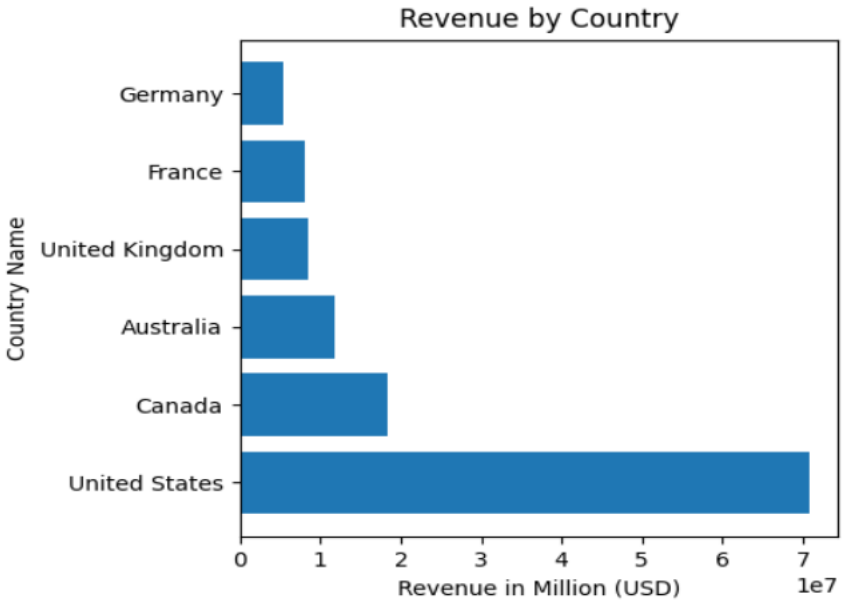
SQL Code:

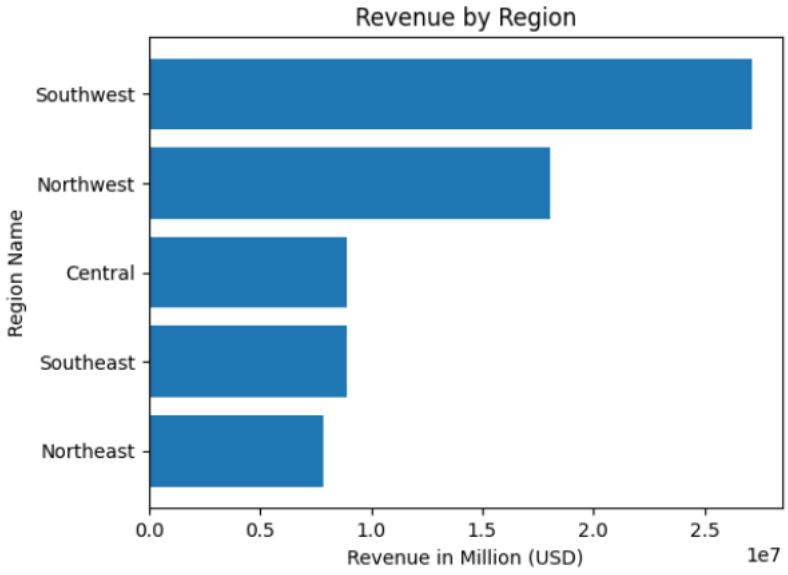
```
SELECT
  st.CountryRegionCode,
  cr.Name AS CountryName,
  SUM(soh.TotalDue) AS Revenue
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesTerritory AS st
  ON soh.TerritoryID = st.TerritoryID
JOIN Person.CountryRegion AS cr
  ON st.CountryRegionCode = cr.CountryRegionCode
GROUP BY
  st.CountryRegionCode,
  cr.Name
ORDER BY
  Revenue DESC;

SELECT
  st.Name AS RegionName,
  SUM(soh.TotalDue) AS Revenue
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesTerritory AS st
  ON soh.TerritoryID = st.TerritoryID
WHERE st.CountryRegionCode = 'US'   -- Because US is classified in regions
GROUP BY st.Name
ORDER BY Revenue DESC;
```

