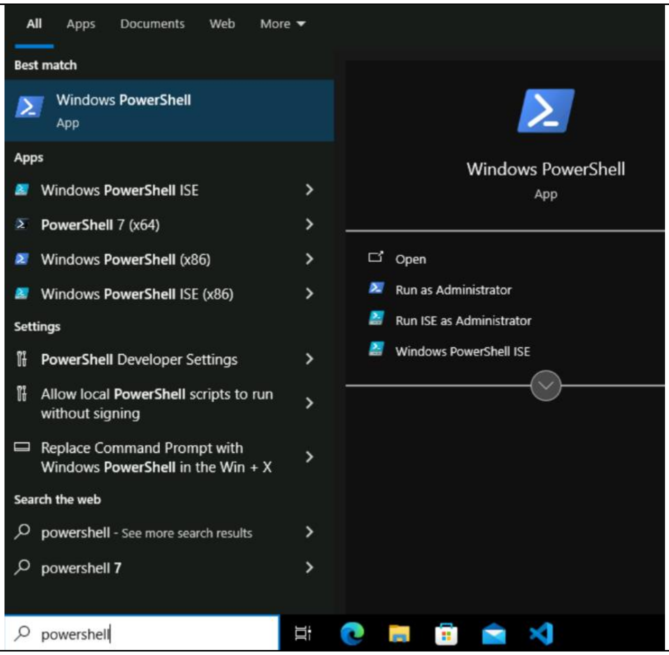
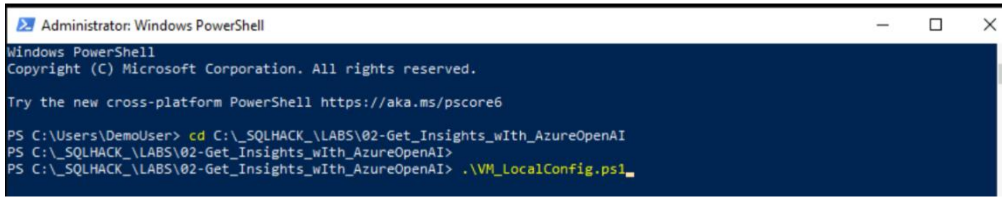
Lab 3 - SQL&GPT: Deploy Application and test SQL Writing & Data Analysis Assistants

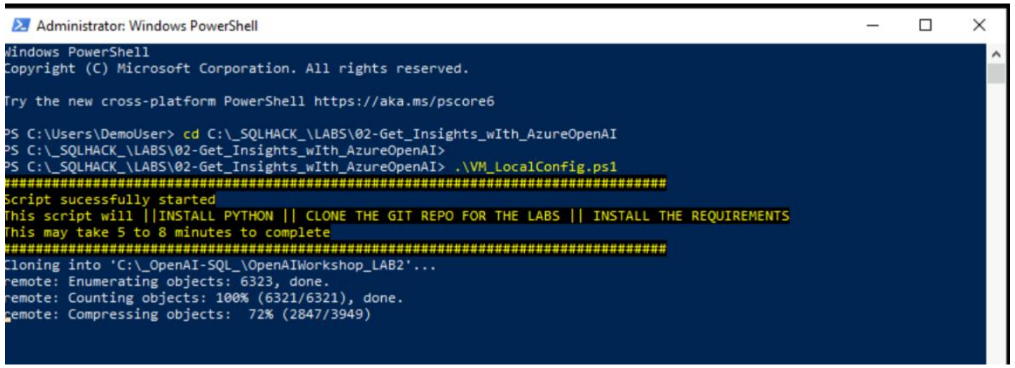
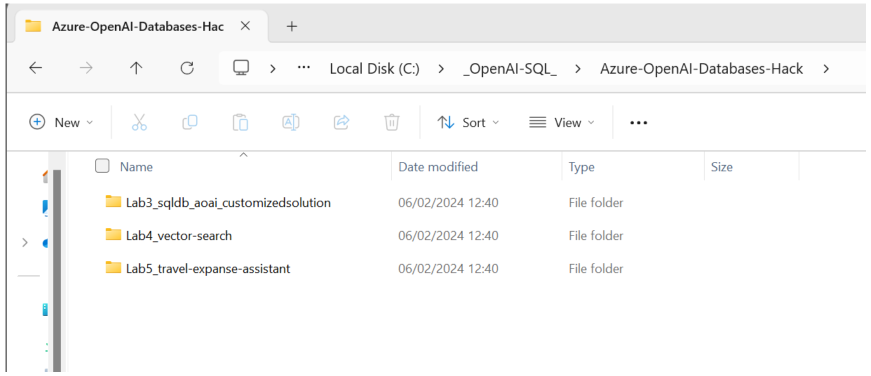
Contents

1. Install the application locally

The required environment has been provisioned upfront for this workshop. This includes Visual Studio Code, GIT, Python, and an Azure OpenAI resource. If you are interested in setting up the environment on your local machine, you can refer to the “environment setup” section in the appendix.

3. **Narrative**	**Screenshot**	**Notes**
Switch back to your Team VM. Once you connect to your Virtual Machine, start **Windows PowerShell** from the Windows start menu, make sure you **run it as Administrator** (right click on Windows PowerShell).	 All Apps Documents Web More Best match Windows PowerShell App Apps Windows PowerShell ISE PowerShell 7 (x64) Windows PowerShell (x86) Windows PowerShell ISE (x86) Settings PowerShell Developer Settings Allow local PowerShell scripts to run without signing Replace Command Prompt with Windows PowerShell in the Win + X Search the web powershell - See more search results powershell 7 Windows PowerShell App Open Run as Administrator Run ISE as Administrator Windows PowerShell ISE powershell	

In PowerShell command line, **navigate to following folder** with the command:
 	
cd C:_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI
	Administrator: Windows PowerShell
	`Windows PowerShell`
	`Copyright (C) Microsoft Corporation. All rights reserved.`
	
	`Try the new cross-platform PowerShell https://aka.ms/pscore6`
	
	`PS C:\Users\DemoUser> cd C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI`
	`PS C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI>`
	`PS C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI> .\VM_LocalConfig.ps1`	

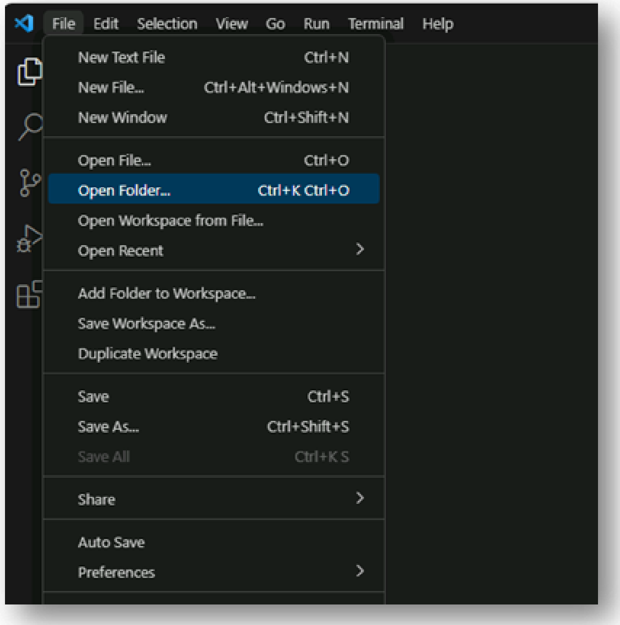
Run the file ***VM_LocalConfig.ps1*** by typing in: ***.\VM_LocalConfig.ps1*** Hit enter and let the script run. This might take some time.		This script will: - Install Python - clone the git Repo for the Labs 2 & 3 - Install the Python Requirements - Set some environment variables
The repository with the 3 labs was cloned for you. Check by navigating to path in the file explorer: ***C:_OpenAI-SQL_\Azure-OpenAI-Databases-Hack*** This is the folder/project you will open in Visual Studio Code.		

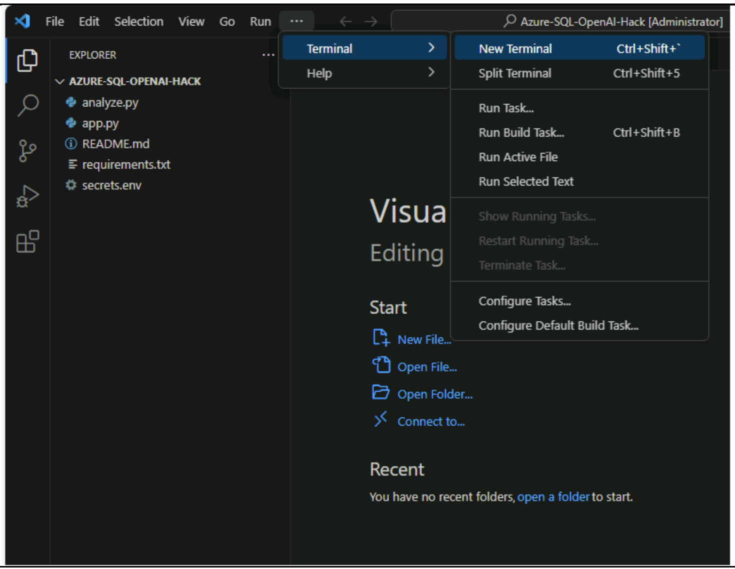
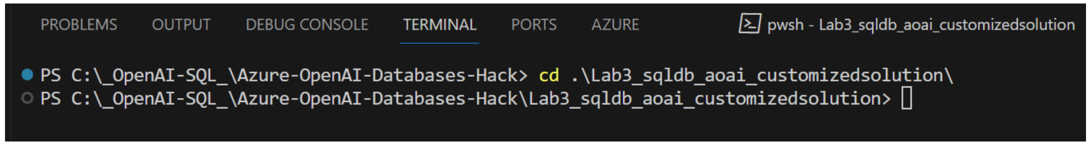
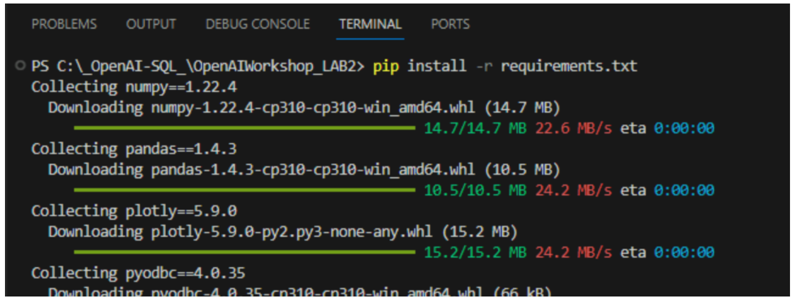
```
Administrator: Windows PowerShell
indows PowerShell
opyright (C) Microsoft Corporation. All rights reserved.

ry the new cross-platform PowerShell https://aka.ms/pscore6

S C:\Users\DemoUser> cd C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI
S C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI>
S C:\_SQLHACK_\LABS\02-Get_Insights_wIth_AzureOpenAI> .\VM_LocalConfig.ps1
################################################################################
cript sucessfully started
his script will ||INSTALL PYTHON || CLONE THE GIT REPO FOR THE LABS || INSTALL THE REQUIREMENTS
his may take 5 to 8 minutes to complete
################################################################################
loning into 'C:\_OpenAI-SQL_\OpenAIWorkshop_LAB2'...
emote: Enumerating objects: 6323, done.
emote: Counting objects: 100% (6321/6321), done.
emote: Compressing objects:  72% (2847/3949)
```

Azure-OpenAI-Databases-Hac
Local Disk (C:) › _OpenAI-SQL_ › Azure-OpenAI-Databases-Hack ›
New
Sort
View

Name	Date modified	Type	Size
Lab3_sqldb_aoai_customizedsolution	06/02/2024 12:40	File folder	
Lab4_vector-search	06/02/2024 12:40	File folder	
Lab5_travel-expanse-assistant	06/02/2024 12:40	File folder	

Microsoft

Open **Visual Studio Code** in the Windows start menu. Import the project/folder via ***File > Open Folder.*** Then select the *folder “**C:\_OpenAI-SQL_\Azure-OpenAI-Databases-Hack**”* When prompted, check the “Trust the authors of all files” option and click **Yes, I trust the authors**.	 File Edit Selection View Go Run Terminal Help New Text File Ctrl+N New File... Ctrl+Alt+Windows+N New Window Ctrl+Shift+N Open File... Ctrl+O Open Folder... Ctrl+K Ctrl+O Open Workspace from File... Open Recent Add Folder to Workspace... Save Workspace As... Duplicate Workspace Save Ctrl+S Save As... Ctrl+Shift+S Save All Ctrl+K S Share Auto Save Preferences	

Open a **terminal** via ***Terminal Tab > New Terminal***		
Navigate to the lab 3 subfolder with the cd command: ***cd .\Lab3_sqldb_aoai_customizedsolution\***		
Check if the needed **requirements** are already installed with the command: ***pip install -r requirements.txt***		

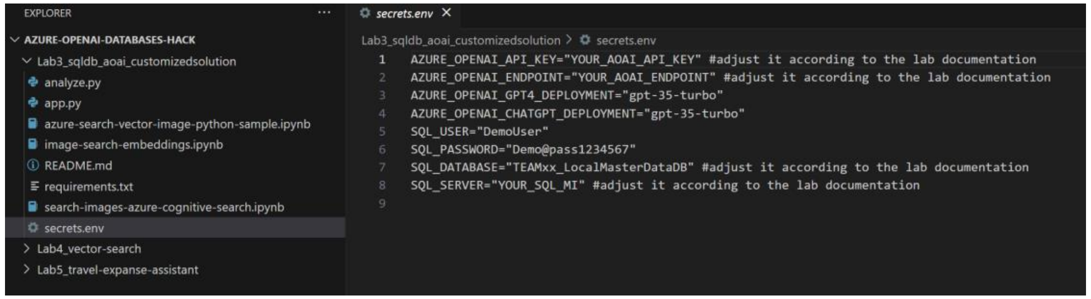
```
● PS C:\_OpenAI-SQL_\Azure-OpenAI-Databases-Hack> cd .\Lab3_sqldb_aoai_customizedsolution\
○ PS C:\_OpenAI-SQL_\Azure-OpenAI-Databases-Hack\Lab3_sqldb_aoai_customizedsolution> 
```

```
○ PS C:\_OpenAI-SQL_\OpenAIWorkshop_LAB2> pip install -r requirements.txt
  Collecting numpy==1.22.4
    Downloading numpy-1.22.4-cp310-cp310-win_amd64.whl (14.7 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.7/14.7 MB 22.6 MB/s eta 0:00:00
  Collecting pandas==1.4.3
    Downloading pandas-1.4.3-cp310-cp310-win_amd64.whl (10.5 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.5/10.5 MB 24.2 MB/s eta 0:00:00
  Collecting plotly==5.9.0
    Downloading plotly-5.9.0-py2.py3-none-any.whl (15.2 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.2/15.2 MB 24.2 MB/s eta 0:00:00
  Collecting pyodbc==4.0.35
    Downloading pyodbc-4.0.35-cp310-cp310-win_amd64.whl (66 kB)
```

Now set the **credential values** within the application code (environment file).

Navigate to the ***secrets.env*** file with the secrets in Lab3 folder.

These should be replaced with the values of the services we use from the Hack subscriptions.


EXPLORER
AZURE-OPENAI-DATABASES-HACK
Lab3_sqldb_aoai_customizedsolution
analyze.py
app.py
azure-search-vector-image-python-sample.ipynb
image-search-embeddings.ipynb
README.md
requirements.txt
search-images-azure-cognitive-search.ipynb
secrets.env
Lab4_vector-search
Lab5_travel-expanse-assistant
secrets.env
Lab3_sqldb_aoai_customizedsolution > secrets.env

```
AZURE_OPENAI_API_KEY="YOUR_AOAI_API_KEY" #adjust it according to the lab documentation
AZURE_OPENAI_ENDPOINT="YOUR_AOAI_ENDPOINT" #adjust it according to the lab documentation
AZURE_OPENAI_GPT4_DEPLOYMENT="gpt-35-turbo"
AZURE_OPENAI_CHATGPT_DEPLOYMENT="gpt-35-turbo"
SQL_USER="DemoUser"
SQL_PASSWORD="Demo@pass1234567"
SQL_DATABASE="TEAMxx_LocalMasterDataDB" #adjust it according to the lab documentation
SQL_SERVER="YOUR_SQL_MI" #adjust it according to the lab documentation

```

Update all your credentials according to the ones on the right side. Make sure you fill in the **highlighted values**.
You can **copy the credentials** on the right side to your env file.

```
AZURE_OPENAI_API_KEY="CHOOSE FROM THE TABLE BELOW"
AZURE_OPENAI_ENDPOINT="CHOOSE FROM THE TABLE BELOW”
AZURE_OPENAI_GPT4_DEPLOYMENT="gpt-4-32k"
AZURE_OPENAI_CHATGPT_DEPLOYMENT="gpt-4-32k"
SQL_USER="DemoUser"
SQL_PASSWORD="Demo@pass1234567"
SQL_DATABASE="TEAMXX_LocalMasterDataDB"
SQL_SERVER=" sqlhackmi-k45dnenbp275u.b8df49bc9122.database.windows.net"
```

For the **Azure OpenAI Endpoint + API Key**, use the values according to your team number.

