Group1 Interim Project Portfolio

1. Introduction

In this Interim Project, we work on AdventureWorks - Microsoft SQL Server sample database

The AdventureWorks database supports standard online transaction processing scenarios for a fictitious bicycle manufacturer (Adventure Works Cycles). Scenarios include Manufacturing, Sales, Purchasing, Product Management, Contact Management, and Human Resources.

We also have used the AdventureWorks Data Dictionary, to check the database.

We use the GitHub website as an open-source library comparison to conclude the project.Because GitHub records all of our contributions, helps us to track team's progress together and save all versions of our work in a safe place. Also, we can keep the code and steps public to contribute to the technology community.

2. Project Questions

To start this part there's some important steps useful to consider about the list of ideas generated, consisting of conceptual sketches, function, and key components.

One each participant worked in one or 2 questions first and used the same steps to get the result which include

1.Download the database and connect it to Microsoft SQL Server
2. Observe the database and table
3. Create a Query database on the question's requirement
4. Run the Query (Execute)
5. Export the result
After we make the query on SQL server, we can export that data to a CSV file, from that CSV file, we can use python to make visualization (using pandas and matplotlib that we learned so far) and save that visualization as png file

To export the query result to a csv file, follow these steps:

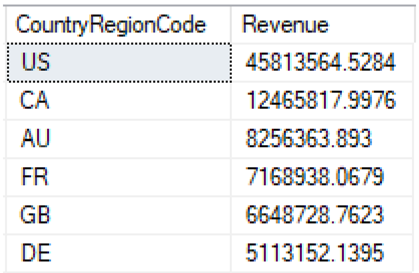
1. Create an empty .csv file on your pc (open your Excel, save as .csv)
2. In SQL Server Management Studio, after you have run a query, go to the Results tab.
3. Right-click the result set and click Select All:
4. Right-click the result set again and click Copy with Headers
5. Paste to the empty .csv file and save.
6. Open Visual Studio Code, create chart with that data result by python (using pandas and matplotlib)
7. Create the document, answer the question and explain how you get that answer.
   We stored the steppes : Group1_Portfolio_Interim_Project_Answers

TASK 1 - What is the regional sales in the best performing country?

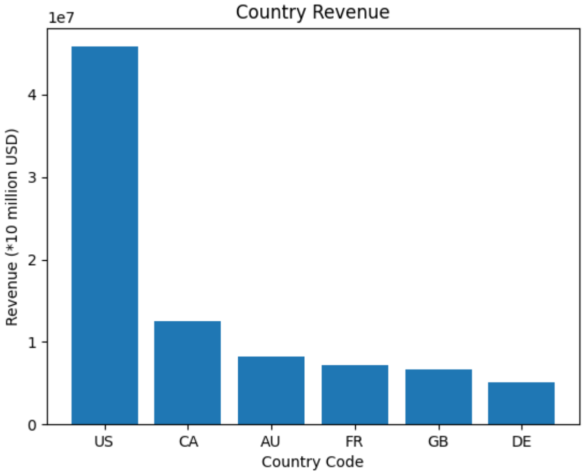
1. To find the best performing country, I need to find the country which has the highest sales by using the table Sales.SalesTerritory. I aggregated revenue by sum 2 values SalesYTD and SalesLastYear.

```
SELECT s.CountryRegionCode, SUM(s.SalesYTD + s.SalesLastYear) as Revenue
FROM Sales.SalesTerritory AS s
GROUP BY s.CountryRegionCode
ORDER BY Revenue DESC;
```

Here is the result:



CountryRegionCode	Revenue
US	45813564.5284
CA	12465817.9976
AU	8256363.893
FR	7168938.0679
GB	6648728.7623
DE	5113152.1395

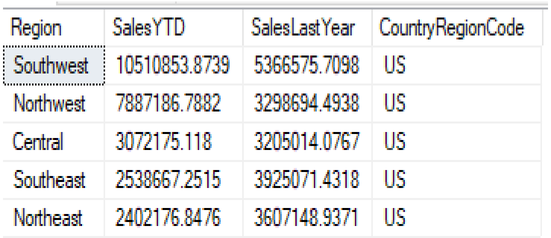


From this result, the best performing country is US

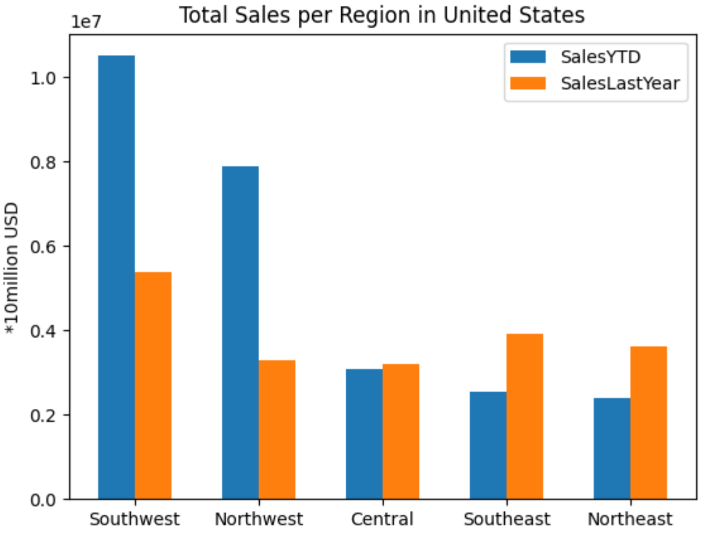
2. I use the query above as a CTE to join with the **SalesTeritory** table using Country_code to find out what are the regional sale in the US:

```
WITH sales_per_country AS (
SELECT TOP 1 s.CountryRegionCode, SUM(s.SalesYTD + s.SalesLastYear) as Revenue
FROM Sales.SalesTerritory AS s
GROUP BY s.CountryRegionCode
ORDER BY Revenue DESC)
SELECT t.Name as Region,
        t.SalesYTD,
        t.SalesLastYear,
        s.CountryRegionCode
FROM Sales.SalesTerritory AS t
INNER JOIN sales_per_country AS s
        ON t.CountryRegionCode = s.CountryRegionCode
ORDER BY t.SalesYTD DESC
```

3. And here is the result of the query, this sales in the best performing country?”



Region	SalesYTD	SalesLastYear	CountryRegionCode
Southwest	10510853.8739	5366575.7098	US
Northwest	7887186.7882	3298694.4938	US
Central	3072175.118	3205014.0767	US
Southeast	2538667.2515	3925071.4318	US
Northeast	2402176.8476	3607148.9371	US