Python Code:

```
import pandas as pd
import matplotlib.pyplot as plt
 country = pd.read_csv('python/Q1country.csv', header = None, names = ['Name','Revenue'])#adding header
print(country )
plt.barh(country['Name'], country['Revenue'])
plt.xlabel('Revenue in Million (USD)')
plt.ylabel('Country Name')
plt.title('Revenue by Country')
plt.show()
plt.savefig('python/Q1region.png')
region = pd.read_csv('python/Q1region.csv', header = None, names = ['Name','Revenue'])  # adding header
```

```
print(region)
plt.barh(region['Name'], region['Revenue'])
plt.xlabel('Revenue')
plt.ylabel('Region Name')
plt.title('Revenue by Region')
plt.savefig('python/Q1region.png')
plt.show()
```





Conclusion:

We first calculated total revenue per country to identify the best-performing country (US). Then we filtered the data to this country and grouped by region to find which region contributed the most revenue. A horizontal bar chart was used to compare regional revenues visually, as it makes

differences between regions easy to see: the Southwest region is the region performing better in the US, and in the whole company.

What is the relationship between annual leave taken and bonus?

What we need to find: Do employees who take more annual leave earn more/less bonuses?

- HumanRessources Database
- Annual leave
- Bonus

With the tables:

- HumanRessources.Employee : VacationHours = annual leave, BusinessEntityID
- Sales.SalesPerson : Bonus, BusinessEntityID

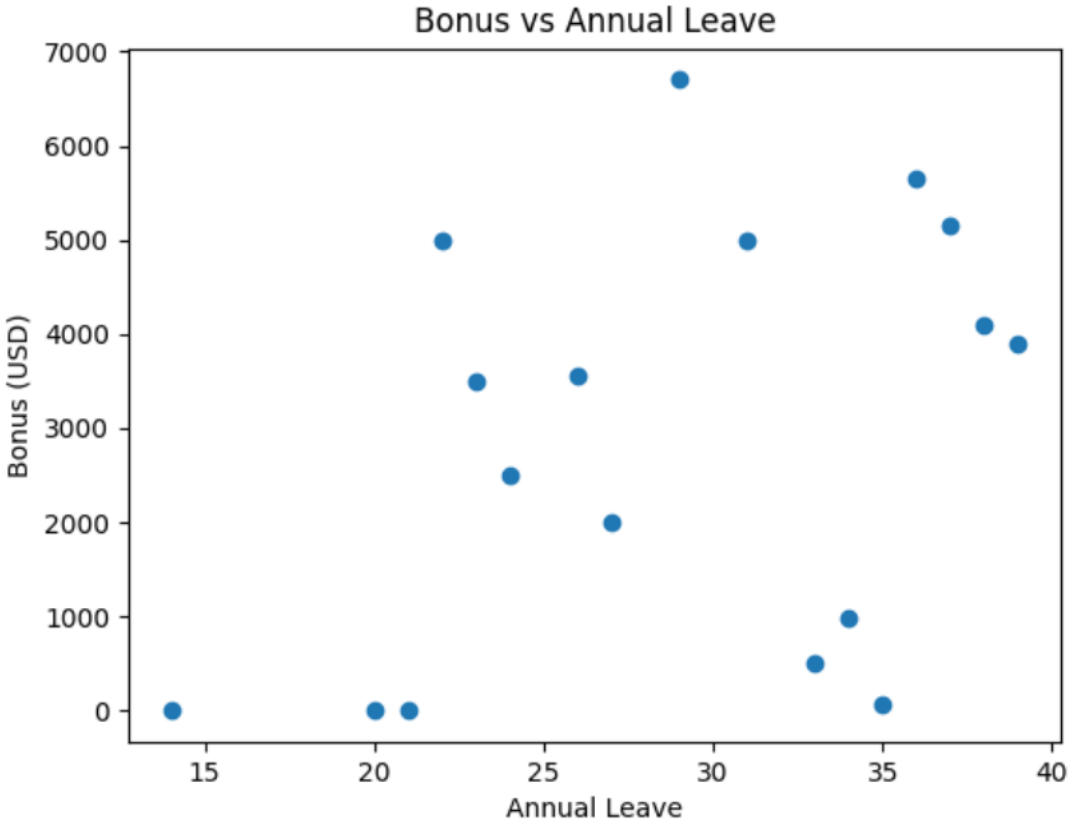
Only employees who are salespeople receive a bonus.