TEAM	AOAI Endpoint	AOAI API KEY
Team 01 – 05	https://sqloaihack-aue.openai.azure.com/	f93a55528af44339b9b8e0016c1d2c98
Team 06 – 10	https://sqloaihack-cae.openai.azure.com/	8b951e93d34143e895fb848af7322cc8
Team 11 – 15	https://sqloaihack-swd.openai.azure.com/	38302f55245d4d7e87d60f76926aca9a
Team 16 - 20	https://sqloaihack-szn.openai.azure.com/	bb4c8791176840a18a037301c892739e

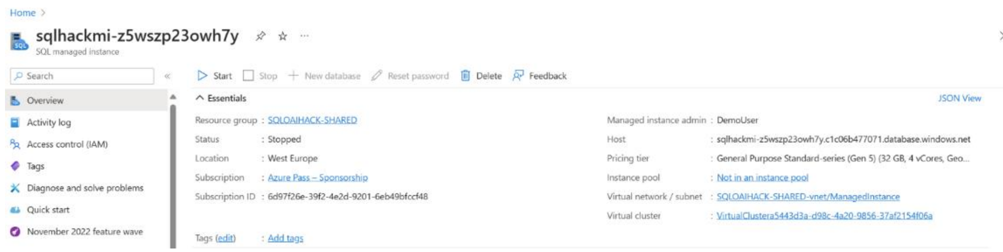
For the **SQL_DATABASE** value, change it to your team's value, e.g. for Team 01, set it to TEAM01_LocalMasterDataDB.

Save the updates (press CTRL+S).

The secrets.env file will prepopulate the values in the application UI (user interfaces).

Later, you can still change the values using the application's User Interface (UI). This will be done in exercise 3 "Test application and its assistants".

Optionally, you can check the SQL_SERVER value again in the Azure Portal. The SQL_SERVER value is the managed instance host name. Example:



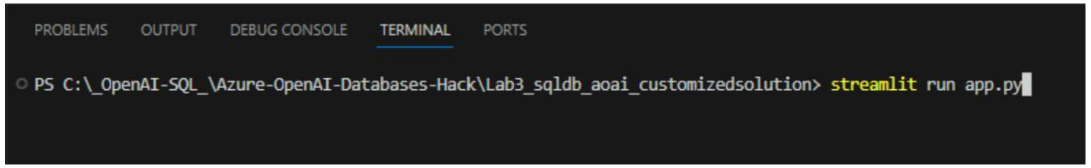
Resource group : SQLOAIHACK-SHARED
Status : Stopped
Location : West Europe
Subscription : Azure Pass – Sponsorship
Subscription ID : 6d97f26e-39f2-4e2d-9201-6eb49bfccf48
Tags (edit) : Add tags

Managed instance admin : DemoUser
Host : sqlhackmi-z5wszp23owh7y.c1c06b477071.database.windows.net
Pricing tier : General Purpose Standard-series (Gen 5) (32 GB, 4 vCores, Geo...
Instance pool : Not in an instance pool
Virtual network / subnet : SQLOAIHACK-SHARED-vnet/ManagedInstance
Virtual cluster : VirtualClustera5443d3a-d98c-4a20-9856-37af2154f06a

In Visual Studio Code, in a terminal, make sure you navigate to the lab 3 folder.

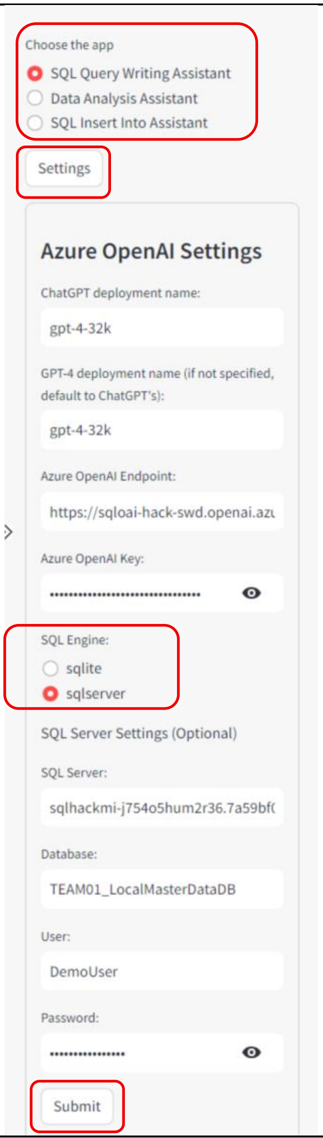
To **run the application** run the command:

streamlit run app.py



```
PS C:\_OpenAI-SQL_\Azure-OpenAI-Databases-Hack\Lab3_sqldb_aoai_customizedsolution> streamlit run app.py
```

2. Test application and its assistants

If prompted to give an **e-mail address**, put your demo user address. E.g. sqlhackuserXX@M365x90478194.onmicrosoft.com An **Edge browser page** opens with settings retrieved from secrets.env. You can also manually change the connection information in the user interface if needed. On the left side, select the assistant app "**SQL Query writing Assistant**". Open **Settings**, and make sure you select "**sqlserver**" as the SQL Engine. Click **Submit** to save the settings. Keep GPT Model as "**ChatGPT**". Select "**Show code**" & "**Show prompt**" to check what happened in the background to return the answer. *Side note:* The option "sqlite" is a built-in server by Python for you to test out.		

Choose the app
[✓ SQL Query Writing Assistant]
Data Analysis Assistant
SQL Insert Into Assistant

Settings

Azure OpenAI Settings

ChatGPT deployment name: gpt-4-32k

GPT-4 deployment name (if not specified, default to ChatGPT's): gpt-4-32k

Azure OpenAI Endpoint: https://sqloai-hack-swd.openai.azu

Azure OpenAI Key: ••••••••••••••••••••••••••••••••

SQL Engine: sqlite [✓ sqlserver]

SQL Server Settings (Optional)

SQL Server: sqlhackmi-j754o5hum2r36.7a59bf(

Database: TEAM01_LocalMasterDataDB

User: DemoUser

Password: ••••••••••••••••

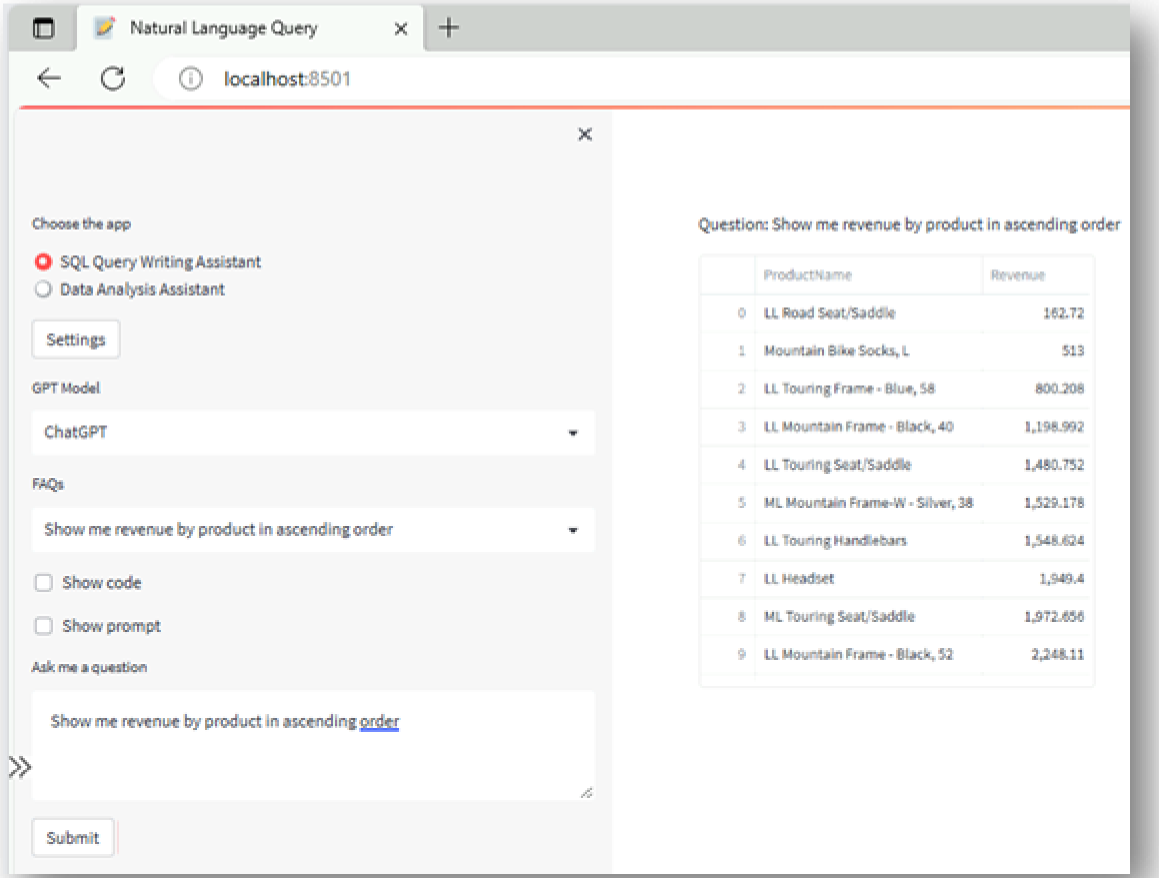
Submit

In this workshop, we will focus on using our own managed instance (option “**sql server**”).

Let’s test the **SQL Query writing Assistant**. Make sure you have selected this assistant.

Under “**Ask me a question**” below, you can try out your first sample queries in natural language.
Past the sample 1 below and hit the **Submit** button after.

<u>Sample 1:</u> Show me revenue by product in ascending order



Choose the app
[✓ SQL Query Writing Assistant]
Data Analysis Assistant

GPT Model: ChatGPT

FAQs: Show me revenue by product in ascending order

Show code
Show prompt

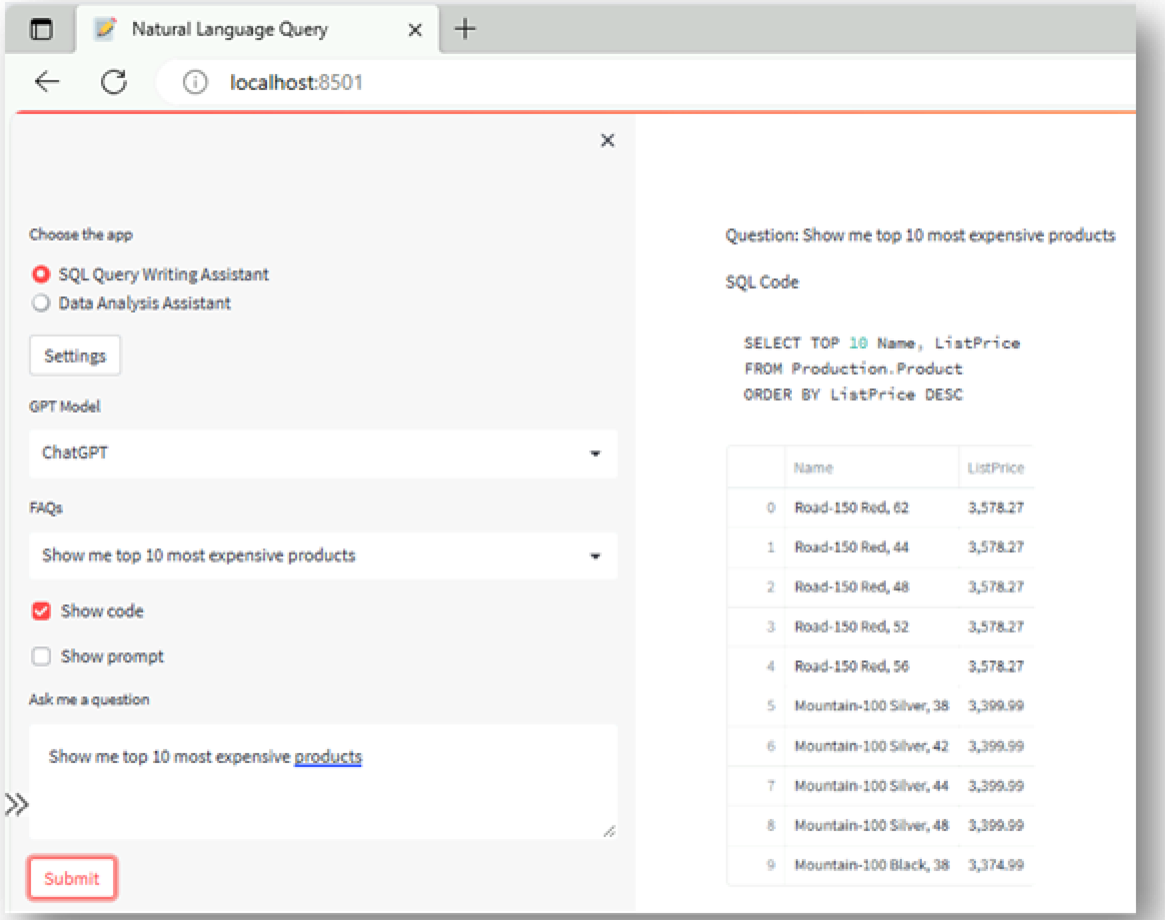
Ask me a question: Show me revenue by product in ascending order

Question: Show me revenue by product in ascending order

	ProductName	Revenue
0	LL Road Seat/Saddle	162.72
1	Mountain Bike Socks, L	513
2	LL Touring Frame - Blue, 58	800.208
3	LL Mountain Frame - Black, 40	1,198.992
4	LL Touring Seat/Saddle	1,480.752
5	ML Mountain Frame-W - Silver, 38	1,529.178
6	LL Touring Handlebars	1,548.624
7	LL Headset	1,949.4
8	ML Touring Seat/Saddle	1,972.656
9	LL Mountain Frame - Black, 52	2,248.11

Microsoft

<u>Sample 2:</u> Show me top 10 most expensive products



Choose the app

- [✓ SQL Query Writing Assistant]
- Data Analysis Assistant

GPT Model: ChatGPT

FAQs: Show me top 10 most expensive products

- [✓] Show code
- [ ] Show prompt

Ask me a question: Show me top 10 most expensive products

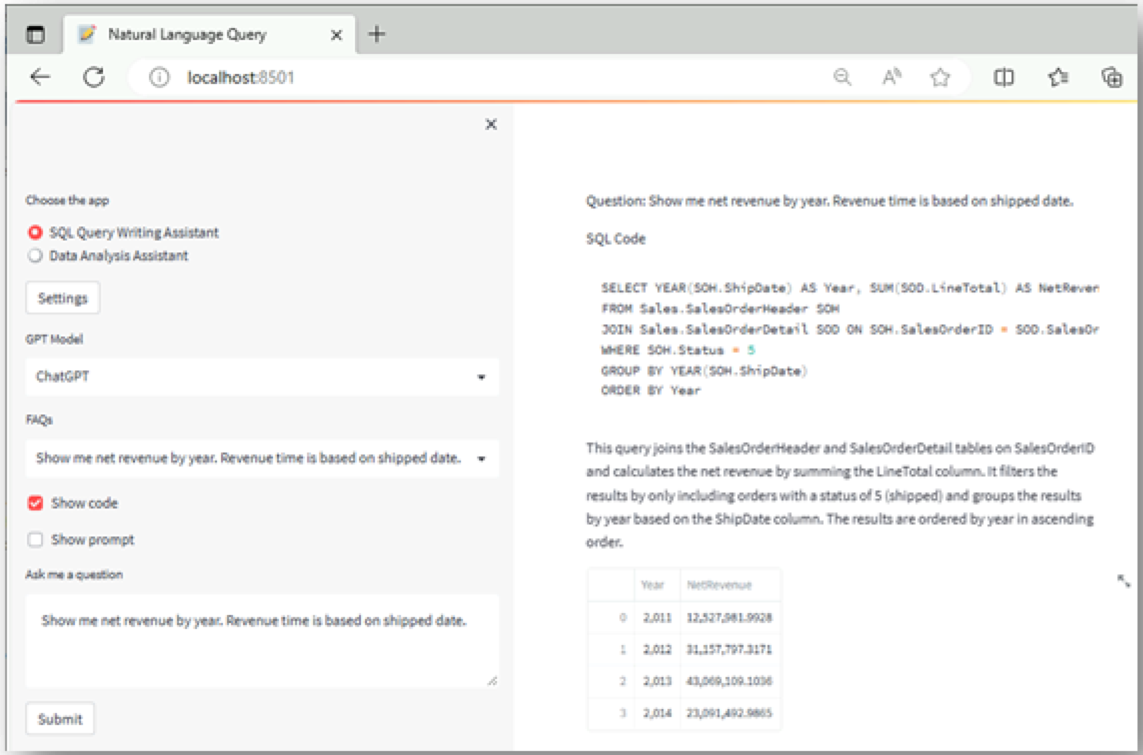
Question: Show me top 10 most expensive products

SQL Code

```
SELECT TOP 10 Name, ListPrice
FROM Production.Product
ORDER BY ListPrice DESC
```

	Name	ListPrice
0	Road-150 Red, 62	3,578.27
1	Road-150 Red, 44	3,578.27
2	Road-150 Red, 48	3,578.27
3	Road-150 Red, 52	3,578.27
4	Road-150 Red, 56	3,578.27
5	Mountain-100 Silver, 38	3,399.99
6	Mountain-100 Silver, 42	3,399.99
7	Mountain-100 Silver, 44	3,399.99
8	Mountain-100 Silver, 48	3,399.99
9	Mountain-100 Black, 38	3,374.99

<u>Sample 3:</u> Show me net revenue by year. Revenue time is based on shipped date.