TASK 2 - What is the relationship between annual leave taken and bonus

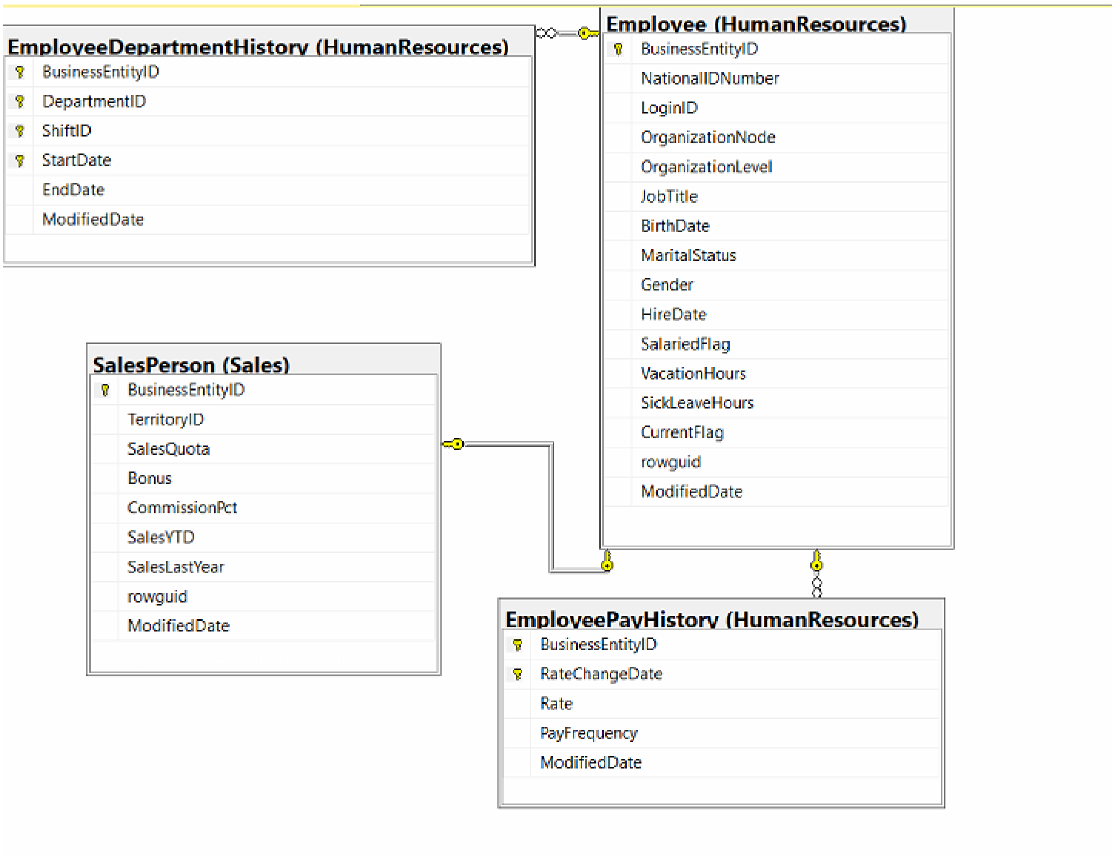
Step by Step

1.Import the database Adventure 2019 , to get access to the dataset.

2.Use the Microsoft SQL SERVER explore data . So we can start exploring the Object and to observe the metadata, tables, key words, columns, rows values ect, we can use the function. We can also check the data on AdventureWorks_Dictionary.

3.So, we started observing tables such as HumanResources.Employee, HumanResources.EmployeeDepartmentHistory,HumanResources.EmployeeDepartmentHistory, HumanResources.EmployeePayHistory. And also checking (using the AdventureWorks_Dictionary) , to find in which table we could find Bonus and VacationHours variables. And after did a Diagram to check the key value for both variables. We placed in a SQL query called - Question2Bonus.

***fig. Diagram Question 2**

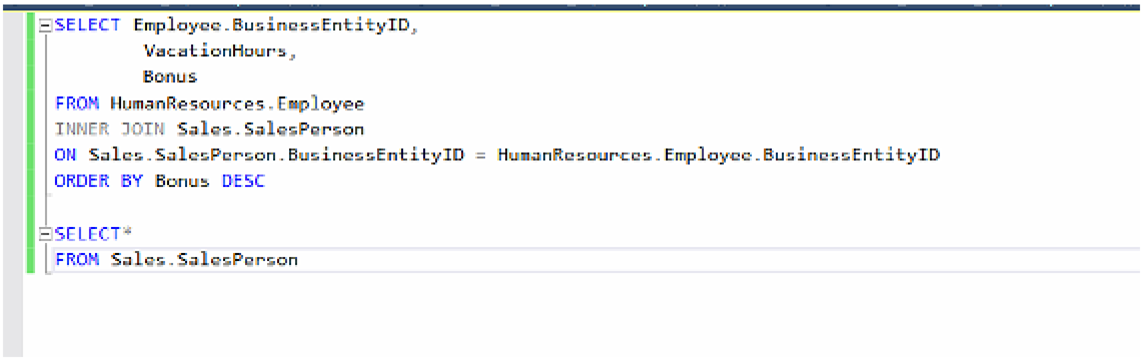


4. Execute the code. We've chosen the function INNER JOIN to join the columns from Sales.SAlesPerson, when we've got the column values for Bonus variable, and HumanResources.Employee, when we've got the VacationHours. The Key value for both is (BusinessEntityID).

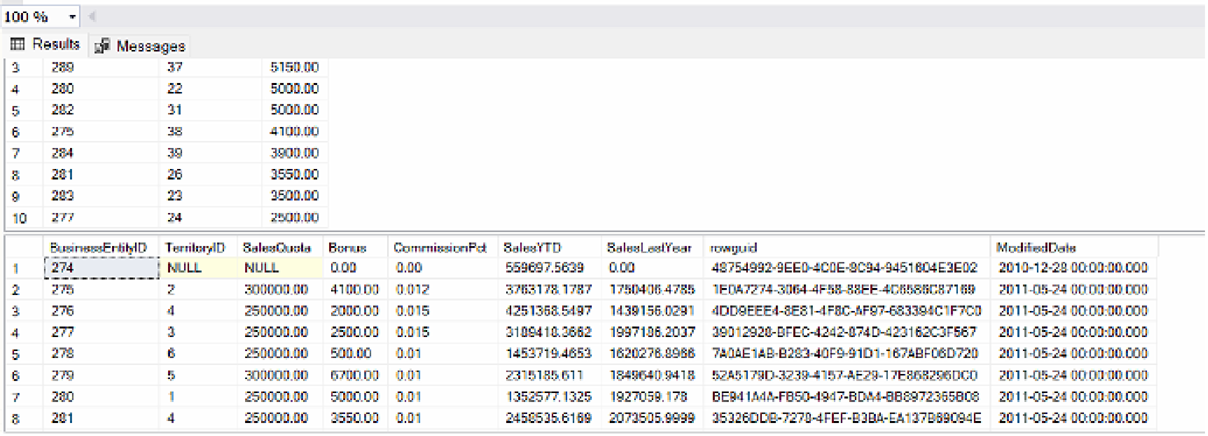
```
SELECT Employee.BusinessEntityID,
           VacationHours,
           Bonus
FROM HumanResources.Employee
INNER JOIN Sales.SalesPerson
ON Sales.SalesPerson.BusinessEntityID = HumanResources.Employee.BusinessEntityID
ORDER BY Bonus DESC
```

**We executed the Sales.Sale.Person table to confirm it is just 17 values.

fig. Question 2- Query and Results

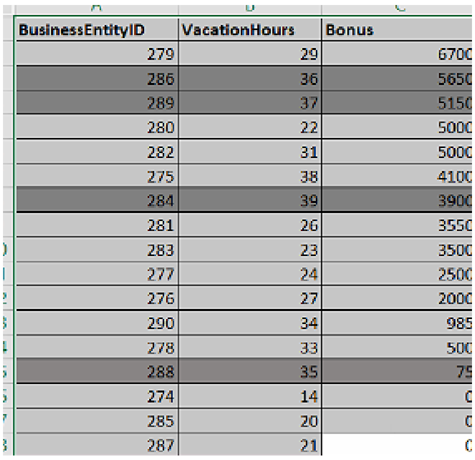


```
SELECT Employee.BusinessEntityID,
        VacationHours,
        Bonus
FROM HumanResources.Employee
INNER JOIN Sales.SalesPerson
ON Sales.SalesPerson.BusinessEntityID = HumanResources.Employee.BusinessEntityID
ORDER BY Bonus DESC

SELECT*
FROM Sales.SalesPerson
```



3	289	37	5150.00
4	280	22	5000.00
5	282	31	5000.00
6	275	38	4100.00
7	284	39	3900.00
8	281	26	3550.00
9	283	23	3500.00
10	277	24	2500.00

	BusinessEntityID	TerritoryID	SalesQuota	Bonus	CommissionPct	SalesYTD	SalesLastYear	rowguid	ModifiedDate
1	274	NULL	NULL	0.00	0.00	559697.5639	0.00	48754992-9EE0-4C0E-8C94-9451604E3E02	2010-12-28 00:00:00.000
2	275	2	300000.00	4100.00	0.012	3763178.1787	1750406.4785	1E0A7274-3064-4F58-88EE-4C6586C87169	2011-05-24 00:00:00.000
3	276	4	250000.00	2000.00	0.015	4251368.5497	1439156.0291	4DD9EEE4-8E81-4F8C-AF97-683394C1F7C0	2011-05-24 00:00:00.000
4	277	3	250000.00	2500.00	0.015	3189418.3662	1997186.2037	39012928-BFEC-4242-874D-423162C3F567	2011-05-24 00:00:00.000
5	278	6	250000.00	500.00	0.01	1453719.4653	1620276.8966	7A0AE1AB-B283-40F9-91D1-167ABF06D720	2011-05-24 00:00:00.000
6	279	5	300000.00	6700.00	0.01	2315185.611	1849640.9418	52A5179D-3239-4157-AE29-17E868296DC0	2011-05-24 00:00:00.000
7	280	1	250000.00	5000.00	0.01	1352577.1325	1927059.178	BE941A4A-FB50-4947-BDA4-BB8972365B08	2011-05-24 00:00:00.000
8	281	4	250000.00	3550.00	0.01	2458535.6169	2073505.9999	35326DDB-7278-4FEF-B3BA-EA137B69094E	2011-05-24 00:00:00.000