SQL:

```
SELECT em.VacationHours as Annual_Leave, sp.Bonus, em.BusinessEntityID
FROM HumanResources.Employee as em
JOIN Sales.SalesPerson as sp
ON sp.BusinessEntityID = em.BusinessEntityID
ORDER BY em.VacationHours ;
```

Python:

```
bonus = pd.read_csv('python/Q2bonus.csv', header = None, names = ['Annual Leave','Bonus',
'ID'])#adding header
print(bonus )
plt.scatter(bonus['Annual Leave'], bonus['Bonus'])
plt.xlabel('Annual Leave')
plt.ylabel('Bonus')
plt.title('Bonus vs Annual Leave')
plt.savefig('python/Q2bonus.png')
plt.show()
Corr = bonus[['Annual Leave', 'Bonus']]
# Print correlation between Annual Leave and Bonus
correlation_value = Corr.corr().loc['Annual Leave', 'Bonus']
print("Correlation between Annual Leave and Bonus:", correlation_value)
```



Conclusion:

We examined whether employees who take more annual leave receive higher or lower bonuses by joining HR employee data with salesperson bonus records. We then visualised the relationship using a scatter plot and calculated the correlation between annual leave and bonus amounts. The results show a correlation of only 0.30, which is very weak and close to zero. This means there is no meaningful relationship between annual leave and bonus. Employees who take more leave do not receive higher or lower bonuses in any consistent pattern.

The scatter plot confirms this: the points are widely spread out with no visible trend upward or downward. This indicates that bonuses are likely based on performance, sales achievements, or commission structures rather than the amount of annual leave taken.

What is the relationship between Country and Revenue?

What we need to find: How much revenue does each country generate?
With the tables:

- Sales.SalesOrderHeader : TotalDue for Revenue, TerritoryID, CurrencyRateID
- Sales.SalesTerritory : TerritoryID, Name, CountryRegionCode
- Person.CountryRegion : Name, CountryRegionCode
- Sales;CurrencyRate : AverageRate, ToCurrencyCode for 'USD'

Assumption:

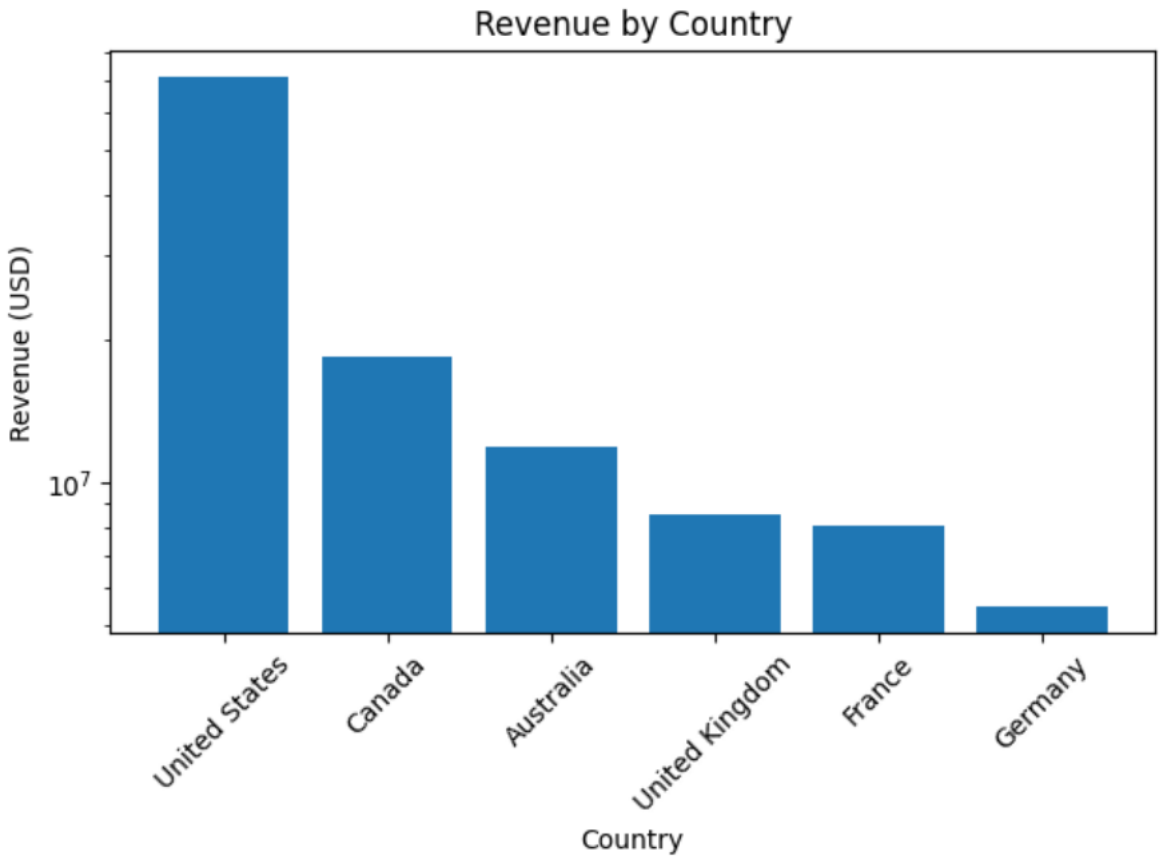
- If CurrencyRateID is NULL, we assume the order is already in **USD** (company base currency).
- If CurrencyRateID is NOT NULL and ToCurrencyCode = 'USD', we multiply TotalDue by AverageRate to convert it to USD.
- If CurrencyRate doesn't convert to USD, we fall back to TotalDue as-is.

SQL:

```
SELECT st.CountryRegionCode , cr.Name as Country, SUM (
CASE
WHEN soh.CurrencyRateID IS NOT NULL
AND cr2.ToCurrencyCode = 'USD'
THEN soh.TotalDue * cr2.AverageRate
ELSE soh.TotalDue
END) as Revenue
FROM Sales.SalesOrderHeader as soh
JOIN Sales.SalesTerritory as st ON st.TerritoryID = soh.TerritoryID
JOIN Person.CountryRegion as cr ON cr.CountryRegionCode = st.CountryRegionCode
LEFT JOIN Sales.CurrencyRate as cr2 ON soh.CurrencyRateID = cr2.CurrencyRateID
GROUP BY st.CountryRegionCode, cr.name
ORDER BY Revenue DESC;
```

Python:

```
CountryRevenue = pd.read_csv('python/Q3RevenueCountry.csv', header = None, names =
['Country','Revenue'])#adding header
print(CountryRevenue )
plt.bar(CountryRevenue['Country'], CountryRevenue['Revenue'])
plt.yscale("log")#to change y axis to log scale
plt.xticks(rotation=45)
plt.xlabel('Country')
plt.ylabel('Revenue')
plt.title('Revenue by Country')
plt.tight_layout ()
plt.show()
```



Conclusion:

We converted all sales amounts into USD using the CurrencyRate table and then aggregated total revenue by country to ensure a fair comparison across different markets. This allowed us to identify which countries generate the highest revenue regardless of local currency.