Choose the app

[✓ SQL Query Writing Assistant]
Data Analysis Assistant

GPT Model: ChatGPT

FAQs: Show me net revenue by year. Revenue time is based on shipped date.

[✓ Show code]
Show prompt

Ask me a question: Show me net revenue by year. Revenue time is based on shipped date.

Question: Show me net revenue by year. Revenue time is based on shipped date.

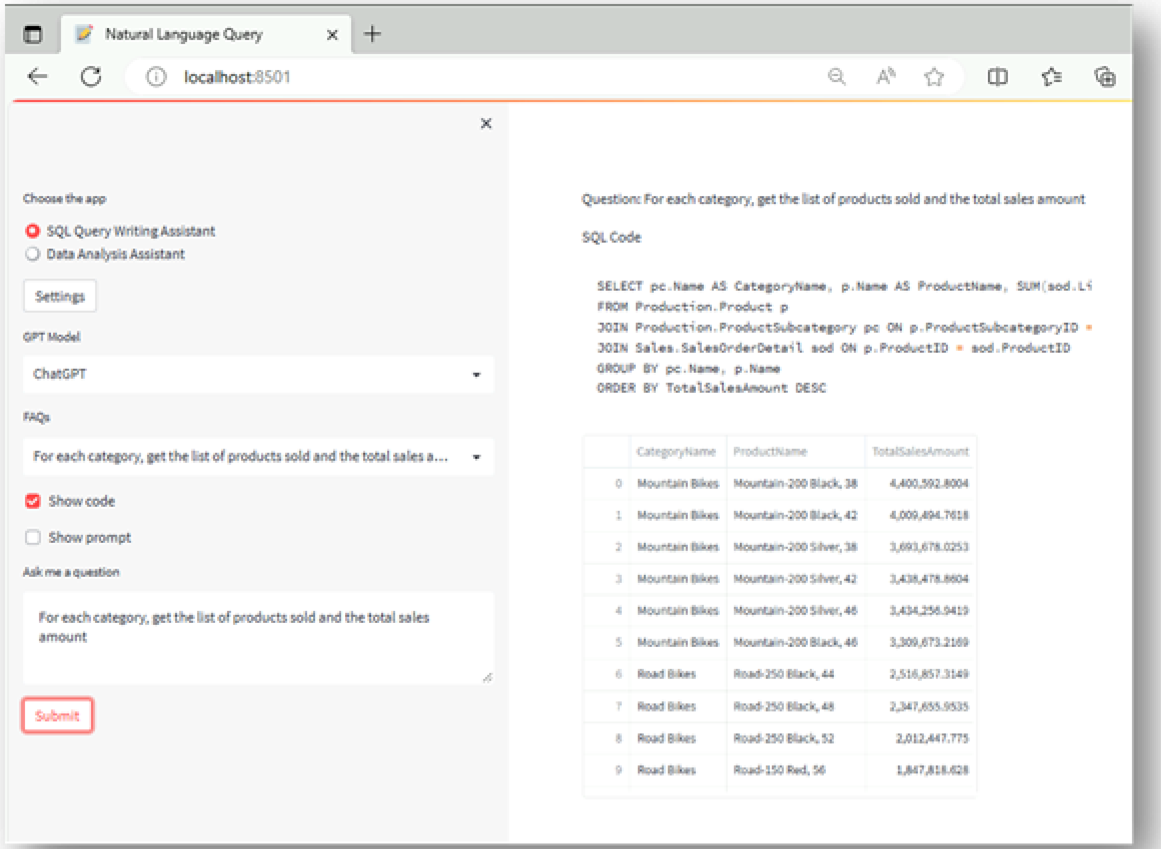
SQL Code

```
SELECT YEAR(SOH.ShipDate) AS Year, SUM(SOD.LineTotal) AS NetRever
FROM Sales.SalesOrderHeader SOH
JOIN Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOr
WHERE SOH.Status = 5
GROUP BY YEAR(SOH.ShipDate)
ORDER BY Year
```

This query joins the SalesOrderHeader and SalesOrderDetail tables on SalesOrderID and calculates the net revenue by summing the LineTotal column. It filters the results by only including orders with a status of 5 (shipped) and groups the results by year based on the ShipDate column. The results are ordered by year in ascending order.

	Year	NetRevenue
0	2,011	12,527,981.9928
1	2,012	31,157,797.3171
2	2,013	43,069,109.1036
3	2,014	23,091,492.9865

<u>Sample 4:</u> For each category, get the list of products sold and the total sales amount



Choose the app

[✓ SQL Query Writing Assistant]
Data Analysis Assistant

GPT Model: ChatGPT

FAQs: For each category, get the list of products sold and the total sales a...

[✓ Show code]
Show prompt

Ask me a question: For each category, get the list of products sold and the total sales amount

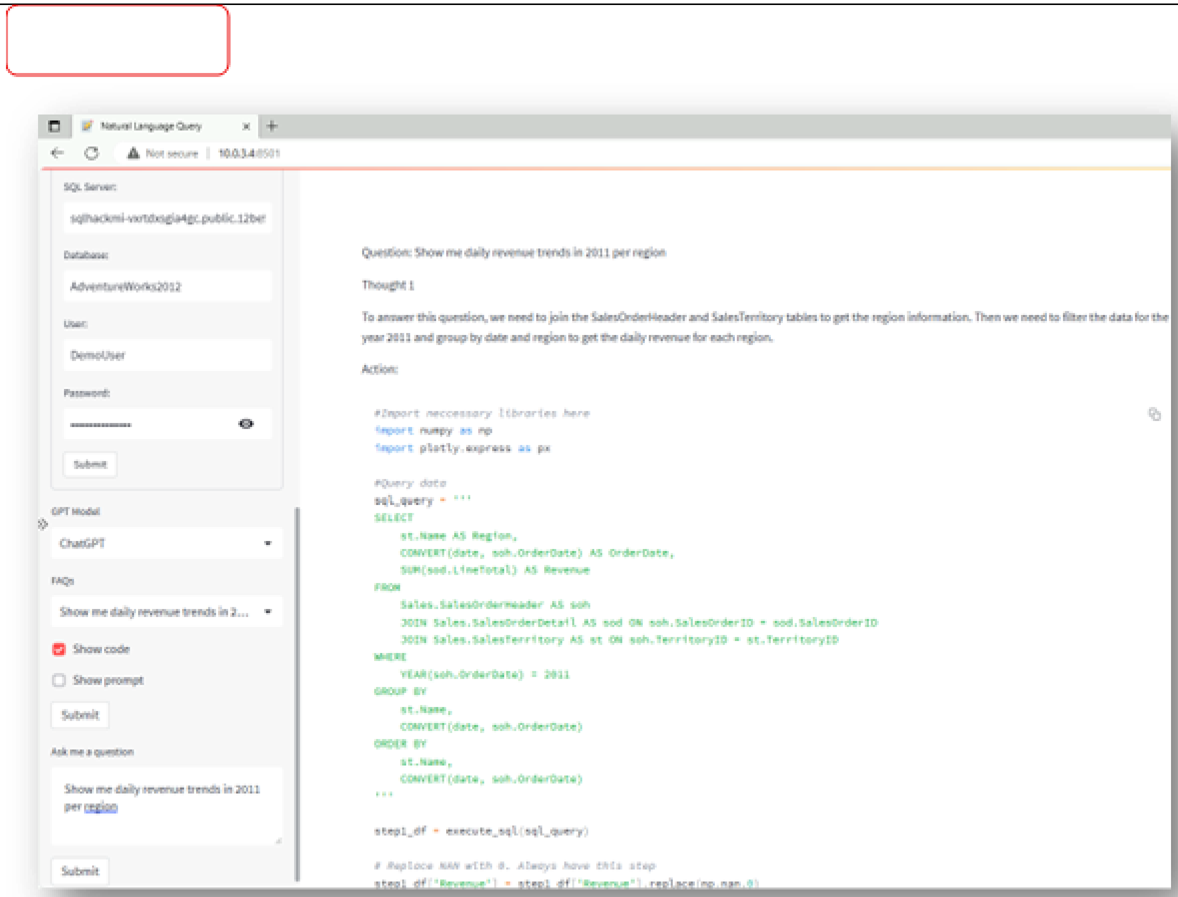
Question: For each category, get the list of products sold and the total sales amount

SQL Code

```
SELECT pc.Name AS CategoryName, p.Name AS ProductName, SUM(sod.Li
FROM Production.Product p
JOIN Production.ProductSubcategory pc ON p.ProductSubcategoryID =
JOIN Sales.SalesOrderDetail sod ON p.ProductID = sod.ProductID
GROUP BY pc.Name, p.Name
ORDER BY TotalSalesAmount DESC
```

	CategoryName	ProductName	TotalSalesAmount
0	Mountain Bikes	Mountain-200 Black, 38	4,400,592.8004
1	Mountain Bikes	Mountain-200 Black, 42	4,009,494.7618
2	Mountain Bikes	Mountain-200 Silver, 38	3,693,678.0253
3	Mountain Bikes	Mountain-200 Silver, 42	3,438,478.8604
4	Mountain Bikes	Mountain-200 Silver, 46	3,434,256.9419
5	Mountain Bikes	Mountain-200 Black, 46	3,309,673.2169
6	Road Bikes	Road-250 Black, 44	2,516,857.3149
7	Road Bikes	Road-250 Black, 48	2,347,655.9535
8	Road Bikes	Road-250 Black, 52	2,012,447.775
9	Road Bikes	Road-150 Red, 56	1,847,818.628

Now, let’s test the second assistant “Data Analysis Assistant” to generate further insights – also using natural language.

Narrative	Screenshot	Notes
In the UI, switch to the “**Data Analysis Assistant**”. **Make sure to select “sqlserver” as SQL Engine under connecting settings. Then hit submit.** Now, you can paste the following query and hit submit: ***Sample 1**: Show me daily revenue trends in 2011 per region*		

SQL Server: sqlhackmi-vxrtdxsgia4gc.public.12be

Database: AdventureWorks2012

User: DemoUser

GPT Model: ChatGPT

FAQs: Show me daily revenue trends in 2...

[✓ Show code] Show prompt

Ask me a question: Show me daily revenue trends in 2011 per region

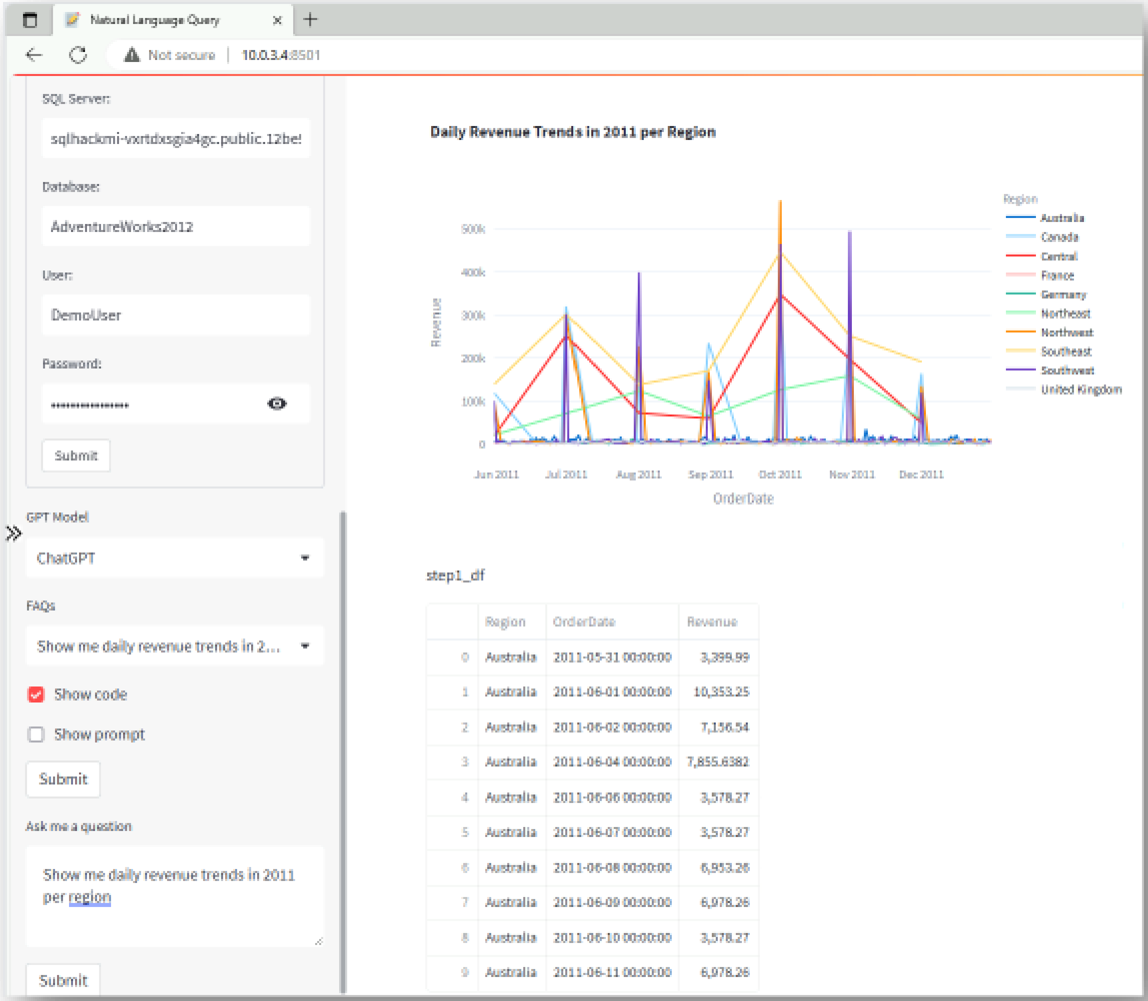
Question: Show me daily revenue trends in 2011 per region

Thought 1

To answer this question, we need to join the SalesOrderHeader and SalesTerritory tables to get the region information. Then we need to filter the data for the year 2011 and group by date and region to get the daily revenue for each region.

Action:

```
#Import neccessary libraries here
import numpy as np
import plotly.express as px

#Query data
sql_query = '''
SELECT
    st.Name AS Region,
    CONVERT(date, soh.OrderDate) AS OrderDate,
    SUM(sod.LineTotal) AS Revenue
FROM
    Sales.SalesOrderHeader AS soh
    JOIN Sales.SalesOrderDetail AS sod ON soh.SalesOrderID = sod.SalesOrderID
    JOIN Sales.SalesTerritory AS st ON soh.TerritoryID = st.TerritoryID
WHERE
    YEAR(soh.OrderDate) = 2011
GROUP BY
    st.Name,
    CONVERT(date, soh.OrderDate)
ORDER BY
    st.Name,
    CONVERT(date, soh.OrderDate)
'''

step1_df = execute_sql(sql_query)

# Replace NAN with 0. Always have this step
step1_df['Revenue'] = step1_df['Revenue'].replace(np.nan,0)
```



SQL Server: sqlhackmi-vxrtdxsgia4gc.public.12be!

Database: AdventureWorks2012

User: DemoUser

Password: ••••••••••••••••

GPT Model: ChatGPT

FAQs: Show me daily revenue trends in 2...