BusinessEntityID	VacationHours	Bonus
279	29	6700
286	36	5650
289	37	5150
280	22	5000
282	31	5000
275	38	4100
284	39	3900
281	26	3550
283	23	3500
277	24	2500
276	27	2000
290	34	985
278	33	500
288	35	75
274	14	0
285	20	0
287	21	0

5. Download the Results and save as .csv Question2Bonus.csv
6. Open Python and import the methods pandas, numpy and matplotlib to use functions

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

7. Use the function pd.read_csv() to run the code.

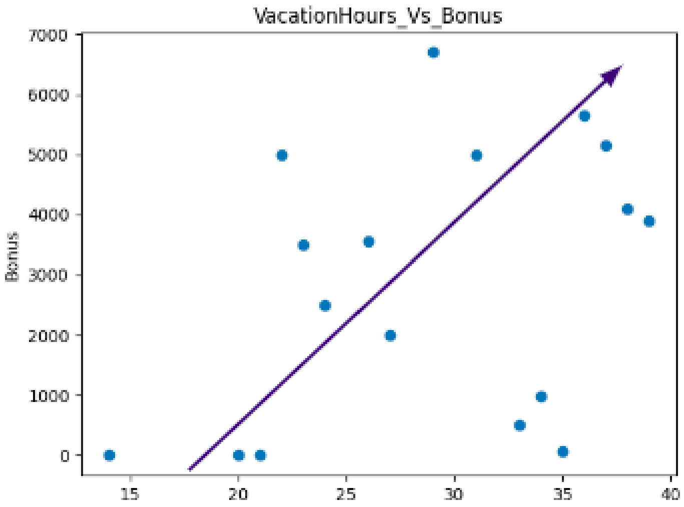
8. Use the functions plt.plot( ) ,plt.scatter( ) ,plt.show( ) to present some chartes. During the process to get answers we started thinking in a hypothetical thoughts

There is a correlation (not strong) between Annual Leave Taken and Bonus, considering as the same time that annual leave taken increases the bonus(money) decreases.

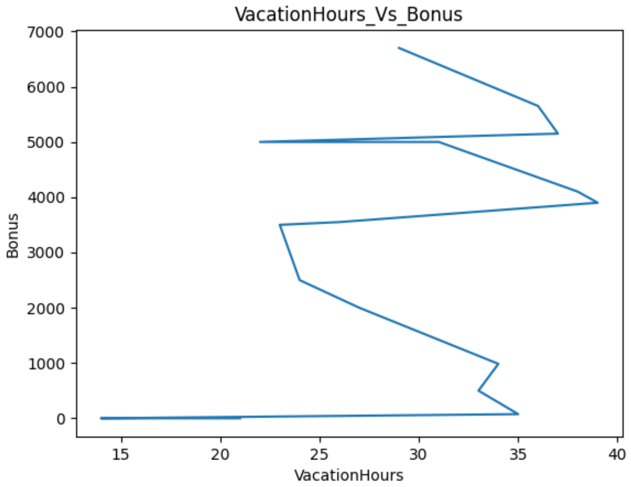
To test it we used the charts : Scatterplot, line plot, because they are all used to compare continuous variables and check the correlation between theirs.

We saved the code on Question2_Python

****Fig3- Scatter Chart VacationHours in Total Hours p/SalePerson vs Bonus(money)p.SalePerson**



There is a positive relation between Annual Leave Taken and Bonus,
But it is not a strength correlation.
The greater the number of hours taken on vacation, the greater the amount earned in bonus.



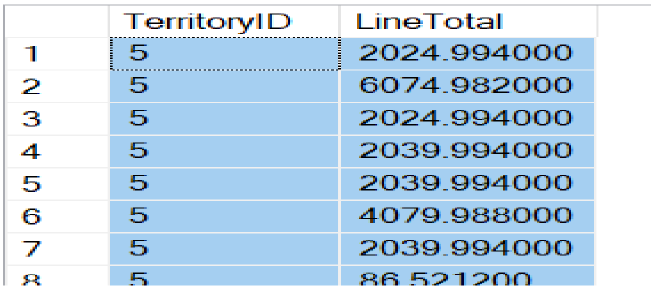
The Scatter chart show the po not close, but going in the same direction
And the line plot , has high points showing this relation positive, but not strong.

TASK 3 - Relationship between Country and Revenue

1. First I identify the tables that might contained the data that was relevant for revenue and countries.
The tables that I found to be relevant were **“SalesOrderHeader”**, **“SalesOrderDetail”** and **“SalesTerritory”**.

2First I extracted the **TerritoryID** and the **LineTotal** from **SalesOrderHeader** and **SalesOrderDetail** by joining the table tables on **SalesOrderID** using the query below:

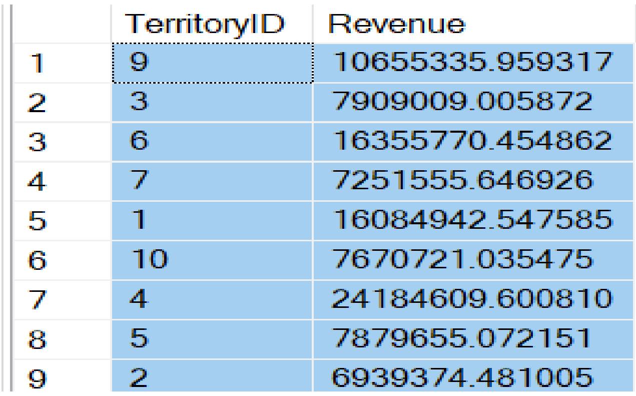
```
SELECT  TerritoryID, LineTotal
FROM Sales.SalesOrderHeader
JOIN Sales.SalesOrderDetail
ON Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID
```



	TerritoryID	LineTotal
1	5	2024.994000
2	5	6074.982000
3	5	2024.994000
4	5	2039.994000
5	5	2039.994000
6	5	4079.988000
7	5	2039.994000
8	5	86.521200

3. Next step I summarized the LineTotal by Territory ID Using this query bellow:

```
SELECT TerritoryID, SUM(LineTotal) AS Revenue
FROM Sales.SalesOrderHeader
JOIN Sales.SalesOrderDetail
ON Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID
GROUP BY TerritoryID
```

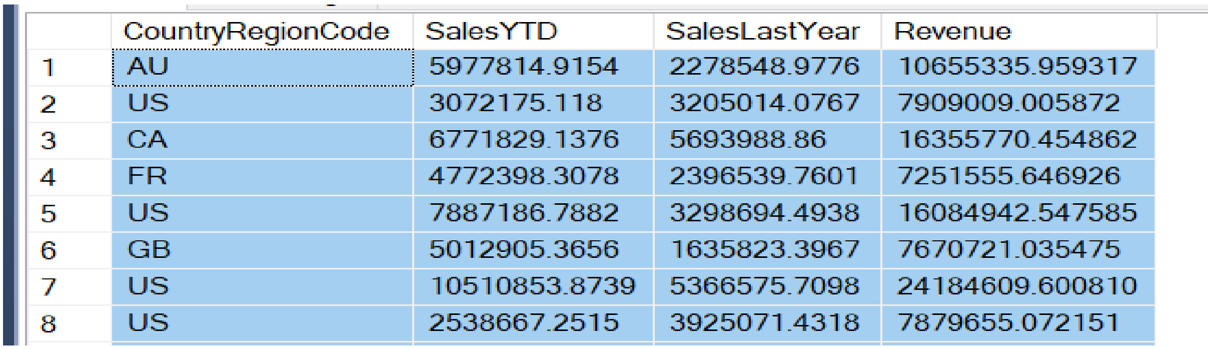


	TerritoryID	Revenue
1	9	10655335.959317
2	3	7909009.005872
3	6	16355770.454862
4	7	7251555.646926
5	1	16084942.547585
6	10	7670721.035475
7	4	24184609.600810
8	5	7879655.072151
9	2	6939374.481005

4. The next thing step is to withdraw the **SaleLastYear, SalesYTD**, and CountryRegionCode from SalesTerritory

5. That was done by joining the result from the previous query to the **SalesTerritor**y on TerrtoryID according query below:

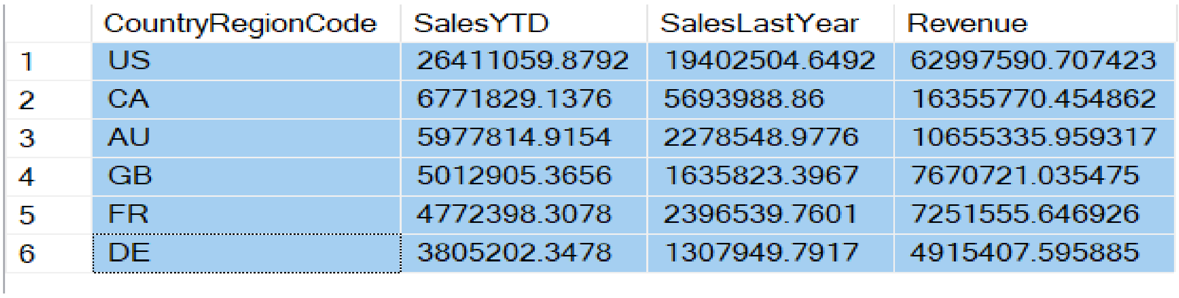
```
SELECT CountryRegionCode, SalesYTD, SalesLastYear, Revenue
FROM Sales.SalesTerritory
JOIN
  (SELECT TerritoryID, SUM(LineTotal) AS Revenue
   FROM Sales.SalesOrderHeader
  JOIN Sales.SalesOrderDetail
  ON Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID
  GROUP BY TerritoryID) AS TerritoryRevenue
ON Sales.SalesTerritory.TerritoryID = TerritoryRevenue.TerritoryID
```



	CountryRegionCode	SalesYTD	SalesLastYear	Revenue
1	AU	5977814.9154	2278548.9776	10655335.959317
2	US	3072175.118	3205014.0767	7909009.005872
3	CA	6771829.1376	5693988.86	16355770.454862
4	FR	4772398.3078	2396539.7601	7251555.646926
5	US	7887186.7882	3298694.4938	16084942.547585
6	GB	5012905.3656	1635823.3967	7670721.035475
7	US	10510853.8739	5366575.7098	24184609.600810
8	US	2538667.2515	3925071.4318	7879655.072151

6. On the following step I grouped the totals by **CoutryRegionCode**

```
SELECT CountryRegionCode, SUM(SalesYTD) AS SalesYTD, SUM(SalesLastYear) AS
SalesLastYear, SUM(Revenue) AS Revenue
FROM Sales.SalesTerritory
JOIN
        (SELECT TerritoryID, SUM(LineTotal) AS Revenue
        FROM Sales.SalesOrderHeader
        JOIN Sales.SalesOrderDetail
        ON Sales.SalesOrderHeader.SalesOrderID = Sales.SalesOrderDetail.SalesOrderID
        GROUP BY TerritoryID) AS TerritoryRevenue
ON Sales.SalesTerritory.TerritoryID = TerritoryRevenue.TerritoryID
GROUP BY CountryRegionCode
ORDER BY Revenue DESC
```

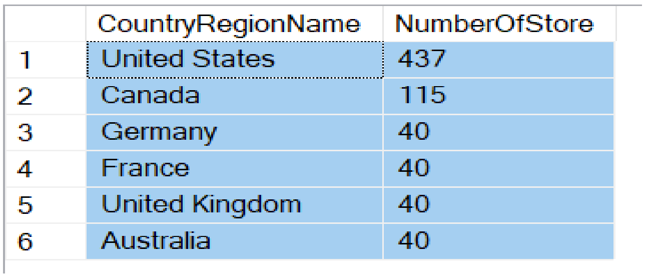


	CountryRegionCode	SalesYTD	SalesLastYear	Revenue
1	US	26411059.8792	19402504.6492	62997590.707423
2	CA	6771829.1376	5693988.86	16355770.454862
3	AU	5977814.9154	2278548.9776	10655335.959317
4	GB	5012905.3656	1635823.3967	7670721.035475
5	FR	4772398.3078	2396539.7601	7251555.646926
6	DE	3805202.3478	1307949.7917	4915407.595885

7. At last I query the table **Sales.vStoreWithAddresses** selecting **CountryRegionName** and counting it by the number of stores to find the relationship between number of stores and revenue..

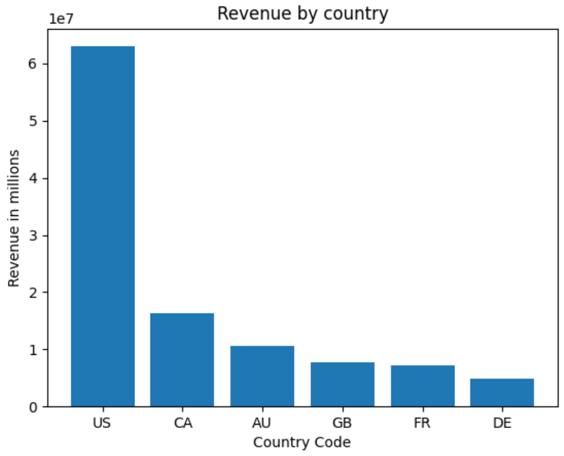
8. We've come to the conclusion that countries with a high number of stores have higher revenue.

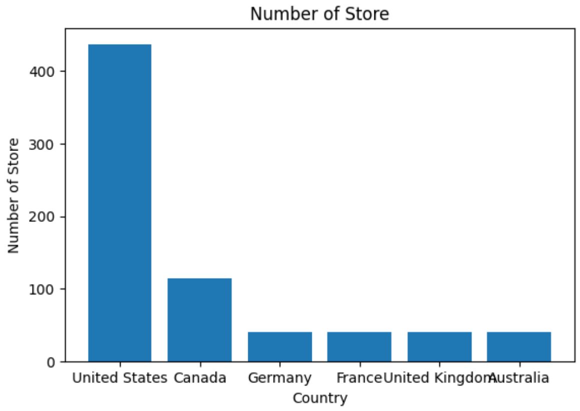
```
SELECT CountryRegionName, COUNT(*)AS NumberOfStore
FROM Sales.vStoreWithAddresses
GROUP BY CountryRegionName
ORDER BY NumberOfStore DESC
```



	CountryRegionName	NumberOfStore
1	United States	437
2	Canada	115
3	Germany	40
4	France	40
5	United Kingdom	40
6	Australia	40

**** Once I have all the queries done I import my csv file into python.So I could generate the charts.**Looking at the charts I come to the conclusion that countries with high number of stores had higher revenue.





TASK 4 - What is the relationship between sick leave and Job Title (PersonType)?

1.To begin, you must find the tables that contain the data you are trying to compare. When looking through the various tables in the database, you can very quickly find that the data you need belong in the columns in the HumanResources.Employees table, and the Person.Person table.

2. To find a relationship between these two tables, you can use the database diagram feature on SSMS (Image 1a). As you can see in Image 1a below, both the HumanResources.Employee and Person.Person tables share the same primary key: BusinessEntityID.

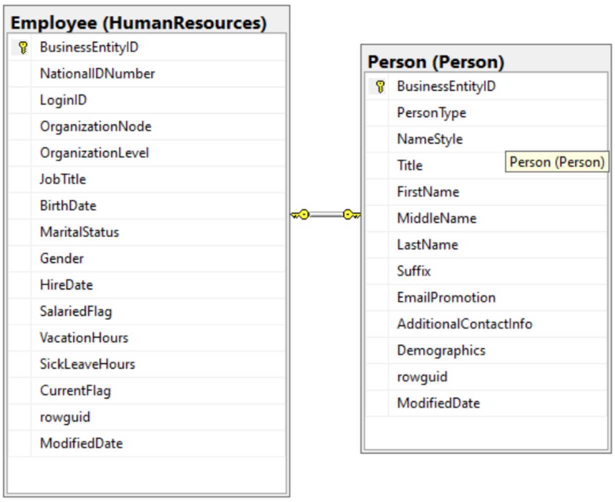
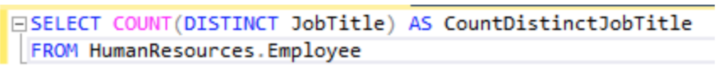


Image 1a

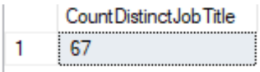
3.Once a relationship can be determined between the two tables, a query can be written to select the columns that hold the data we need.