The results show a strong geographical imbalance: the United States generates by far the highest revenue, followed by Canada and Australia, while European countries contribute significantly less. This means that the company's revenue performance is highly concentrated in a few key markets. Larger or more active sales territories drive out most of the company's income, while other countries play a smaller role in overall revenue. Understanding which countries dominate sales can support better strategic planning, resource allocation, and market development decisions.

What is the relationship between sick leave and Job Title(PersonType)?

What we need to find: We need to find If the Job title/PersonType had an impact on sick leave.

With the tables:

- HumanRessources.Employee: for SickLeaveHours, JobTitle
- Person.Person : BusinessEntityID, PersonType

SQL:

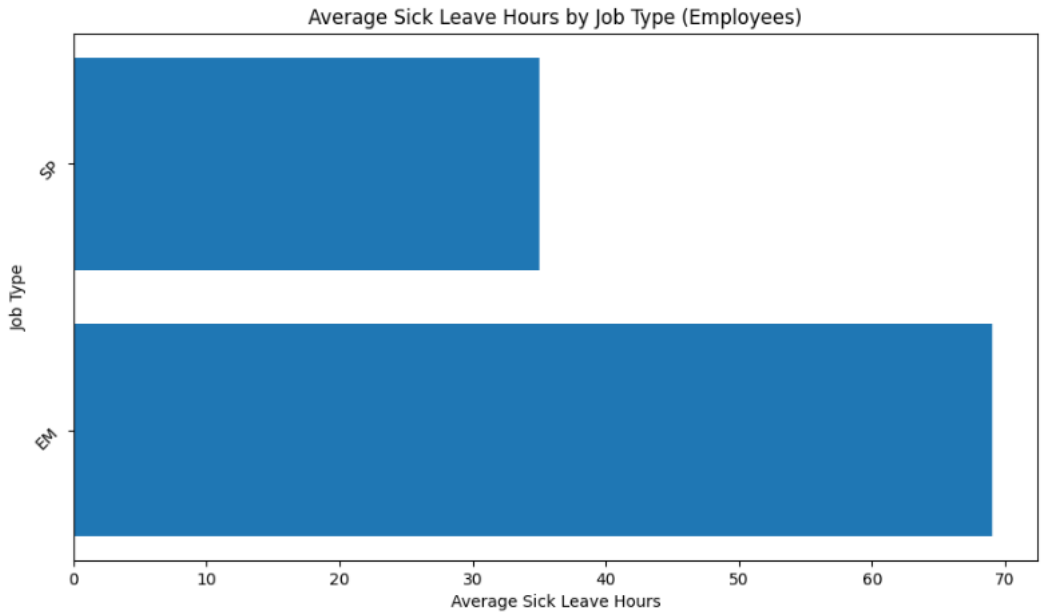
```
SELECT em.JobTitle, pp.PersonType,  AVG(em.SickLeaveHours)
as avgHours, SUM(em.SickLeaveHours) as TotalHours, COUNT (*) as NbEmployee
FROM  HumanResources.Employee as em
JOIN Person.Person as pp on pp.BusinessEntityID = em.BusinessEntityID
WHERE pp.PersonType IN ('EM' ,'SP') -- Only actual employees
GROUP BY em.JobTitle, pp.PersonType
ORDER BY pp.PersonType, avgHours DESC;
```