- [✓] Show code
- [ ] Show prompt

Submit

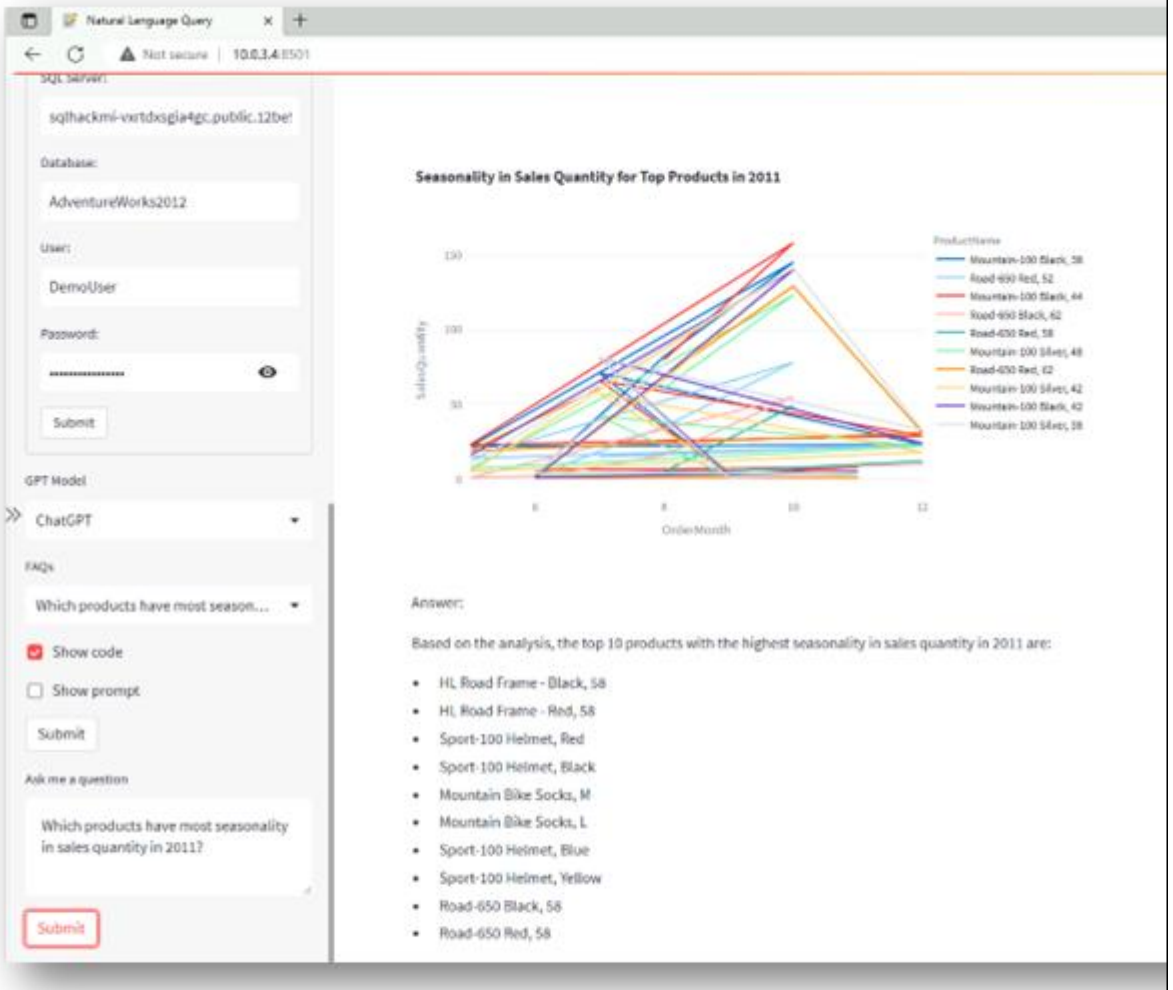
Ask me a question

Show me daily revenue trends in 2011 per region

Submit

step1_df

	Region	OrderDate	Revenue
0	Australia	2011-05-31 00:00:00	3,399.99
1	Australia	2011-06-01 00:00:00	10,353.25
2	Australia	2011-06-02 00:00:00	7,156.54
3	Australia	2011-06-04 00:00:00	7,855.6382
4	Australia	2011-06-06 00:00:00	3,578.27
5	Australia	2011-06-07 00:00:00	3,578.27
6	Australia	2011-06-08 00:00:00	6,953.26
7	Australia	2011-06-09 00:00:00	6,978.26
8	Australia	2011-06-10 00:00:00	3,578.27
9	Australia	2011-06-11 00:00:00	6,978.26

Sample 2: ***Which products have most seasonality in sales quantity in 2011?***		

SQL Server: sqlhackmi-vxrtdxsgia4gc.public.12be!

Database: AdventureWorks2012

User: DemoUser

Password: ••••••••••••••••

GPT Model: ChatGPT

FAQs: Which products have most season...

[✓] Show code

[ ] Show prompt

Ask me a question: Which products have most seasonality in sales quantity in 2011?

Answer:

Based on the analysis, the top 10 products with the highest seasonality in sales quantity in 2011 are:

- HL Road Frame - Black, 58
- HL Road Frame - Red, 58
- Sport-100 Helmet, Red
- Sport-100 Helmet, Black
- Mountain Bike Socks, M
- Mountain Bike Socks, L
- Sport-100 Helmet, Blue
- Sport-100 Helmet, Yellow
- Road-650 Black, 58
- Road-650 Red, 58

Microsoft

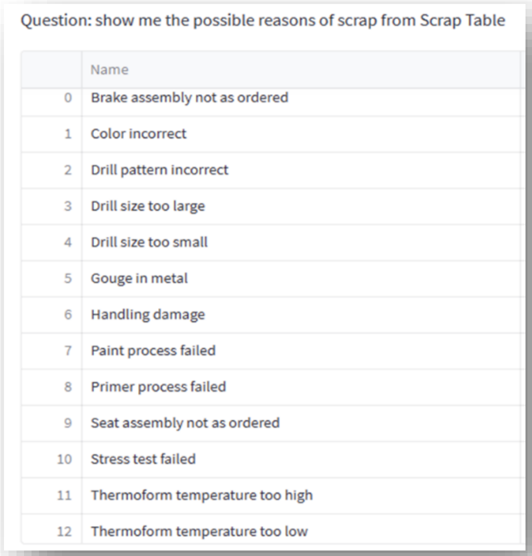
3. Insert new data via natural language

The task is dedicated to presentation of diverse ways how the initial logic of communication with your SQL DB data might be extended. For this Hackathon, we have worked with INSERT function (flow is presented below). It is worth mentioning that the function was implemented as a trial by Hackathon presenters.

Use Case Scenario: Recording Scrap Data in LocalMasterDataDB.

Background: LocalMasterDataDB is a manufacturing plant that produces a variety of products. To maintain quality control and minimize waste, the plant has a Scrap table in its database to record and analyze scrapped parts and materials. (Note: there are special systems that collect information about the scrapped parts on the production lines at plants. The presented Use Case is just an example of insert functionality).

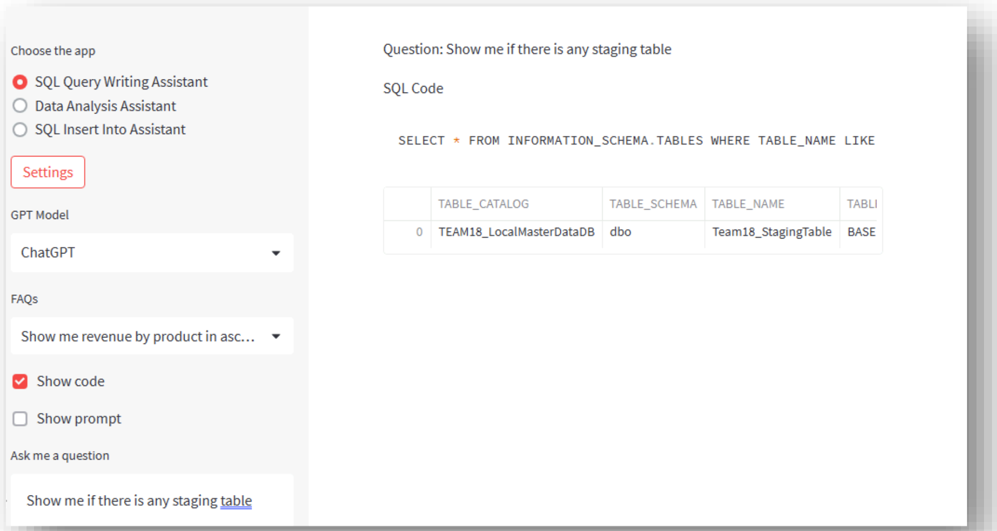
Scenario: In the LocalMasterDataDB manufacturing plant, the Scrap Redemption Center plays a crucial role in documenting and managing scrap data. To provide information about the reason of part rejection, the worker enters information about the type of scrap (e.g., "Gouge in metal," " Drill size too large").

Narrative	Screenshot	Notes
Select the **SQL Query Writing Assistant.** Make sure in settings, **sqlserver** is selected as SQL Engine. Click **Submit** to save the settings. Check the reasons for the scrap parts with the following prompt: ***Show me the possible reasons of Scrap Parts***	 Question: show me the possible reasons of scrap from Scrap Table | | Name | | 0 | Brake assembly not as ordered | | 1 | Color incorrect | | 2 | Drill pattern incorrect | | 3 | Drill size too large | | 4 | Drill size too small | | 5 | Gouge in metal | | 6 | Handling damage | | 7 | Paint process failed | | 8 | Primer process failed | | 9 | Seat assembly not as ordered | | 10 | Stress test failed | | 11 | Thermoform temperature too high | | 12 | Thermoform temperature too low |	

Insert with the Staging Table

Narrative	Screenshot/Code	Notes

Back in **SSMS** (SQL Server Management Studio), open a query window within your team **LocalMasterDataDB** database (e.g. TEAM01_LocalMasterDataDB).
You can do so by right clicking on the DB and selecting "**New Query**".
 	
Make sure you are in the right context (your Team DB) before executing the query.
 	
Create a **staging table** with the T-SQL statement on the right and insert your TeamXX name for the table.
	`CREATE TABLE [TeamXX_StagingTable] (`
	`    ScrapID INT IDENTITY(1,1) PRIMARY KEY, -- Auto-incremented unique identifier for each scrap record`
	`    ProductID INT,                              -- Foreign key to identify the product associated with the scrap`
	`    ScrappedQty INT,                          -- Quantity of scrapped parts`
	`    ScrapReasonID INT,                          -- Foreign key to specify the reason for scrap`
	`    ScrapDate DATETIME DEFAULT GETDATE()  -- Date and time of scrap (default to current timestamp)`
	`);`	

Check if the table is already there with the help of **SQL Query Writing Assistant**. Use the prompt: ***Show me if there is any staging table***		

Choose the app

- [✓ SQL Query Writing Assistant]
- Data Analysis Assistant
- SQL Insert Into Assistant

Settings

GPT Model: ChatGPT

FAQs: Show me revenue by product in asc…

[✓] Show code

[ ] Show prompt

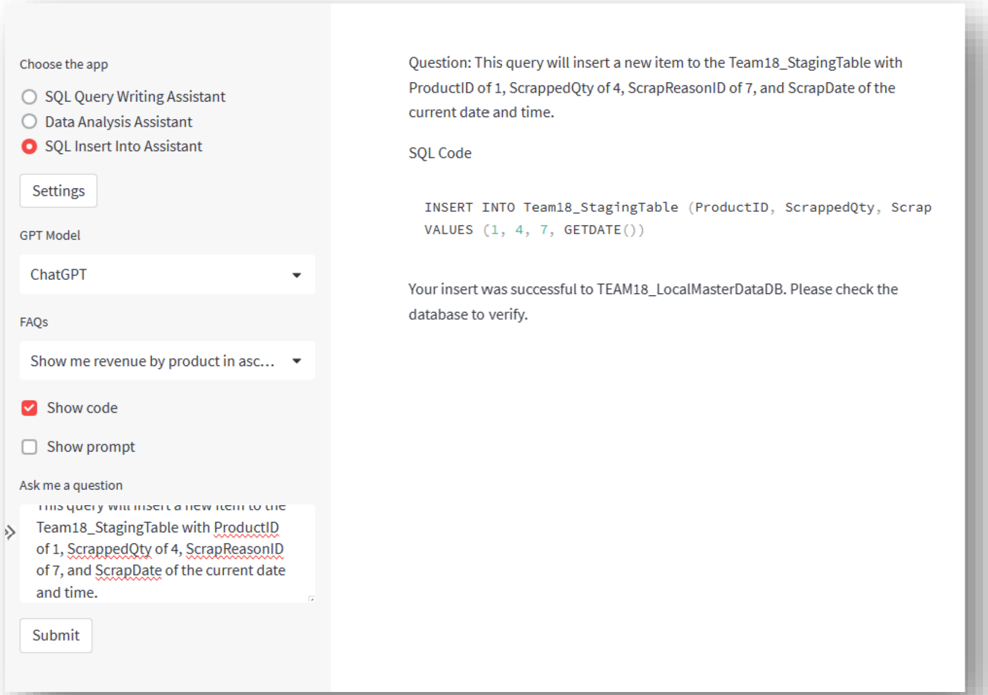
Ask me a question: Show me if there is any staging table

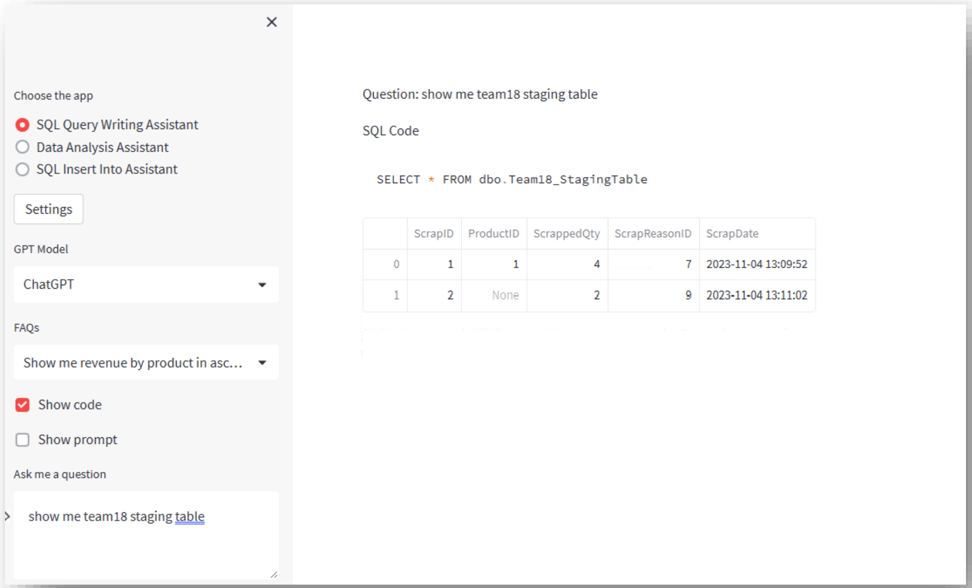
Question: Show me if there is any staging table

SQL Code

```
SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME LIKE
```

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLI
0	TEAM18_LocalMasterDataDB	dbo	Team18_StagingTable	BASE

Now we are fine to add our first information about the scrap parts (add your values here).
 	
Switch to **SQL Insert Into Assistant**.
Make sure to select "sqlserver" as SQL Engine in the Settings and hit submit.
 	
Now paste this prompt (**adjust your staging table name**):
 	
Insert a new item to TeamXX_StagingTable where productID is 1, scrap quantity is 4 and scrap reason is 7.
	 Question: This query will insert a new item to the Team18_StagingTable with ProductID of 1, ScrappedQty of 4, ScrapReasonID of 7, and ScrapDate of the current date and time.
	
	SQL Code
	
	`INSERT INTO Team18_StagingTable (ProductID, ScrappedQty, Scrap`
	`VALUES (1, 4, 7, GETDATE())`
	
	Your insert was successful to TEAM18_LocalMasterDataDB. Please check the database to verify.
	
	Ask me a question: This query will insert a new item to the Team18_StagingTable with ProductID of 1, ScrappedQty of 4, ScrapReasonID of 7, and ScrapDate of the current date and time.	

Insert a new value and check that the primary key (ScrapID) is automatically increased with the following prompt (again, adjust to your staging table name: ***Insert a new item to the TeamXX_StagingTable with ScrappedQty of 2, ScrapReasonID of 9, and ScrapDate of the current date and time.*** **Verify** in **SSMS** or in the **SQL Server Writing Assistant** that the item was inserted and there was an automatic increase of ScrapID.		