Before we could begin pulling the data we want, we needed to firstly find out exactly how many unique JobTitles there were in the organization. We did this because if there are many different job titles, then we might not be able to plot them all clearly onto a chart. The query used to find unique JobTitles is shown in the image below (Image 1b).



```
SELECT COUNT(DISTINCT JobTitle) AS CountDistinctJobTitle
FROM HumanResources.Employee
```

Image 1b

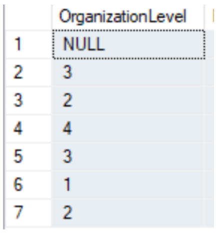


	CountDistinctJobTitle
1	67

Image 1c

4.We can see that there are 67 distinct job titles in the organization (Image 1c), this is far too many to plot on a chart, so we need to find a way to group them into smaller sections. This can be done by selecting the **OrganizationLevel** column in the **HumanResources.Employee table.**

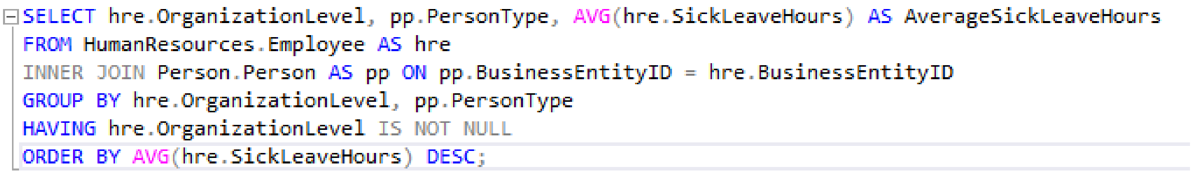
There are four levels in the organization, with the seniority of the job title depending on the number assigned, in this case, the lower the number (1, 2, 3 ,4), the more senior the title. As you can see in the image below (Image 1e), one of the levels in the organization is a null value, this is because the role of CEO in the organization is not assigned a level. We can use the '**HAVING'** clause along with the **LIKE or '='** operator to make sure the query result does not include any values that are null, as this won't help us when trying to see a relationship.



	OrganizationLevel
1	NULL
2	3
3	2
4	4
5	3
6	1
7	2

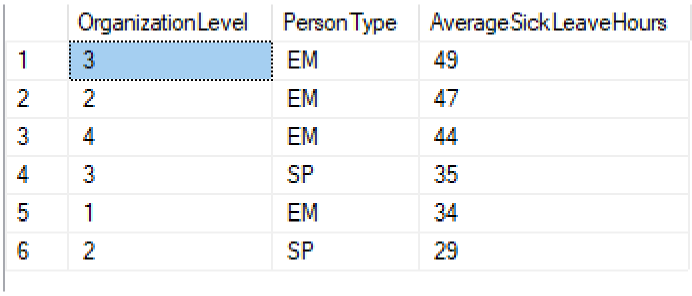
Image 1e

5.The query below (Image 1d) allows us to select the necessary columns, grouping and ordering them, as well as joining the two tables together. The SickLeaveHours are averaged as we are grouping by the organization level and person type.



```
SELECT hre.OrganizationLevel, pp.PersonType, AVG(hre.SickLeaveHours) AS AverageSickLeaveHours
FROM HumanResources.Employee AS hre
INNER JOIN Person.Person AS pp ON pp.BusinessEntityID = hre.BusinessEntityID
GROUP BY hre.OrganizationLevel, pp.PersonType
HAVING hre.OrganizationLevel IS NOT NULL
ORDER BY AVG(hre.SickLeaveHours) DESC;
```

Image 1d

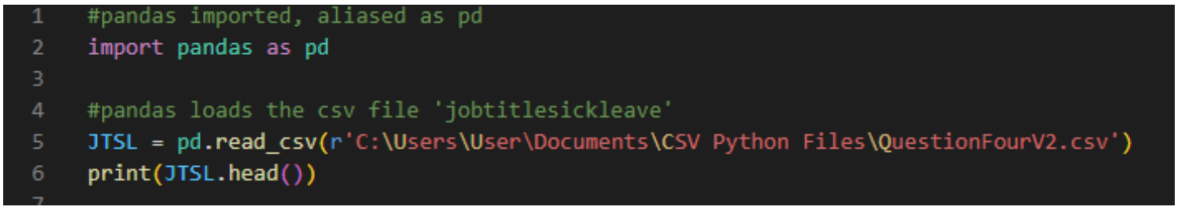


	OrganizationLevel	PersonType	AverageSickLeaveHours
1	3	EM	49
2	2	EM	47
3	4	EM	44
4	3	SP	35
5	1	EM	34
6	2	SP	29

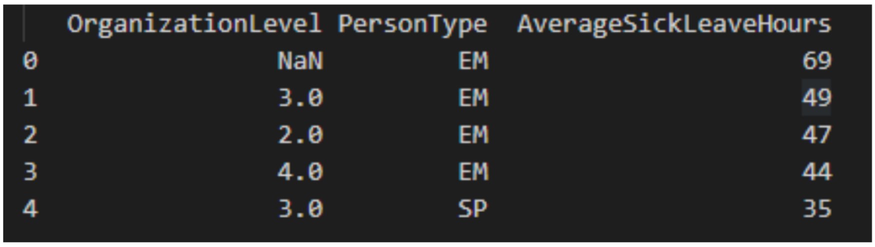
6.Above are the results of the query (Image 1d).

7.The results are saved as a CSV file which can be read by VisualStudioCode by importing the pandas package and using the code 'pandas.read_csv'.

8.I used the function .head() to see the first 5 rows of the dataframe, this allowed me to quickly test if the object had the right type of data in it.



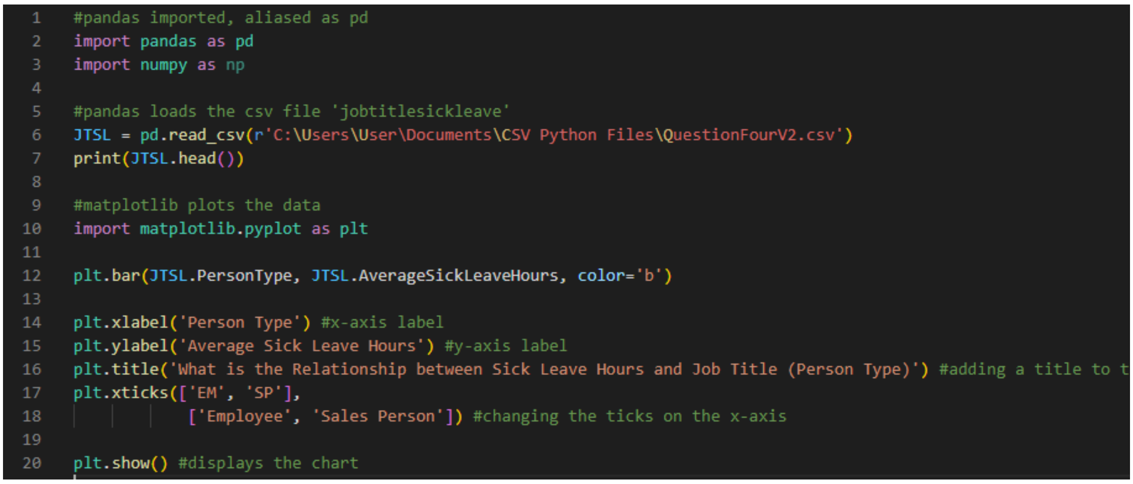
```python
#pandas imported, aliased as pd
import pandas as pd

#pandas loads the csv file 'jobtitlesickleave'
JTSL = pd.read_csv(r'C:\Users\User\Documents\CSV Python Files\QuestionFourV2.csv')
print(JTSL.head())
```



```
  OrganizationLevel PersonType  AverageSickLeaveHours
0               NaN         EM                     69
1               3.0         EM                     49
2               2.0         EM                     47
3               4.0         EM                     44
4               3.0         SP                     35
```