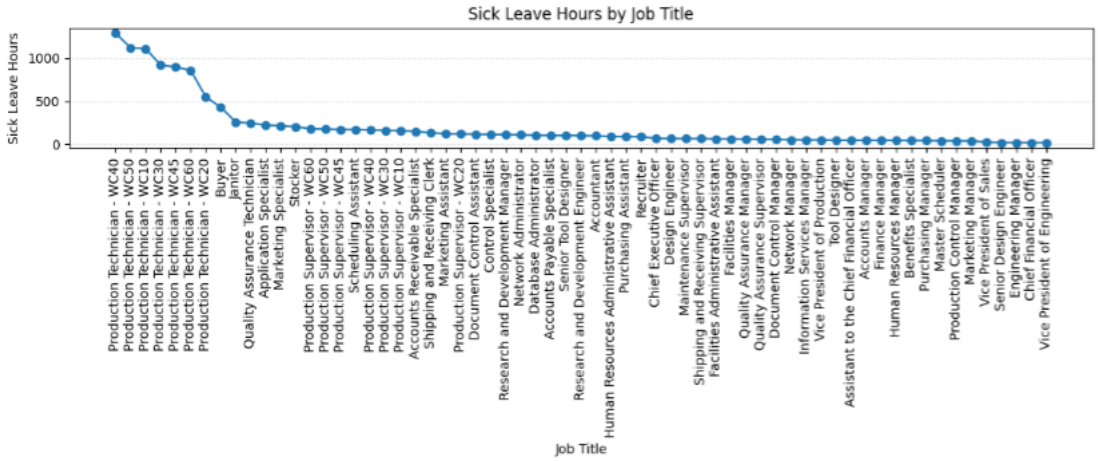
Python:

This one is a full data version, but we can also select only the top 15 by .head(15)

```
import numpy as np   plt.showndas as pd
iplt.showtplotlib.pyplot as plt

df = pd.read_csv(r"C:\Users\amira\OneDrive\Documents\PYTHON\Group-Project\Q4.csv",
header=None, names=['Job Title', 'Sick Leave'])
print(df.columns)
print(df.head())
x = df['Job Title']
y = df['Sick Leave']
plt.plot(df["Job Title"], df["Sick Leave"], marker='o')
plt.xlabel("Job Title")
plt.ylabel("Sick Leave Hours")
plt.title("Sick Leave Hours by Job Title")
plt.xticks(rotation=90)
plt.grid(axis='y', linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```

```python
#Q4

SickLeave = pd.read_csv(
    'python/Q4sickleave.csv',
    header=None,
    names=['Job Title', 'Job Type','Avg Sick Leave', 'Total Sick Leave', 'Num Employees'])

# Plot
plt.figure(figsize=(12, 14))
plt.barh(SickLeave['Job Title'], SickLeave['Avg Sick Leave'])
plt.xlabel('Average Sick Leave Hours')
plt.ylabel('Job Title')
plt.title('Average Sick Leave Hours by Job Title (Employees)')
# Rotate y-axis labels for better readability
plt.yticks(rotation=45)
plt.tight_layout()
plt.savefig('python/Q4sickleave.png')
plt.show()

# Plot
plt.figure(figsize=(12, 14))
plt.barh(SickLeave['Job Type'], SickLeave['Avg Sick Leave'])
plt.xlabel('Average Sick Leave Hours')
plt.ylabel('Job Type')
plt.title('Average Sick Leave Hours by Job Type (Employees)')
# Rotate y-axis labels for better readability
plt.yticks(rotation=45)
plt.tight_layout()
plt.savefig('python/Q4sickleave1.png')
plt.show()
```





Conclusion:

We found that only individuals with PersonType ‘EM’ (Employee) and ‘SP’ (Sales Person) appear in the HumanResources.Employee table, meaning they are the only ones with real job titles and recorded sick-leave hours. All other PersonTypes do not represent internal staff, which is why they were excluded.

After analysing both EM and SP roles, we observed clear differences in average sick-leave hours across job titles. Some positions consistently take more sick leave than others, showing that job role does influence sick-leave behaviour. This suggests that differences may be linked to job demands, workload, or working conditions associated with specific roles.

What is the relationship between store trading duration and revenue?

What we need to find: We need to find If the role store trading duration (in years) has an impact on the store revenue.

With the tables:

- Sales.vStoreWithDemographics : YearOpened, StoreName
- Sales.Customer : link btw store and customers
- Sales.SalesOrderHeader : revenue
- Sales.CurrencyRate: conversion to USD (following Q3)

Code with no currency conversion:

```
SELECT
   vsd.BusinessEntityID AS StoreID,
   vsd.Name AS StoreName,
   vsd.YearOpened,
   YEAR(GETDATE()) - vsd.YearOpened AS TradingDurationYears,
   SUM(soh.TotalDue) AS Revenue
FROM Sales.vStoreWithDemographics AS vsd
JOIN Sales.Customer AS sc ON sc.StoreID = vsd.BusinessEntityID
JOIN Sales.SalesOrderHeader AS soh ON soh.CustomerID = sc.CustomerID
GROUP BY vsd.BusinessEntityID, vsd.Name, vsd.YearOpened;
```

- This matches the same assumption as Q3 for currency conversion:

```
SELECT
   vsd.BusinessEntityID AS StoreID,
   vsd.Name AS StoreName,
   vsd.YearOpened,
   YEAR(GETDATE()) - vsd.YearOpened AS TradingDurationYears,
   SUM(
      CASE
         WHEN soh.CurrencyRateID IS NULL THEN soh.TotalDue
         WHEN cr2.ToCurrencyCode = 'USD' THEN soh.TotalDue * cr2.AverageRate
         ELSE soh.TotalDue
      END
   ) AS RevenueUSD
FROM Sales.vStoreWithDemographics AS vsd
JOIN Sales.Customer AS sc ON sc.StoreID = vsd.BusinessEntityID
JOIN Sales.SalesOrderHeader AS soh ON soh.CustomerID = sc.CustomerID
LEFT JOIN Sales.CurrencyRate AS cr2 ON soh.CurrencyRateID = cr2.CurrencyRateID
GROUP BY vsd.BusinessEntityID, vsd.Name, vsd.YearOpened
HAVING SUM(
      CASE
         WHEN soh.CurrencyRateID IS NULL THEN soh.TotalDue
         WHEN cr2.ToCurrencyCode = 'USD' THEN soh.TotalDue * cr2.AverageRate
         ELSE soh.TotalDue
      END
   ) > 0
ORDER BY vsd.BusinessEntityID DESC;
```

Python:

```
DurationRevenue = pd.read_csv('python/Q5durationRevenue.csv', header = None, names=
['Store ID','Store Name','Year Opened','Trading Duration Years','Revenue'])#adding header
print(DurationRevenue )
```

plt.scatter(DurationRevenue['Trading Duration Years'], DurationRevenue['Revenue'])
plt.xlabel('Trading Duration Years')
plt.ylabel('Revenue (USD)')
plt.title('Relationship Between Store Trading Duration and Revenue')
Plt.yscale('log')
Plt.tight.layout()
plt.savefig('python/Q5durationRevenue.png')
plt.show()

```
#Q5

DurationRevenue = pd.read_csv('python/Q5durationRevenue.csv', header = None, names= ['Store ID','Store Name','Year Opened','Trading Duration Years','Revenue'])#adding header
print(DurationRevenue )

plt.scatter(DurationRevenue['Trading Duration Years'], DurationRevenue['Revenue'])
plt.xlabel('Trading Duration Years')
plt.ylabel('Revenue (USD)')
plt.title('Relationship Between Store Trading Duration and Revenue')
plt.yscale('log')
plt.tight_layout()
plt.savefig('python/Q5durationRevenue.png')
plt.show()
```

Conclusion:

We present two versions of the query: one without currency conversion and one that converts all revenue into USD using the same assumptions as Q3. The first query shows raw revenue, while the second ensures all stores are compared consistently across currencies. For analysis, we use the converted version (RevenueUSD) because it provides a fair comparison between stores.
We analysed store revenue using consistent USD conversion and compared it with trading duration. The results show no significant correlation, meaning a store's age does not predict its revenue.