×
Choose the app
SQL Query Writing Assistant
Data Analysis Assistant
SQL Insert Into Assistant
Settings
GPT Model
ChatGPT
FAQs
Show me revenue by product in asc…
Show code
Show prompt
Ask me a question
show me team18 staging table

Question: show me team18 staging table

SQL Code

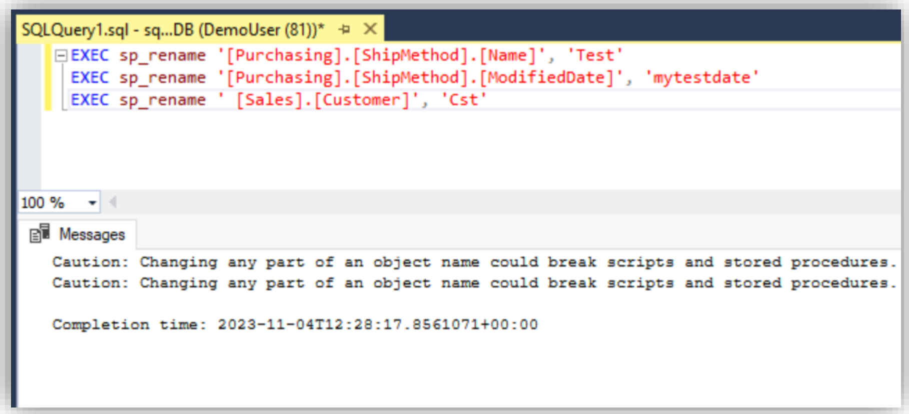
```
SELECT * FROM dbo.Team18_StagingTable
```

	ScrapID	ProductID	ScrappedQty	ScrapReasonID	ScrapDate
0	1	1	4	7	2023-11-04 13:09:52
1	2	None	2	9	2023-11-04 13:11:02

4. Resolve data quality issues

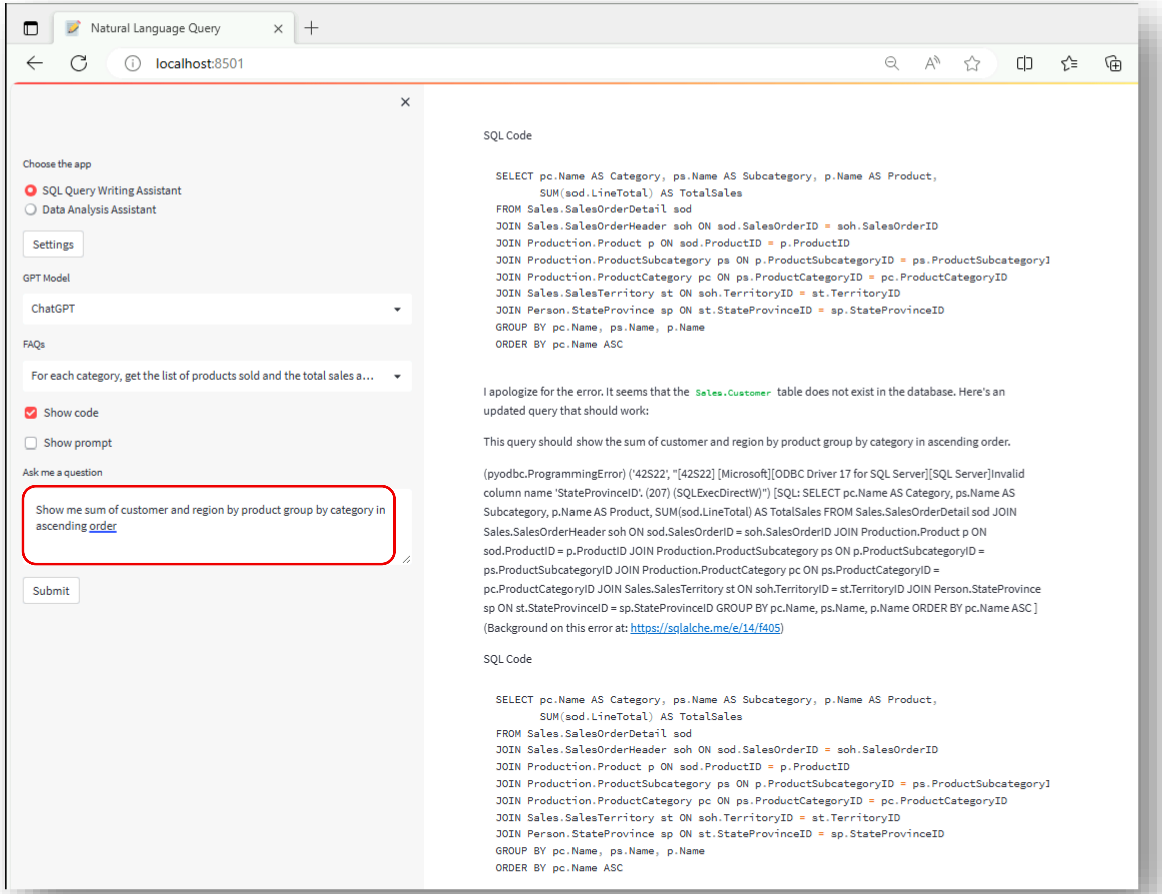
The purpose of this exercise is to understand the importance of ensuring the data which Azure OpenAI works with has the expected quality. First, let us set up the data quality issues by changing object names in our database. We will then analyse the data quality by checking the database object and naming.

Narrative	Screenshot	Notes

First, let's intentionally create "quality issues within our DB.
 	
In SSMS, open a query window on your TeamXX_LocalMasterDataDB and execute the following T-SQL statement:
 	
EXEC sp_rename '[Purchasing].[ShipMethod].[Name]', 'Test'
EXEC sp_rename '[Purchasing].[ShipMethod].[ModifiedDate]', 'mytestdate'
EXEC sp_rename '[Sales].[Customer]', 'Cst'
	
	SQLQuery1.sql - sq...DB (DemoUser (81))*
	`EXEC sp_rename '[Purchasing].[ShipMethod].[Name]', 'Test'`
	`EXEC sp_rename '[Purchasing].[ShipMethod].[ModifiedDate]', 'mytestdate'`
	`EXEC sp_rename ' [Sales].[Customer]', 'Cst'`
	100 %
	Messages
	`Caution: Changing any part of an object name could break scripts and stored procedures.`
	`Caution: Changing any part of an object name could break scripts and stored procedures.`
	
	`Completion time: 2023-11-04T12:28:17.8561071+00:00`	

Now, let's try some samples which will run into errors to understand potential data quality issues we would have to resolve.

Microsoft

Narrative	Screenshot	Notes
Sample 1: Copy the text in the question box, still using SQL Query Assistant: ***Show me customer by product group by category in ascending order***		

Natural Language Query

localhost:8501

Choose the app

[✓ SQL Query Writing Assistant]
Data Analysis Assistant

GPT Model

ChatGPT

FAQs

For each category, get the list of products sold and the total sales a...

[✓ Show code]

Show prompt

Ask me a question

Show me sum of customer and region by product group by category in ascending order

SQL Code

```
SELECT pc.Name AS Category, ps.Name AS Subcategory, p.Name AS Product,
       SUM(sod.LineTotal) AS TotalSales
FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
JOIN Production.Product p ON sod.ProductID = p.ProductID
JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID = ps.ProductSubcategoryI
JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID
JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID
JOIN Person.StateProvince sp ON st.StateProvinceID = sp.StateProvinceID
GROUP BY pc.Name, ps.Name, p.Name
ORDER BY pc.Name ASC
```

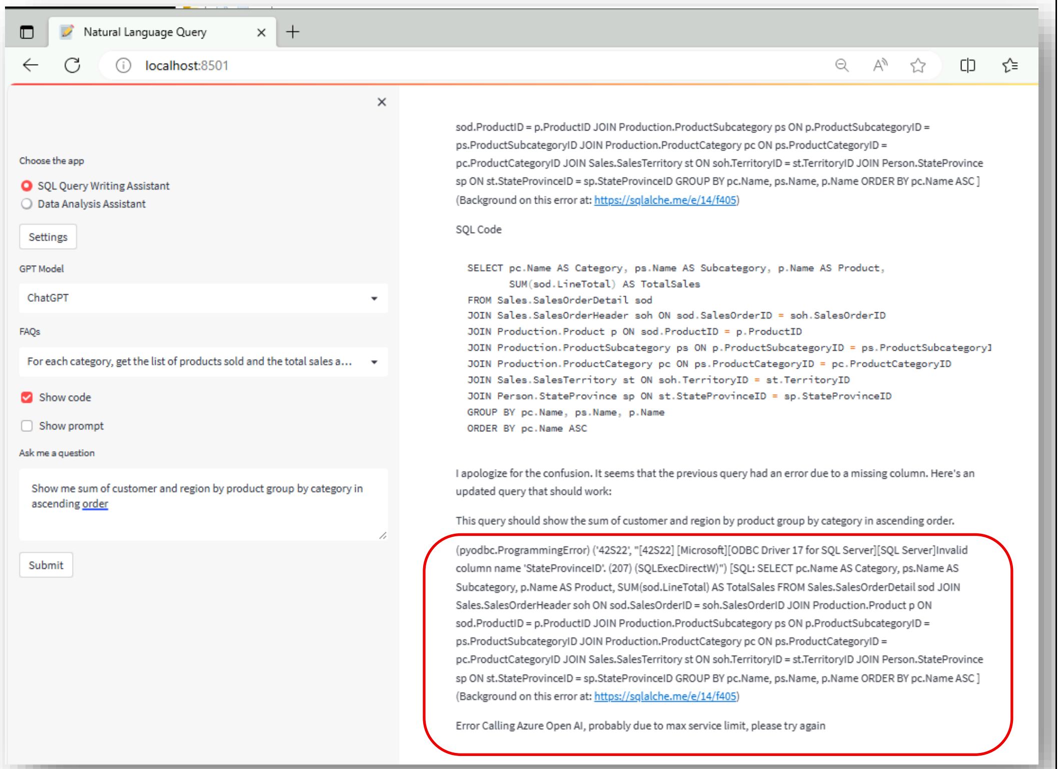
I apologize for the error. It seems that the `Sales.Customer` table does not exist in the database. Here's an updated query that should work:

This query should show the sum of customer and region by product group by category in ascending order.

```
(pyodbc.ProgrammingError) ('42S22', "[42S22] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid
column name 'StateProvinceID'. (207) (SQLExecDirectW)") [SQL: SELECT pc.Name AS Category, ps.Name AS
Subcategory, p.Name AS Product, SUM(sod.LineTotal) AS TotalSales FROM Sales.SalesOrderDetail sod JOIN
Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID JOIN Production.Product p ON
sod.ProductID = p.ProductID JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID =
ps.ProductSubcategoryID JOIN Production.ProductCategory pc ON ps.ProductCategoryID =
pc.ProductCategoryID JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID JOIN Person.StateProvince
sp ON st.StateProvinceID = sp.StateProvinceID GROUP BY pc.Name, ps.Name, p.Name ORDER BY pc.Name ASC ]
(Background on this error at: https://sqlalche.me/e/14/f405)
```

SQL Code

```
SELECT pc.Name AS Category, ps.Name AS Subcategory, p.Name AS Product,
       SUM(sod.LineTotal) AS TotalSales
FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
JOIN Production.Product p ON sod.ProductID = p.ProductID
JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID = ps.ProductSubcategoryI
JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID
JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID
JOIN Person.StateProvince sp ON st.StateProvinceID = sp.StateProvinceID
GROUP BY pc.Name, ps.Name, p.Name
ORDER BY pc.Name ASC
```

Notice an **error** is thrown. You may get a similar message such as "*I apologize for the confusion. It seems there was an error in the previous query. The 'Sales.Customer' table does not exist in the database. Instead, we have a 'Sales.Cst' table that contains the 'CustomerID'.*". The reason is due to the **object naming** in the database. We purposely changed [Sales].Customer to [Sales].CST. The Azure OpenAI model converts the user query to T-SQL (Transact-SQL). However, since no table with the name "Customer" exists, it cannot query from the table.	

Natural Language Query

localhost:8501

Choose the app

[✓ SQL Query Writing Assistant]

Data Analysis Assistant

GPT Model: ChatGPT

FAQs: For each category, get the list of products sold and the total sales a…

[✓] Show code

[ ] Show prompt

Ask me a question: Show me sum of customer and region by product group by category in ascending order

```
sod.ProductID = p.ProductID JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID =
ps.ProductSubcategoryID JOIN Production.ProductCategory pc ON ps.ProductCategoryID =
pc.ProductCategoryID JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID JOIN Person.StateProvince
sp ON st.StateProvinceID = sp.StateProvinceID GROUP BY pc.Name, ps.Name, p.Name ORDER BY pc.Name ASC ]
(Background on this error at: https://sqlalche.me/e/14/f405)
```

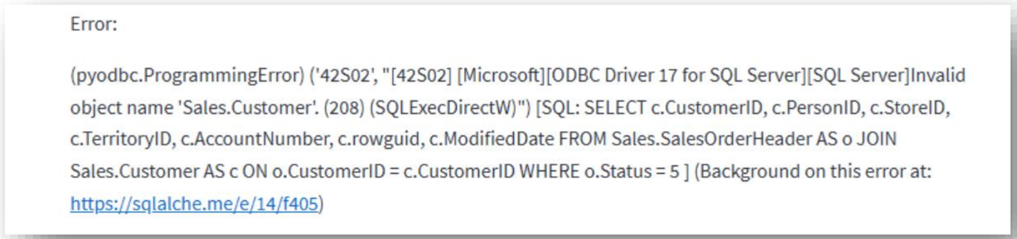
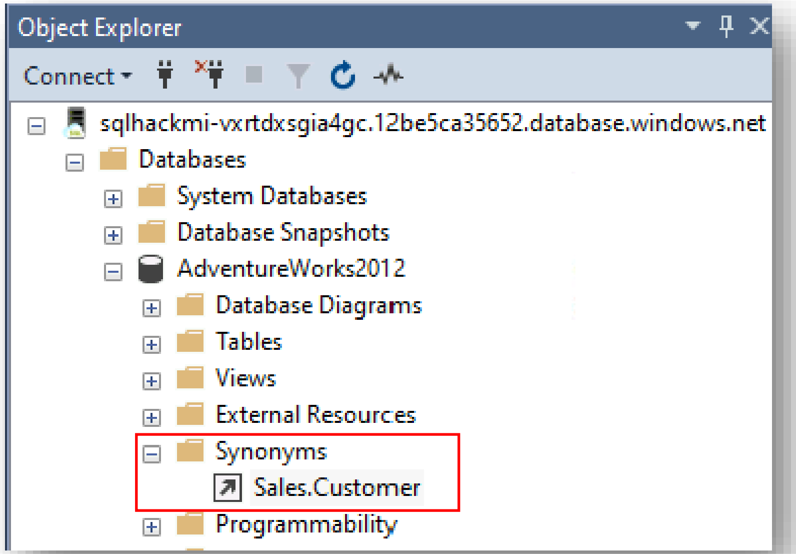
SQL Code

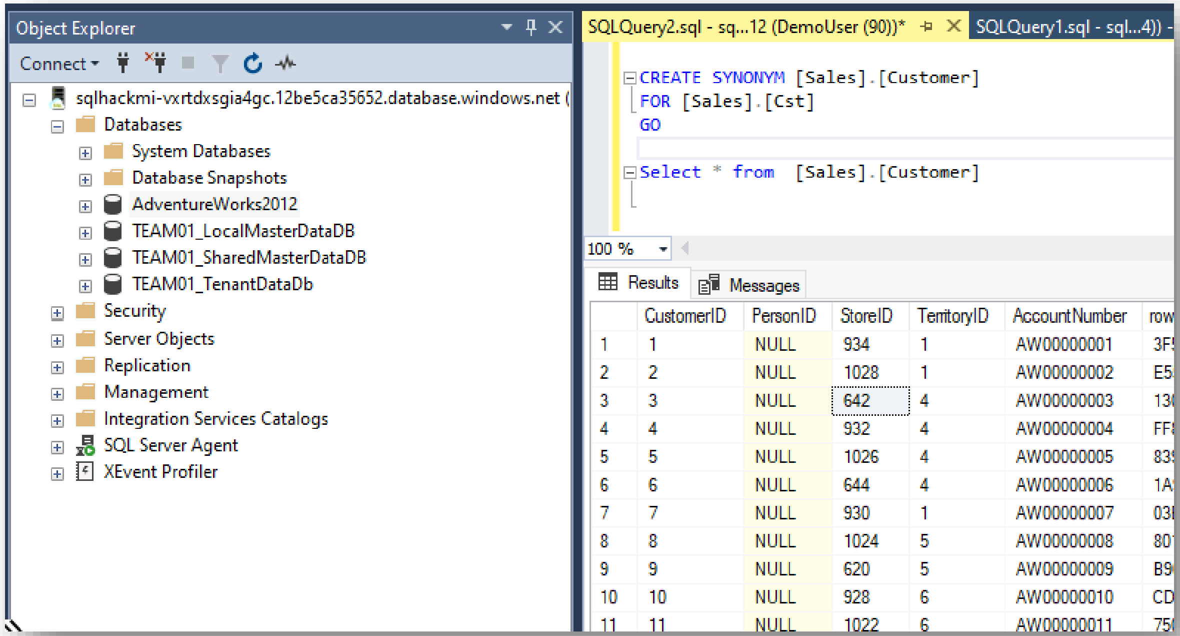
```
SELECT pc.Name AS Category, ps.Name AS Subcategory, p.Name AS Product,
       SUM(sod.LineTotal) AS TotalSales
FROM Sales.SalesOrderDetail sod
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID
JOIN Production.Product p ON sod.ProductID = p.ProductID
JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID = ps.ProductSubcategoryI
JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.ProductCategoryID
JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID
JOIN Person.StateProvince sp ON st.StateProvinceID = sp.StateProvinceID
GROUP BY pc.Name, ps.Name, p.Name
ORDER BY pc.Name ASC
```

I apologize for the confusion. It seems that the previous query had an error due to a missing column. Here's an updated query that should work:

This query should show the sum of customer and region by product group by category in ascending order.

```
(pyodbc.ProgrammingError) ('42S22', "[42S22] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid
column name 'StateProvinceID'. (207) (SQLExecDirectW)") [SQL: SELECT pc.Name AS Category, ps.Name AS
Subcategory, p.Name AS Product, SUM(sod.LineTotal) AS TotalSales FROM Sales.SalesOrderDetail sod JOIN
Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesOrderID JOIN Production.Product p ON
sod.ProductID = p.ProductID JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID =
ps.ProductSubcategoryID JOIN Production.ProductCategory pc ON ps.ProductCategoryID =
pc.ProductCategoryID JOIN Sales.SalesTerritory st ON soh.TerritoryID = st.TerritoryID JOIN Person.StateProvince
sp ON st.StateProvinceID = sp.StateProvinceID GROUP BY pc.Name, ps.Name, p.Name ORDER BY pc.Name ASC ]
(Background on this error at: https://sqlalche.me/e/14/f405)

Error Calling Azure Open AI, probably due to max service limit, please try again
```