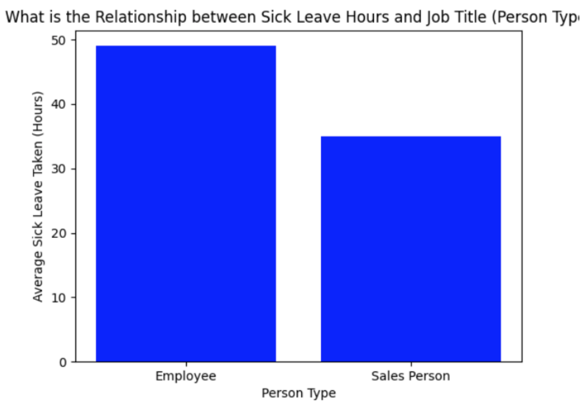
Image 2a

9.Once I was happy that the csv file was working correctly, I began to plot a series of charts to determine which chart showed the relationship between AverageSickLeaveHours and OrganizationLevel the best. The charts I looked at were Bar Charts, Scatter Plots, and Line Plots (Image 2b).



```python
#pandas imported, aliased as pd
import pandas as pd
import numpy as np

#pandas loads the csv file 'jobtitlesickleave'
JTSL = pd.read_csv(r'C:\Users\User\Documents\CSV Python Files\QuestionFourV2.csv')
print(JTSL.head())

#matplotlib plots the data
import matplotlib.pyplot as plt

plt.bar(JTSL.PersonType, JTSL.AverageSickLeaveHours, color='b')

plt.xlabel('Person Type') #x-axis label
plt.ylabel('Average Sick Leave Hours') #y-axis label
plt.title('What is the Relationship between Sick Leave Hours and Job Title (Person Type)') #adding a title to t
plt.xticks(['EM', 'SP'],
            ['Employee', 'Sales Person']) #changing the ticks on the x-axis

plt.show() #displays the chart
```

10. I decided to use the bar chart as I felt it displayed the data in the clearest manner, whilst the scatter plots and line plots did not. I added a title for the chart and labels to the x and y axis.



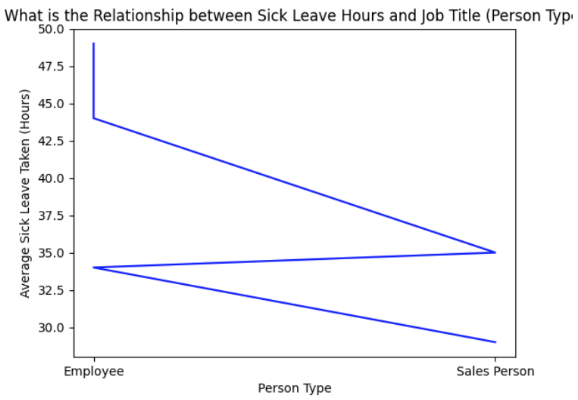
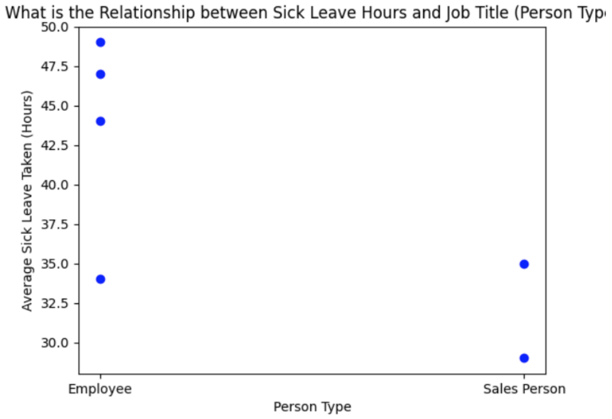
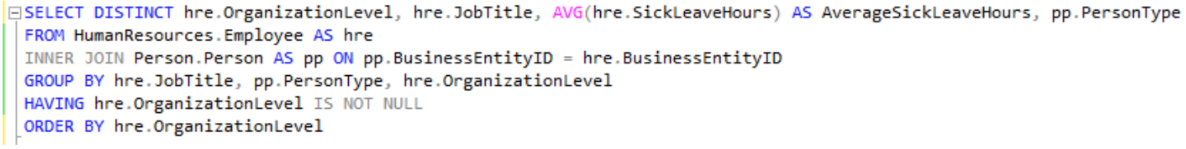


Image 2b



11. Whilst I was happy with the relationship the bar chart was showing, I felt that I could expand on it further by creating another chart showing the average sick leave taken by each level in the organization.

12.I wrote a query on SSMS that showed me how exactly that (Image 2c) and



```
SELECT DISTINCT hre.OrganizationLevel, hre.JobTitle, AVG(hre.SickLeaveHours) AS AverageSickLeaveHours, pp.PersonType
FROM HumanResources.Employee AS hre
INNER JOIN Person.Person AS pp ON pp.BusinessEntityID = hre.BusinessEntityID
GROUP BY hre.JobTitle, pp.PersonType, hre.OrganizationLevel
HAVING hre.OrganizationLevel IS NOT NULL
ORDER BY hre.OrganizationLevel
```

Image 2c

13. I once again made a csv file from the data and imported it onto Visual Studio Code, where I created a bar chart (Image 2d) that showed how the more senior a job title, the less sick leave taken.

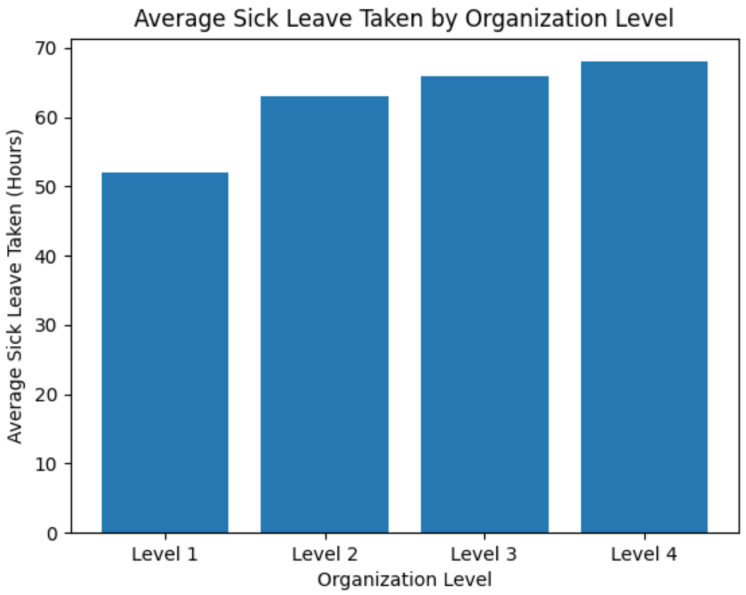
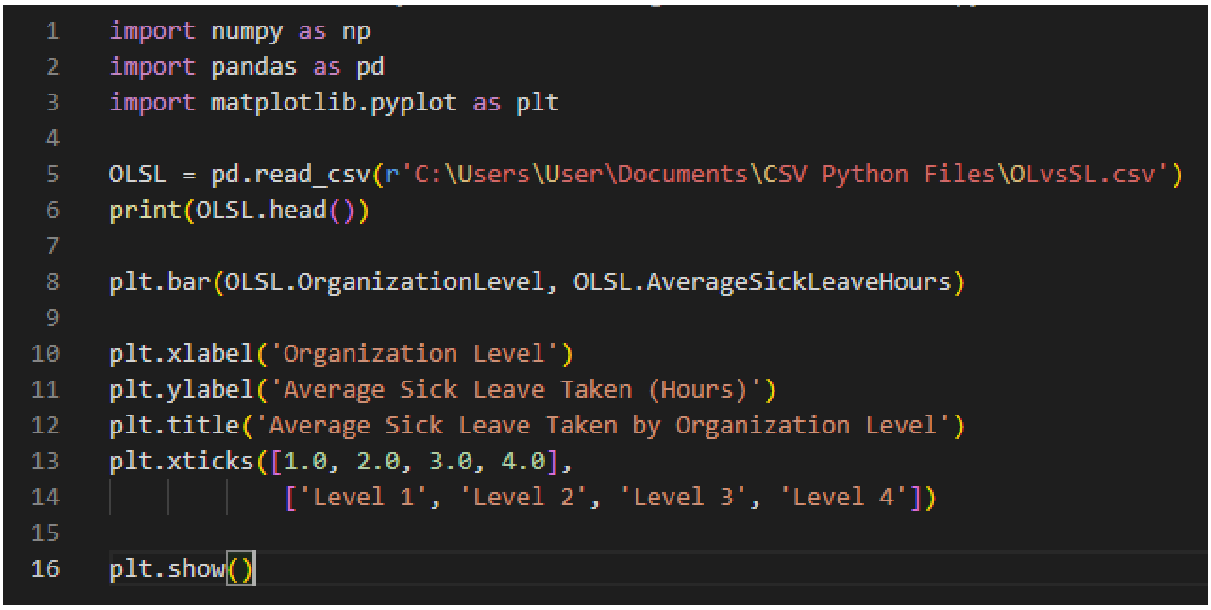