You can also test in the Data Analysis Assistant: *Which customers are most likely to churn?* This will also throw an error since it also refers to the customer table.	 Error: (pyodbc.ProgrammingError) ('42S02', "[42S02] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid object name 'Sales.Customer'. (208) (SQLExecDirectW)") [SQL: SELECT c.CustomerID, c.PersonID, c.StoreID, c.TerritoryID, c.AccountNumber, c.rowguid, c.ModifiedDate FROM Sales.SalesOrderHeader AS o JOIN Sales.Customer AS c ON o.CustomerID = c.CustomerID WHERE o.Status = 5] (Background on this error at: https://sqlalche.me/e/14/f405)	
FIX: In SSMS, create a **synonym** "sales.customer" for the "sales.cst" table. Within your TEAMXX_LocalMasterDataDB, open a new query window and run the following T-SQL statement: *CREATE SYNONYM [Sales].[Customer]* *FOR [Sales].[Cst]* *GO* **Review** the newly created synonym as shown in the screenshot.	 Object Explorer Connect sqlhackmi-vxrtdxsgia4gc.12be5ca35652.database.windows.net Databases System Databases Database Snapshots AdventureWorks2012 Database Diagrams Tables Views External Resources Synonyms Sales.Customer Programmability	Learn more about synonyms in our documentation: [CREATE SYNONYM (Transact-SQL) - SQL Server | Microsoft Learn]

Execute the T-sql query:: *Select * from [Sales].[Customer]*		

Object Explorer

Connect

- sqlhackmi-vxrtdxsgia4gc.12be5ca35652.database.windows.net (
  - Databases
    - System Databases
    - Database Snapshots
    - AdventureWorks2012
    - TEAM01_LocalMasterDataDB
    - TEAM01_SharedMasterDataDB
    - TEAM01_TenantDataDb
  - Security
  - Server Objects
  - Replication
  - Management
  - Integration Services Catalogs
  - SQL Server Agent
  - XEvent Profiler

SQLQuery2.sql - sq...12 (DemoUser (90))* SQLQuery1.sql - sql...4)) -

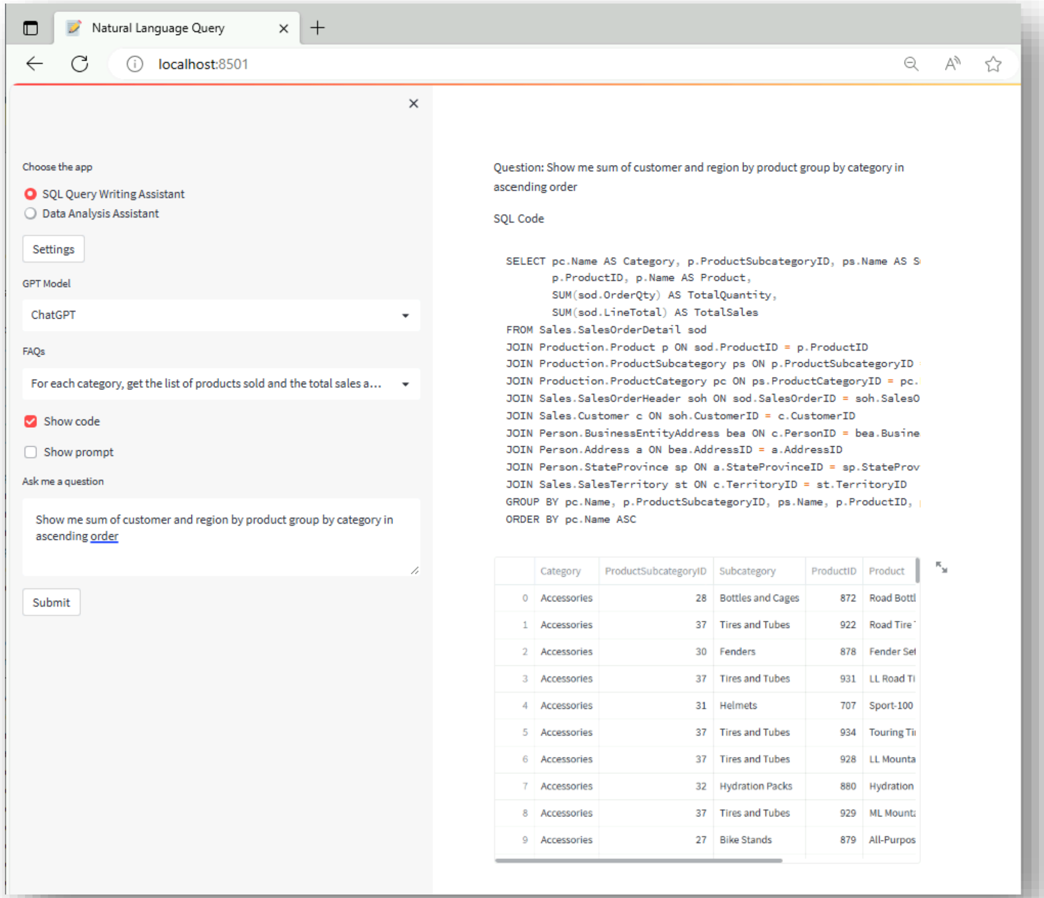
```
CREATE SYNONYM [Sales].[Customer]
FOR [Sales].[Cst]
GO

Select * from  [Sales].[Customer]
```

100 %

Results Messages

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	row
1	1	NULL	934	1	AW00000001	3F
2	2	NULL	1028	1	AW00000002	E5
3	3	NULL	642	4	AW00000003	13
4	4	NULL	932	4	AW00000004	FF
5	5	NULL	1026	4	AW00000005	83
6	6	NULL	644	4	AW00000006	1A
7	7	NULL	930	1	AW00000007	03
8	8	NULL	1024	5	AW00000008	80
9	9	NULL	620	5	AW00000009	B9
10	10	NULL	928	6	AW00000010	CD
11	11	NULL	1022	6	AW00000011	75

Microsoft

Submit the question again and view the result:		

Choose the app

[✓ SQL Query Writing Assistant]
Data Analysis Assistant

GPT Model: ChatGPT

FAQs: For each category, get the list of products sold and the total sales a...

[✓ Show code]
Show prompt

Ask me a question

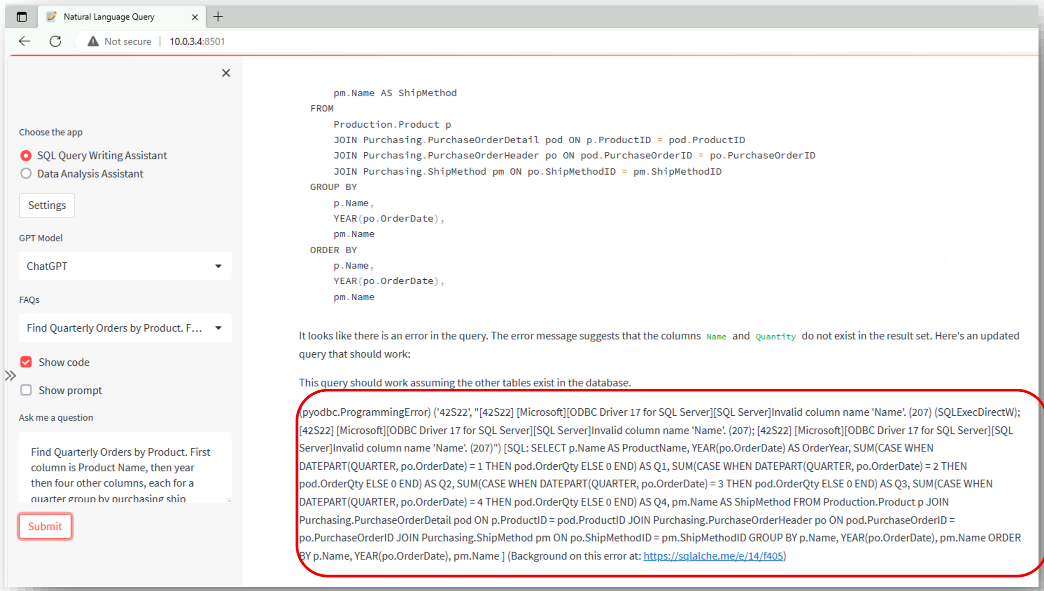
Show me sum of customer and region by product group by category in ascending order

Question: Show me sum of customer and region by product group by category in ascending order

SQL Code

```
SELECT pc.Name AS Category, p.ProductSubcategoryID, ps.Name AS S
       p.ProductID, p.Name AS Product,
       SUM(sod.OrderQty) AS TotalQuantity,
       SUM(sod.LineTotal) AS TotalSales
FROM Sales.SalesOrderDetail sod
JOIN Production.Product p ON sod.ProductID = p.ProductID
JOIN Production.ProductSubcategory ps ON p.ProductSubcategoryID
JOIN Production.ProductCategory pc ON ps.ProductCategoryID = pc.
JOIN Sales.SalesOrderHeader soh ON sod.SalesOrderID = soh.SalesO
JOIN Sales.Customer c ON soh.CustomerID = c.CustomerID
JOIN Person.BusinessEntityAddress bea ON c.PersonID = bea.Busine
JOIN Person.Address a ON bea.AddressID = a.AddressID
JOIN Person.StateProvince sp ON a.StateProvinceID = sp.StateProv
JOIN Sales.SalesTerritory st ON c.TerritoryID = st.TerritoryID
GROUP BY pc.Name, p.ProductSubcategoryID, ps.Name, p.ProductID,
ORDER BY pc.Name ASC
```

	Category	ProductSubcategoryID	Subcategory	ProductID	Product
0	Accessories	28	Bottles and Cages	872	Road Bottl
1	Accessories	37	Tires and Tubes	922	Road Tire
2	Accessories	30	Fenders	878	Fender Set
3	Accessories	37	Tires and Tubes	931	LL Road Ti
4	Accessories	31	Helmets	707	Sport-100
5	Accessories	37	Tires and Tubes	934	Touring Ti
6	Accessories	37	Tires and Tubes	928	LL Mounta
7	Accessories	32	Hydration Packs	880	Hydration
8	Accessories	37	Tires and Tubes	929	ML Mount
9	Accessories	27	Bike Stands	879	All-Purpos

<u>Sample 2</u> Copy the text in the question box: ***Find Quarterly Orders by Product. First column is Product Name, then year then four other columns, each for a quarter group by ship method***		

Choose the app

[✓ SQL Query Writing Assistant] Data Analysis Assistant

GPT Model: ChatGPT

FAQs: Find Quarterly Orders by Product. F…

[✓ Show code] Show prompt

Ask me a question

Find Quarterly Orders by Product. First column is Product Name, then year then four other columns, each for a quarter group by purchasing ship

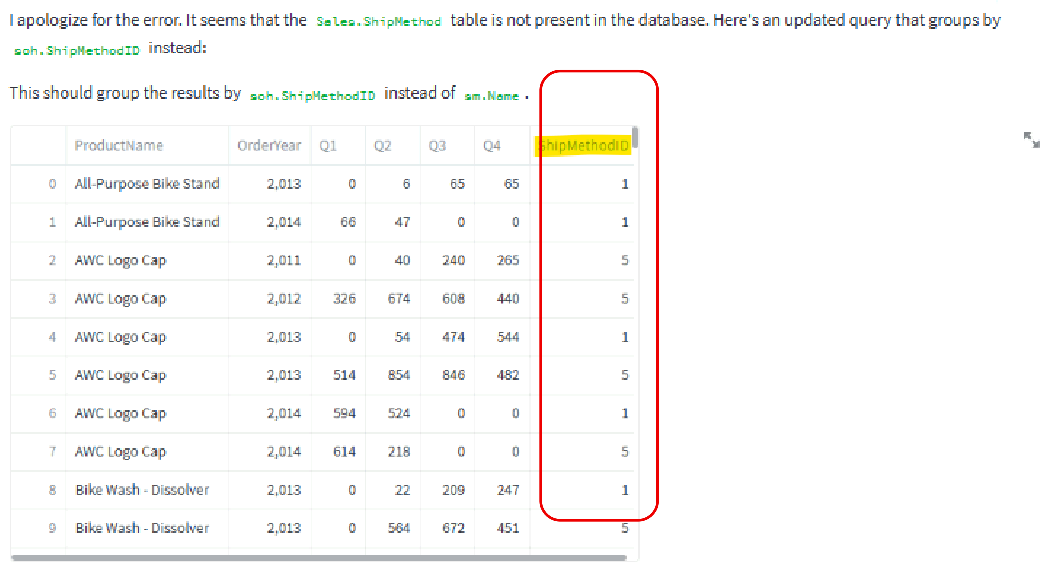
```
        pm.Name AS ShipMethod
    FROM
        Production.Product p
        JOIN Purchasing.PurchaseOrderDetail pod ON p.ProductID = pod.ProductID
        JOIN Purchasing.PurchaseOrderHeader po ON pod.PurchaseOrderID = po.PurchaseOrderID
        JOIN Purchasing.ShipMethod pm ON po.ShipMethodID = pm.ShipMethodID
    GROUP BY
        p.Name,
        YEAR(po.OrderDate),
        pm.Name
    ORDER BY
        p.Name,
        YEAR(po.OrderDate),
        pm.Name
```

It looks like there is an error in the query. The error message suggests that the columns `Name` and `Quantity` do not exist in the result set. Here's an updated query that should work:

This query should work assuming the other tables exist in the database.

```
(pyodbc.ProgrammingError) ('42S22', "[42S22] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid column name 'Name'. (207) (SQLExecDirectW); [42S22] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid column name 'Name'. (207); [42S22] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Invalid column name 'Name'. (207)") [SQL: SELECT p.Name AS ProductName, YEAR(po.OrderDate) AS OrderYear, SUM(CASE WHEN DATEPART(QUARTER, po.OrderDate) = 1 THEN pod.OrderQty ELSE 0 END) AS Q1, SUM(CASE WHEN DATEPART(QUARTER, po.OrderDate) = 2 THEN pod.OrderQty ELSE 0 END) AS Q2, SUM(CASE WHEN DATEPART(QUARTER, po.OrderDate) = 3 THEN pod.OrderQty ELSE 0 END) AS Q3, SUM(CASE WHEN DATEPART(QUARTER, po.OrderDate) = 4 THEN pod.OrderQty ELSE 0 END) AS Q4, pm.Name AS ShipMethod FROM Production.Product p JOIN Purchasing.PurchaseOrderDetail pod ON p.ProductID = pod.ProductID JOIN Purchasing.PurchaseOrderHeader po ON pod.PurchaseOrderID = po.PurchaseOrderID JOIN Purchasing.ShipMethod pm ON po.ShipMethodID = pm.ShipMethodID GROUP BY p.Name, YEAR(po.OrderDate), pm.Name ORDER BY p.Name, YEAR(po.OrderDate), pm.Name ] (Background on this error at: https://sqlalche.me/e/14/f405)
```

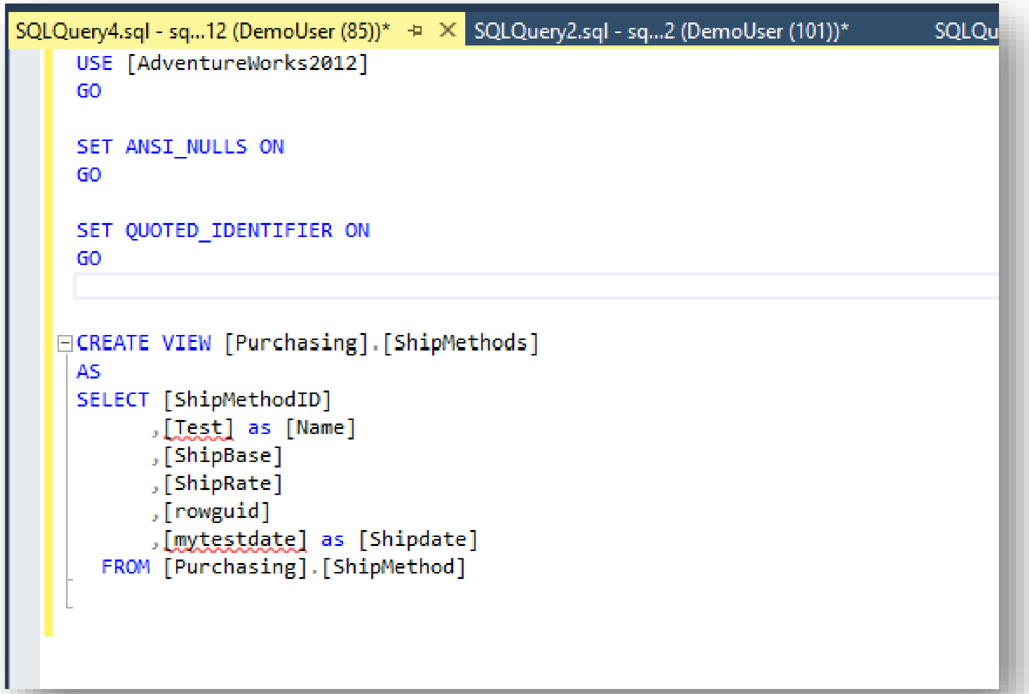
Cannot find the column “name” and therefore takes “ShipMethodID” as a reference.

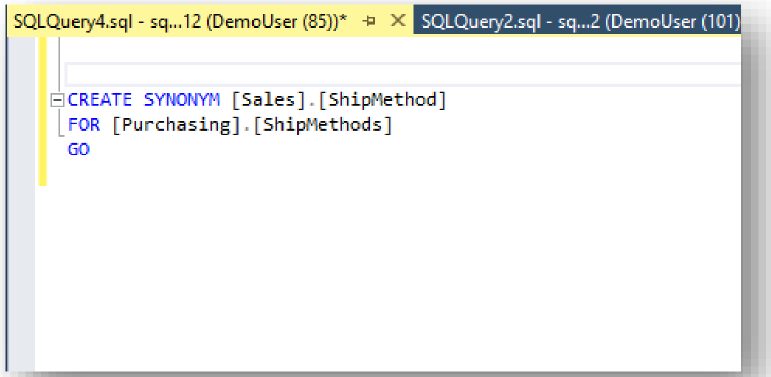


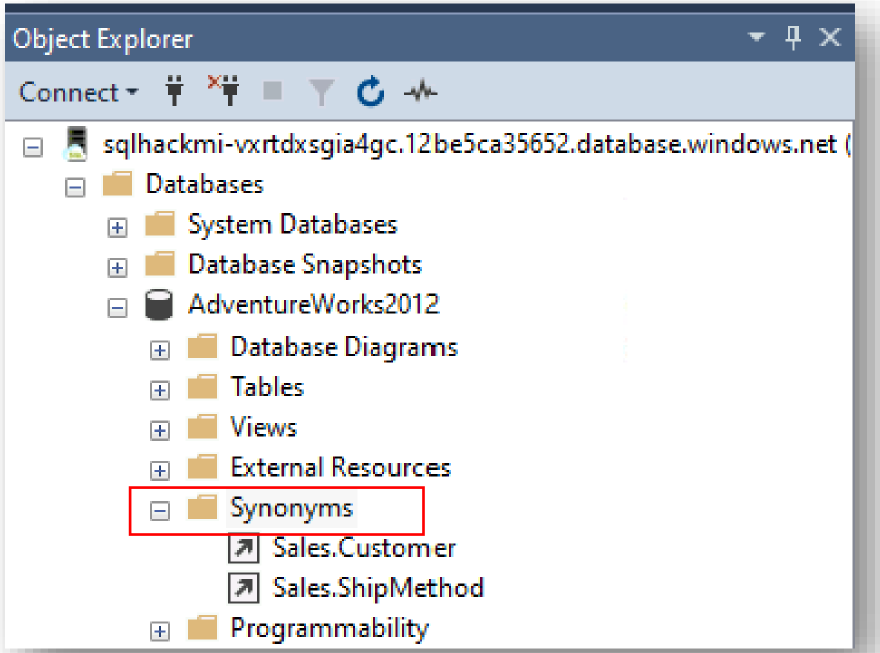
I apologize for the error. It seems that the `Sales.ShipMethod` table is not present in the database. Here's an updated query that groups by `soh.ShipMethodID` instead:

This should group the results by `soh.ShipMethodID` instead of `sm.Name`.

	ProductName	OrderYear	Q1	Q2	Q3	Q4	ShipMethodID
0	All-Purpose Bike Stand	2,013	0	6	65	65	1
1	All-Purpose Bike Stand	2,014	66	47	0	0	1
2	AWC Logo Cap	2,011	0	40	240	265	5
3	AWC Logo Cap	2,012	326	674	608	440	5
4	AWC Logo Cap	2,013	0	54	474	544	1
5	AWC Logo Cap	2,013	514	854	846	482	5
6	AWC Logo Cap	2,014	594	524	0	0	1
7	AWC Logo Cap	2,014	614	218	0	0	5
8	Bike Wash - Dissolver	2,013	0	22	209	247	1
9	Bike Wash - Dissolver	2,013	0	564	672	451	5

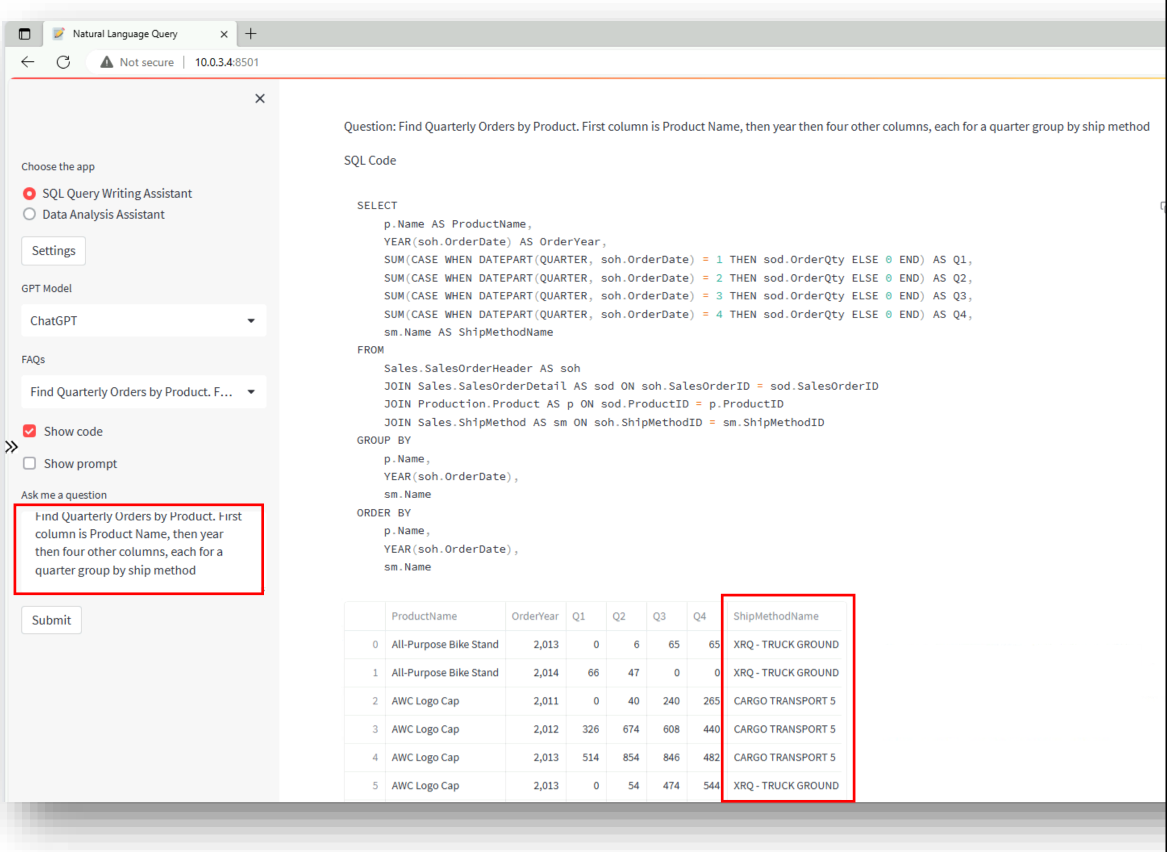
In SSMS, **create a View** via t-sql (adjust to your team DB):
 	
USE [TEAMXX_LocalMasterDataDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
 	
CREATE VIEW [Purchasing].[ShipMethods]
AS
SELECT [ShipMethodID] ,[Test] as [Name] ,[ShipBase] ,[ShipRate] ,[rowguid] ,[mytestdate] as [Shipdate]
FROM [Purchasing].[ShipMethod]
	
	SQLQuery4.sql - sq...12 (DemoUser (85))* SQLQuery2.sql - sq...2 (DemoUser (101))* SQLQu
	`USE [AdventureWorks2012]`
	`GO`
	`SET ANSI_NULLS ON`
	`GO`
	`SET QUOTED_IDENTIFIER ON`
	`GO`
	`CREATE VIEW [Purchasing].[ShipMethods]`
	`AS`
	`SELECT [ShipMethodID]`
	`,[Test] as [Name]`
	`,[ShipBase]`
	`,[ShipRate]`
	`,[rowguid]`
	`,[mytestdate] as [Shipdate]`
	`FROM [Purchasing].[ShipMethod]`	

Create a synonym:
 	
CREATE SYNONYM [Sales].[ShipMethod]
FOR [Purchasing].[ShipMethods]
GO
	
	SQLQuery4.sql - sq...12 (DemoUser (85))* SQLQuery2.sql - sq...2 (DemoUser (101))
	`CREATE SYNONYM [Sales].[ShipMethod]`
	`FOR [Purchasing].[ShipMethods]`
	`GO`	

Review the synonym	 sqlhackmi-vxrtdxsgia4gc.12be5ca35652.database.windows.net (Databases System Databases Database Snapshots AdventureWorks2012 Database Diagrams Tables Views External Resources Synonyms Sales.Customer Sales.ShipMethod Programmability	

Submit the question again and view the result

Find Quarterly Orders by Product. First column is Product Name, then year then four other columns, each for a quarter group by ship method



Natural Language Query

Not secure | 10.0.3.4:8501

Choose the app

[✓ SQL Query Writing Assistant] Data Analysis Assistant

Settings

GPT Model: ChatGPT

FAQs: Find Quarterly Orders by Product. F...

[✓ Show code] Show prompt

Ask me a question

Find Quarterly Orders by Product. First column is Product Name, then year then four other columns, each for a quarter group by ship method

Submit

Question: Find Quarterly Orders by Product. First column is Product Name, then year then four other columns, each for a quarter group by ship method

SQL Code

```
SELECT
    p.Name AS ProductName,
    YEAR(soh.OrderDate) AS OrderYear,
    SUM(CASE WHEN DATEPART(QUARTER, soh.OrderDate) = 1 THEN sod.OrderQty ELSE 0 END) AS Q1,
    SUM(CASE WHEN DATEPART(QUARTER, soh.OrderDate) = 2 THEN sod.OrderQty ELSE 0 END) AS Q2,
    SUM(CASE WHEN DATEPART(QUARTER, soh.OrderDate) = 3 THEN sod.OrderQty ELSE 0 END) AS Q3,
    SUM(CASE WHEN DATEPART(QUARTER, soh.OrderDate) = 4 THEN sod.OrderQty ELSE 0 END) AS Q4,
    sm.Name AS ShipMethodName
FROM
    Sales.SalesOrderHeader AS soh
    JOIN Sales.SalesOrderDetail AS sod ON soh.SalesOrderID = sod.SalesOrderID
    JOIN Production.Product AS p ON sod.ProductID = p.ProductID
    JOIN Sales.ShipMethod AS sm ON soh.ShipMethodID = sm.ShipMethodID
GROUP BY
    p.Name,
    YEAR(soh.OrderDate),
    sm.Name
ORDER BY
    p.Name,
    YEAR(soh.OrderDate),
    sm.Name
```

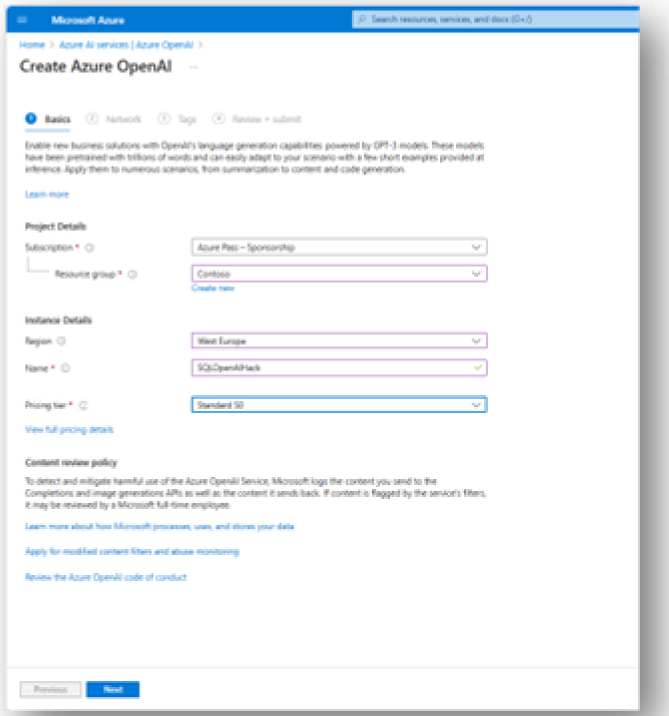
	ProductName	OrderYear	Q1	Q2	Q3	Q4	ShipMethodName
0	All-Purpose Bike Stand	2,013	0	6	65	65	XRQ - TRUCK GROUND
1	All-Purpose Bike Stand	2,014	66	47	0	0	XRQ - TRUCK GROUND
2	AWC Logo Cap	2,011	0	40	240	265	CARGO TRANSPORT 5
3	AWC Logo Cap	2,012	326	674	608	440	CARGO TRANSPORT 5
4	AWC Logo Cap	2,013	514	854	846	482	CARGO TRANSPORT 5
5	AWC Logo Cap	2,013	0	54	474	544	XRQ - TRUCK GROUND

Appendix

(OPTIONAL) Manual Deployment: You can follow the instructions below if you want to manually deploy the Azure OpenAI resource as well as the environment setup including Visual Studio Code, Git, Python, etc.

Azure OpenAI Service

Create an Azure OpenAI resource in an Azure subscription with a **GPT-35-Turbo** deployment and preferably a **GPT-4** deployment. Here we provide options for using both, but GPT-4 should be used to address more difficult & vague questions. We assume that your GPT-4 and ChatGPT deployments are in the same Azure Open AI resource.

Narrative	Screenshot	Notes
See documentation for instructions: How-to: Create and deploy an Azure OpenAI Service resource - Azure OpenAI | Microsoft Learn		

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Azure AI services | Azure OpenAI >
Basics
Network
Tags
Review + submit
Learn more
Create new
View full pricing details
Learn more about how Microsoft processes, uses, and stores your data
Apply for modified content filters and abuse monitoring
Review the Azure OpenAI code of conduct
Previous
Next

Create Azure OpenAI

Enable new business solutions with OpenAI's language generation capabilities powered by GPT-3 models. These models have been pretrained with trillions of words and can easily adapt to your scenario with a few short examples provided at inference. Apply them to numerous scenarios, from summarization to content and code generation.

Project Details

Subscription *: Azure Pass – Sponsorship

Resource group *: Contoso

Instance Details

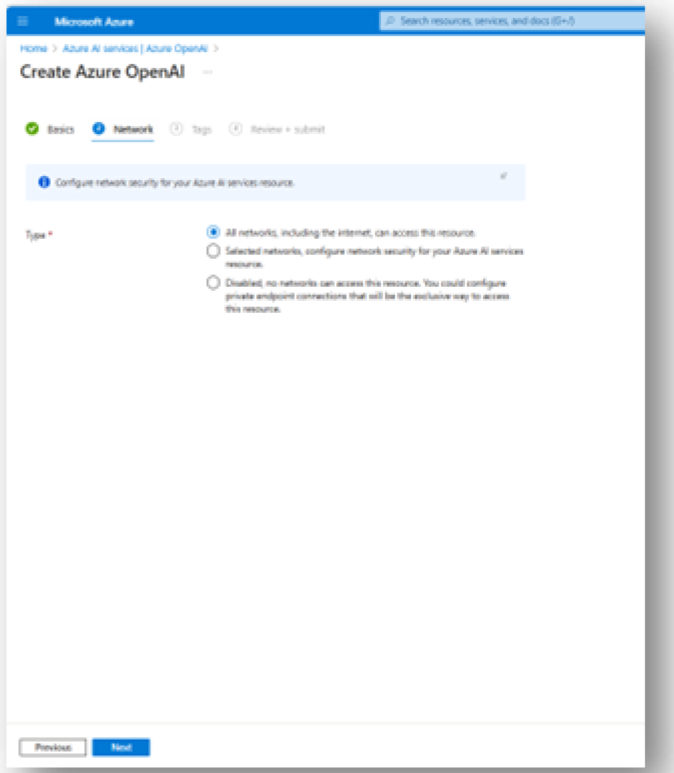
Region: West Europe

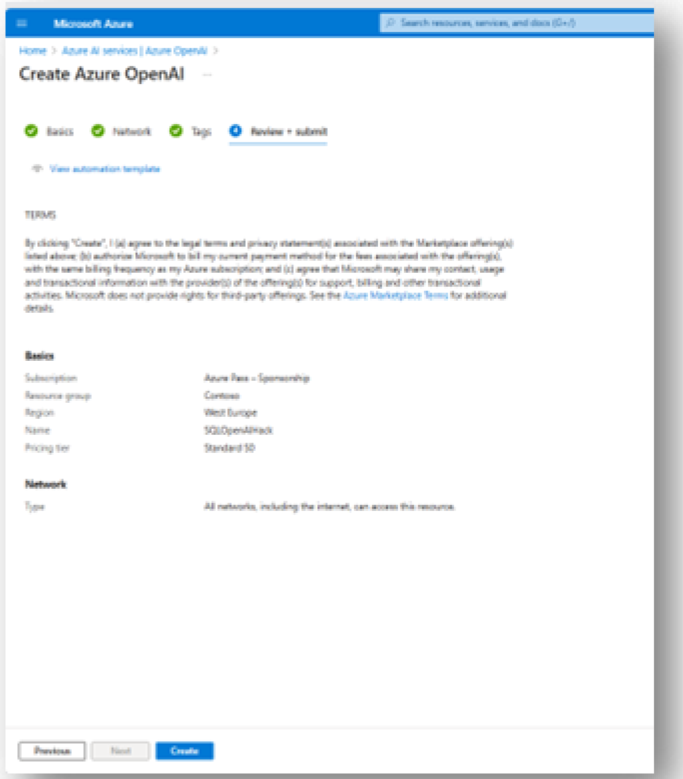
Name *: SQLOpenAIHack

Pricing tier *: Standard S0

Content review policy

To detect and mitigate harmful use of the Azure OpenAI Service, Microsoft logs the content you send to the Completions and image generations APIs as well as the content it sends back. If content is flagged by the service's filters, it may be reviewed by a Microsoft full-time employee.