Image 2d



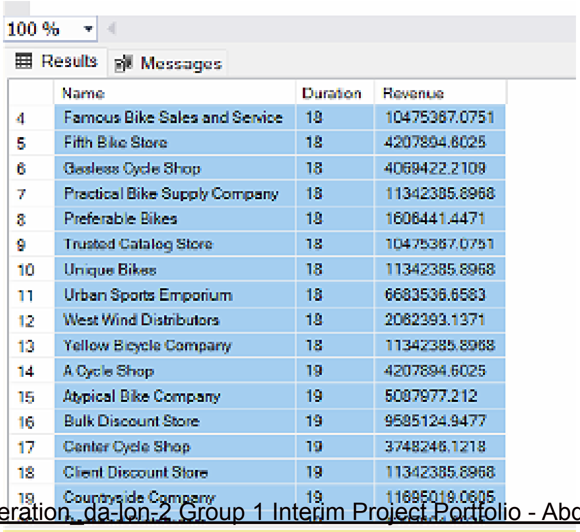
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

OLSL = pd.read_csv(r'C:\Users\User\Documents\CSV Python Files\OLvsSL.csv')
print(OLSL.head())

plt.bar(OLSL.OrganizationLevel, OLSL.AverageSickLeaveHours)

plt.xlabel('Organization Level')
plt.ylabel('Average Sick Leave Taken (Hours)')
plt.title('Average Sick Leave Taken by Organization Level')
plt.xticks([1.0, 2.0, 3.0, 4.0],
            ['Level 1', 'Level 2', 'Level 3', 'Level 4'])

plt.show()
```

TASK 5 - What is the relationship between store trading duration and revenue?

1. Start using the Microsoft SQL SERVER to explore data. It is useful to check the variables on AdventureWorks_Dictionary.
2. After checking the variabel **Trading Store Duration** and **Revenue**, exploring the AdventureWorks_Dictionary. Was decided to discuss which value(Column) value should We use to align **Trading Store Duration** and **Revenue.**
3. So, We decided to use **TotalDue** placed on **SalesOrderHeader** table as the revenue and **YaerOpened** placed on **View: Sales.vStoreWithDemographics** and subtract the values from last year considering in this database.
4. After It, was created an INNER join from Key_ values placed on Tables SalesOrderHeader (Sale.Person.ID), and from the key values placed on View: Sales.vStoreWithDemographics (BusinessEntity.ID)
5. We also selected the StoreName, Grouped the StoreName and YearOpened and Ordered the result by Duration.
6. So, we've got the query SQL QUERY REVENUE BY TotalDue

```
       SELECT st.Name,
       (2019 - de.YearOpened) AS Duration,
       SUM(soh.TotalDue) AS Revenue
FROM Sales.SalesOrderHeader AS soh
INNER JOIN Sales.Store AS st
       ON soh.SalesPersonID =
st.SalesPersonID
INNER JOIN Sales.vStoreWithDemographics AS de
       ON st.BusinessEntityID =
de.BusinessEntityID
WHERE soh.Status = 5
GROUP BY st.Name, de.YearOpened
ORDER BY Duration;
```

7. After executing,download the Results Question5_table_Revenue_by_totalDue.csv.



	Name	Duration	Revenue
4	Famous Bike Sales and Service	18	10475367.0751
5	Fifth Bike Store	18	4207894.6025
6	Gasless Cycle Shop	18	4069422.2109
7	Practical Bike Supply Company	18	11342385.8968
8	Preferable Bikes	18	1606441.4471
9	Trusted Catalog Store	18	10475367.0751
10	Unique Bikes	18	11342385.8968
11	Urban Sports Emporium	18	6683536.6583
12	West Wind Distributors	18	2062393.1371
13	Yellow Bicycle Company	18	11342385.8968
14	A Cycle Shop	19	4207894.6025
15	Atypical Bike Company	19	5087977.212
16	Bulk Discount Store	19	9585124.9477
17	Center Cycle Shop	19	3748246.1218
18	Client Discount Store	19	11342385.8968
19	Countryside Company	19	11695019.0605

8. Open Python and import the methods pandas, numpy and matplotlib to use functions

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

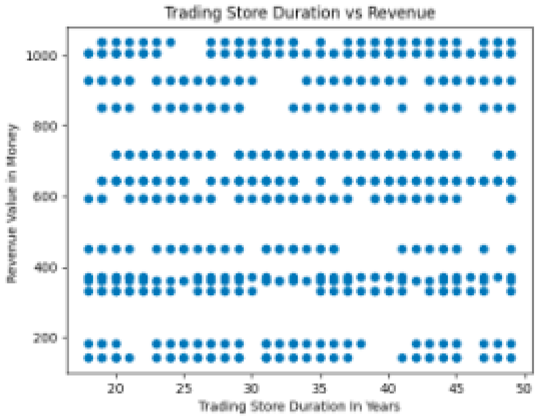
9. Use the function pd.read_csv() to run the code.

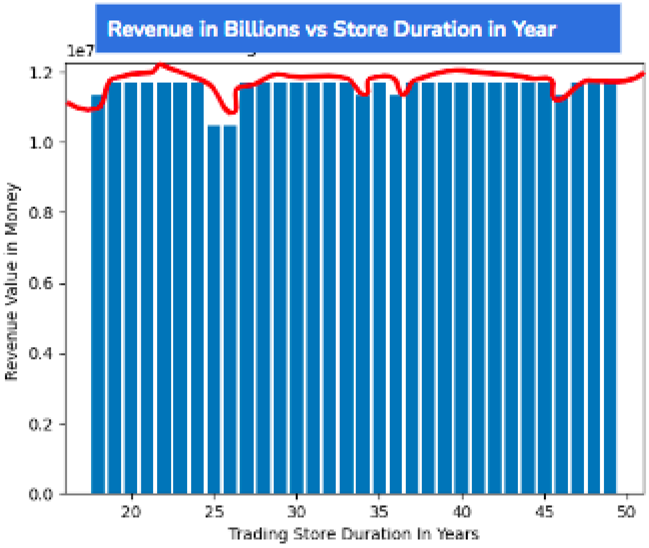
10. So, to check the relation between Trading Store in duration and Revenue, we used the charts : **Scatterplot**, **Line plot** because they are all used to compare continuous variables and check the correlation between theirs. We also did a bar chart and histogram for Revenue values (to check how the distribution was). We used some function to name the axis, add title, and divide the values.
We saved the code on [Question5_Revenue_Total_Due.py]

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt

#td means trading duration vs revenue
td = pd.read_csv("datasets/Question5_table_Revenue_by_totalDue.csv")
print(td.head())

plt.scatter(td.Duration, td.Revenue)
plt.title("Trading Store Duration vs Revenue in Bilions$")
plt.xlabel("Trading Store Duration In Years")
plt.ylabel("Revenue Value in Money ")
plt.show()

#What is the relationship between Trading Store Duration and Revenue ?
#plt.bar(td.Duration, td.Revenue)
#plt.barh (td.Duration, td.Revenue)
#plt.hist (td.Duration)
#plt.hist (td.Duration)
#plt.title("Trading Store Duration vs Revenue")
#plt.xlabel("Trading Store Duration In Years")
#plt.ylabel("Revenue Value in Money ")
#plt.show()
```

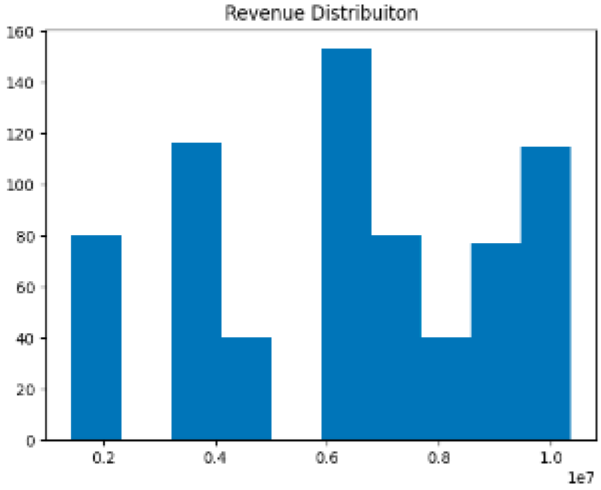
11.After executing the charts below, we have got the conclusion that there's no relation between Trading Store Duration and Revenue. As we can see on the charts below. Because the scatter chart comes with points not relationed, and also, checking the bar chart, we could see a small variation which is considered not relevant.

There's none Relation between Store Trading Duration (in Years) and Revenue. The Revenue from each store name are not relation with your duration time.





We can see the revenue decreased in specific duration stores, but we can not consider it as relevant). There's none variation.



Also, the Revenue value there's none normalized distribution. Which doesn't mean much, but assumes he doesn't follow a standard behavior

TASK 6 - What is the relationship between the size of the stores, number of employees and revenue?

1. I had to find the information for the following; revenue, number of employees and size of stores.

2. I used the TotalDue in SalesOrderHeader table as the revenue.

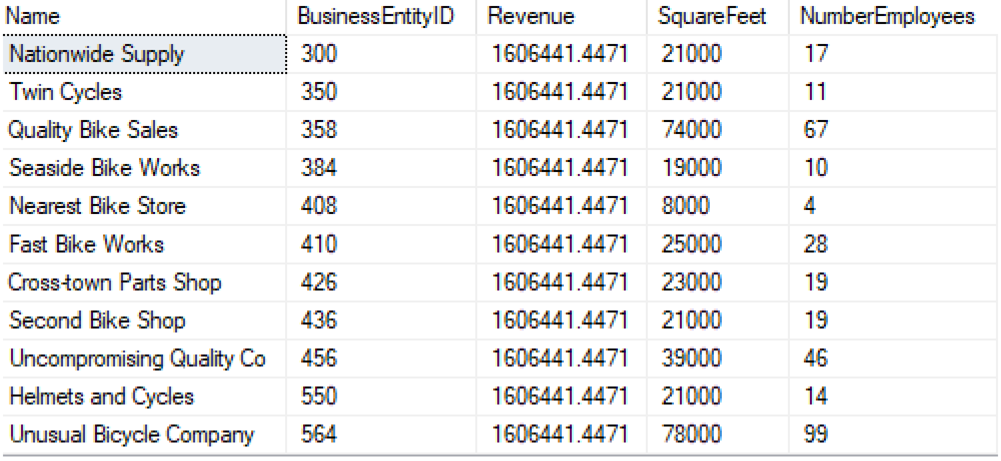
3. To find Store Name, I joined the SalesOrderHeader table with Sales.Store using SalesPerson ID. Which also has the Store's BusinessEntityID in Sales.Store.

4. To find Store size and number of employees, I used the view Sales.vStoreWithDemographics and joined the Sales.Store with the view using BusinessEntityID.

The code:

```
 SELECT
        s.Name,
        s.BusinessEntityID,
        SUM(o.TotalDue) as Revenue,
        d.SquareFeet,
        d.NumberEmployees
FROM Sales.SalesOrderHeader AS o
INNER JOIN Sales.Store AS s
        ON o.SalesPersonID = s.SalesPersonID
INNER JOIN Sales.vStoreWithDemographics AS d
        ON s.BusinessEntityID = d.BusinessEntityID
GROUP BY o.SalesPersonID, s.BusinessEntityID, s.Name, d.SquareFeet, d.NumberEmployees
ORDER BY Revenue;
```

The result of the query:



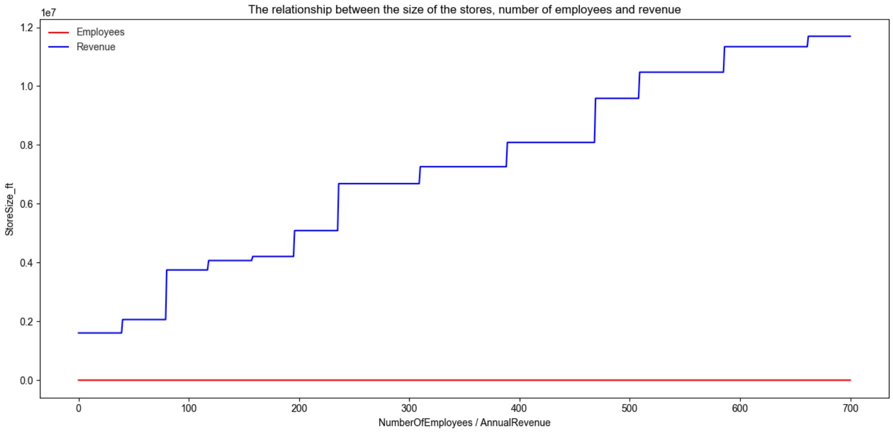
Name	BusinessEntityID	Revenue	SquareFeet	NumberEmployees
Nationwide Supply	300	1606441.4471	21000	17
Twin Cycles	350	1606441.4471	21000	11
Quality Bike Sales	358	1606441.4471	74000	67
Seaside Bike Works	384	1606441.4471	19000	10
Nearest Bike Store	408	1606441.4471	8000	4
Fast Bike Works	410	1606441.4471	25000	28
Cross-town Parts Shop	426	1606441.4471	23000	19
Second Bike Shop	436	1606441.4471	21000	19
Uncompromising Quality Co	456	1606441.4471	39000	46
Helmets and Cycles	550	1606441.4471	21000	14
Unusual Bicycle Company	564	1606441.4471	78000	99

5. I then opened python and imported pandas and matplotlib so i was to use its data plotting and visualization functions.

6.Then I used pd.read_csv() and was able to successfully import the table into python so the data from the previous table could be turned into a graph

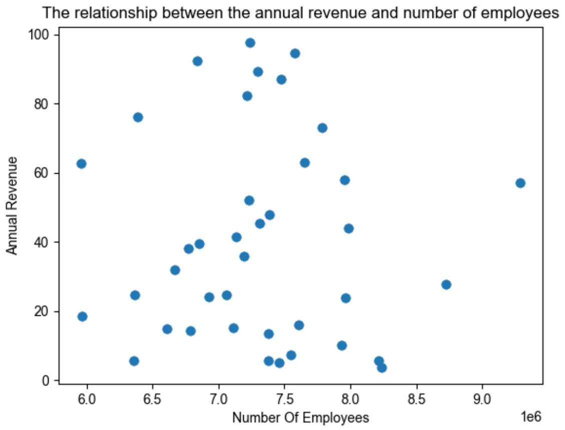
7. Using plt.plot() and plt.scatter() I was able to create a line graph and scatter graph, which i added the axis' label, a title and a key to the graphs.

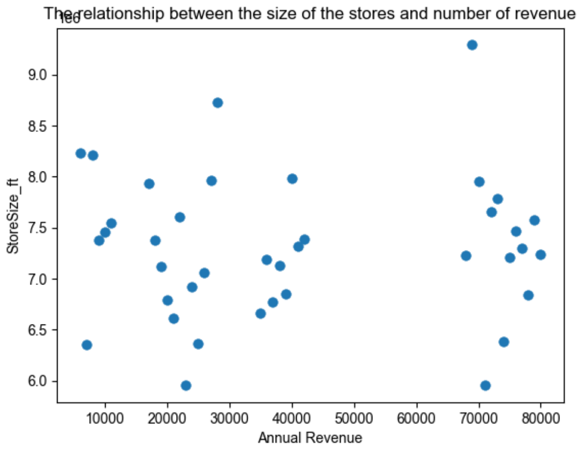
8. Using plt.legend() i was able to describe the elements of the graph and with plt.show() i was able to see whether or not the graph had been created properly.



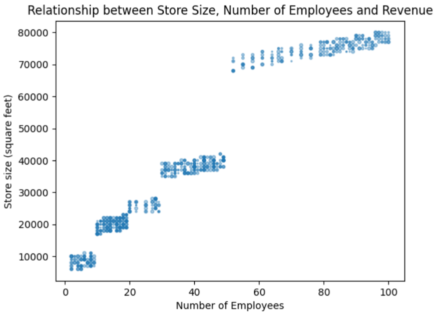
9. I first created a line graph, however I felt the data for the employees was misleading as the y-axis did not accurately represent the data as the integers were too large, so it would be unclear to see whether or not it had a correlation.

10. I then created a scatter graph to see if the categories had any relation. The scatter graph clearly showed the correlation between the two data sets whilst the line graph did not. As I was happy with the result the scatter graph provided, I created 2 more graphs to compare the remaining categories with each other and was able to find whether or not they were also correlated.





11. My findings with the scatter graph showed that there was no correlation between the annual revenue or the number of employees or the size of the stores and the number of revenue.



However there was a relation between the size of the stores and number of employees. Both factors had a strong positive correlation. So I was able to infer that the larger a store is, the number of employees increases.