	 Microsoft Azure · Home > Azure AI services | Azure OpenAI > · **Create Azure OpenAI** · Basics · Network · Tags · Review + submit · Configure network security for your Azure AI services resource. · Type * [✓ All networks, including the internet, can access this resource.] Selected networks, configure network security for your Azure AI services resource. Disabled, no networks can access this resource. You could configure private endpoint connections that will be the exclusive way to access this resource.	

Microsoft Azure
Search resources, services, and docs (G+/)
Home > Azure AI services | Azure OpenAI >

Create Azure OpenAI

Basics
Network
Tags
Review + submit
View automation template

TERMS

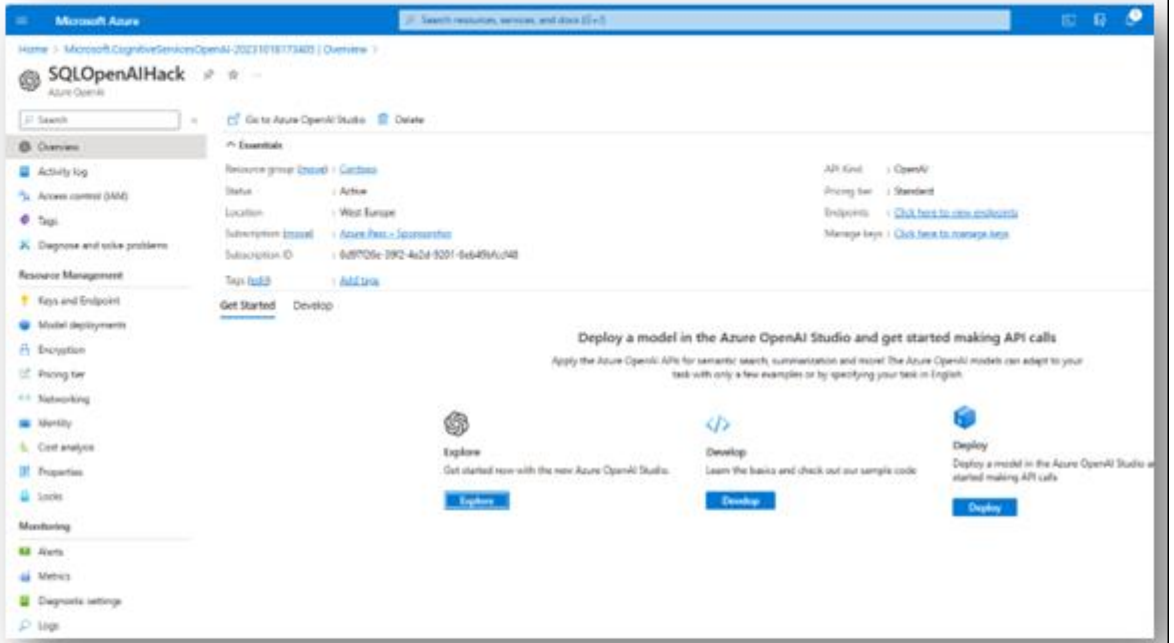
By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.

Basics

Subscription	Azure Pass - Sponsorship
Resource group	Contoso
Region	West Europe
Name	SQLOpenAIHack
Pricing tier	Standard S0

Network

Type	All networks, including the internet, can access this resource.



Home > Microsoft.CognitiveServicesOpenAI-20231018173405 | Overview >

SQLOpenAIHack

Azure OpenAI

Go to Azure OpenAI Studio | Delete

Essentials

Resource group (move)	Contoso
Status	Active
Location	West Europe
Subscription (move)	Azure Pass - Sponsorship
Subscription ID	6d97O6c-3992-4a2d-9201-6eb49bAcd48
Tags (edit)	Add tags
API Kind	OpenAI
Pricing tier	Standard
Endpoints	Click here to view endpoints
Manage keys	Click here to manage keys

Get Started | Develop

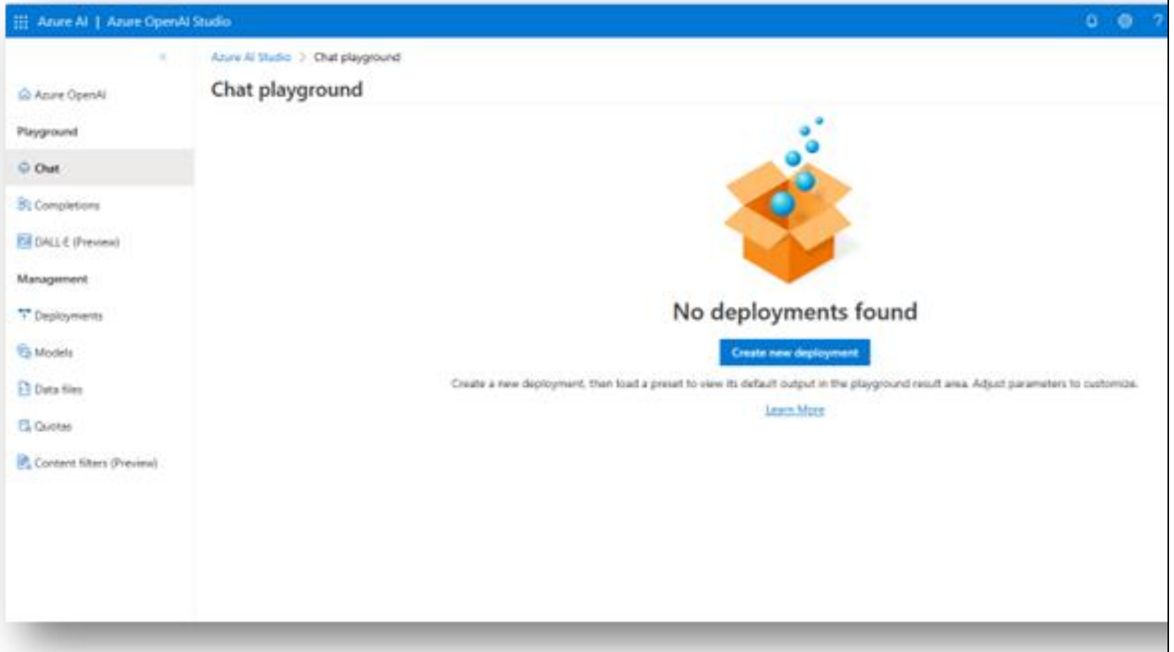
Deploy a model in the Azure OpenAI Studio and get started making API calls

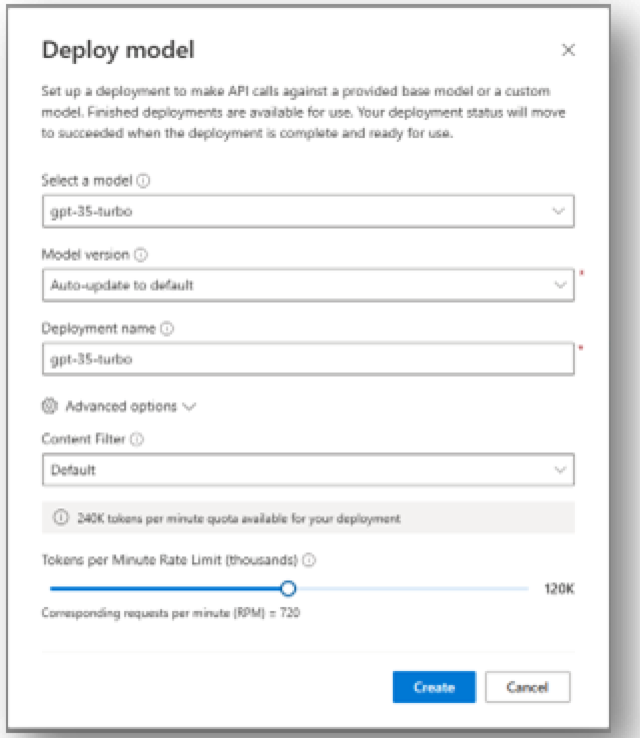
Apply the Azure OpenAI APIs for semantic search, summarization and more! The Azure OpenAI models can adapt to your task with only a few examples or by specifying your task in English.

Explore — Get started now with the new Azure OpenAI Studio.

Develop — Learn the basics and check out our sample code.

Deploy — Deploy a model in the Azure OpenAI Studio a… started making API calls

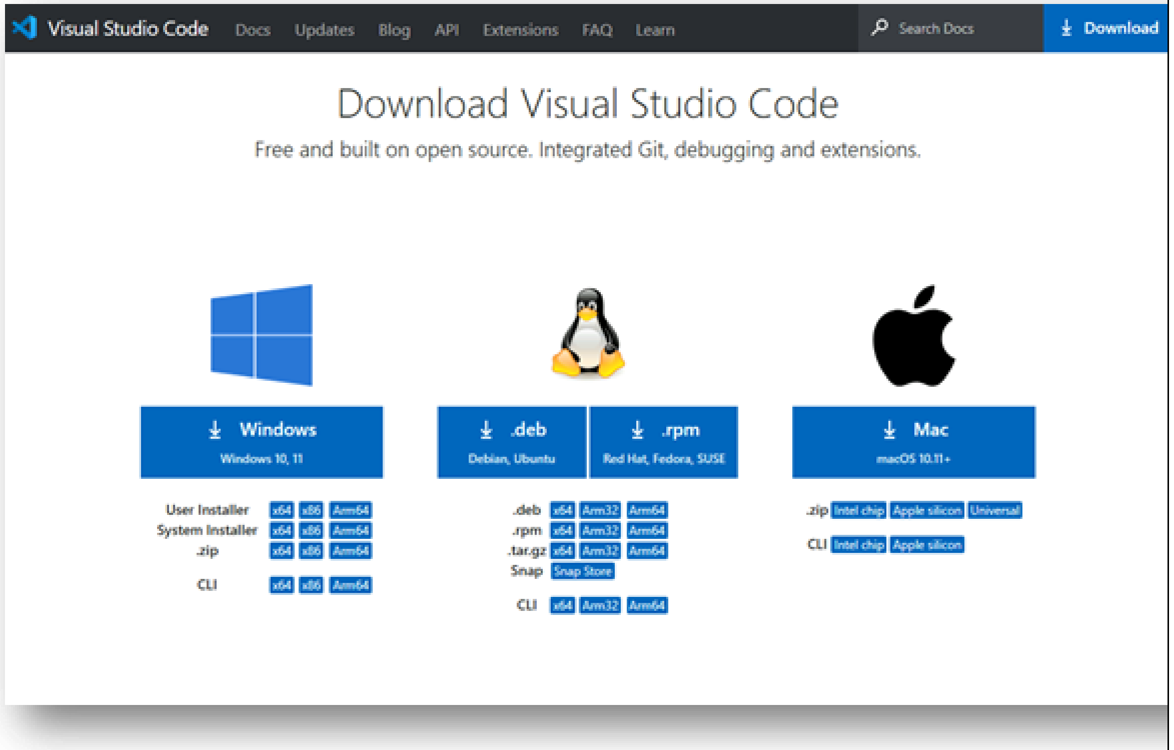
	 Azure AI | Azure OpenAI Studio Azure AI Studio > Chat playground Azure OpenAI Playground Chat Completions DALL-E (Preview) Management Deployments Models Data files Quotas Content filters (Preview) Create new deployment Learn More Chat playground No deployments found Create a new deployment, then load a preset to view its default output in the playground result area. Adjust parameters to customize.	

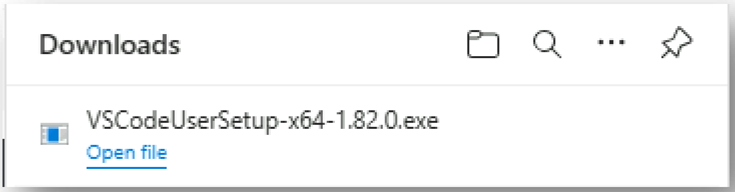
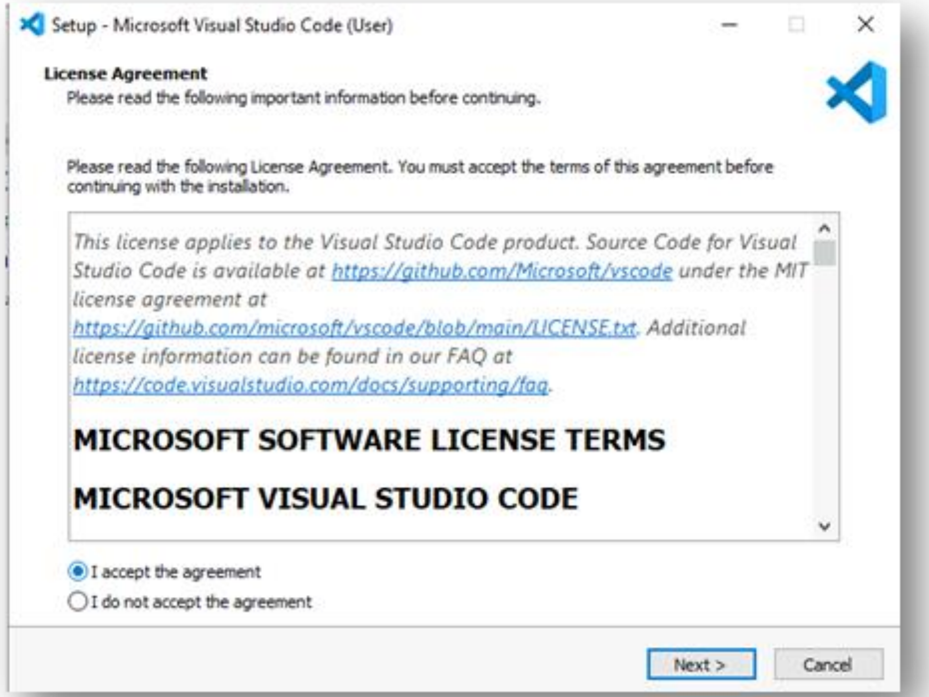
	 Deploy model Set up a deployment to make API calls against a provided base model or a custom model. Finished deployments are available for use. Your deployment status will move to succeeded when the deployment is complete and ready for use. Select a model: gpt-35-turbo Model version *: Auto-update to default Deployment name *: gpt-35-turbo Advanced options Content Filter: Default 240K tokens per minute quota available for your deployment Tokens per Minute Rate Limit (thousands): 120K Corresponding requests per minute (RPM) = 720 Create Cancel	

Microsoft

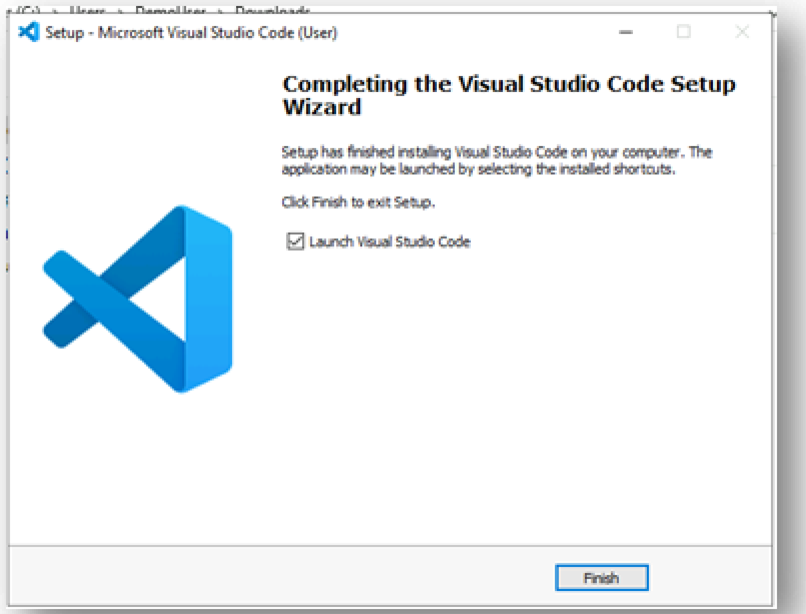
Environment Setup

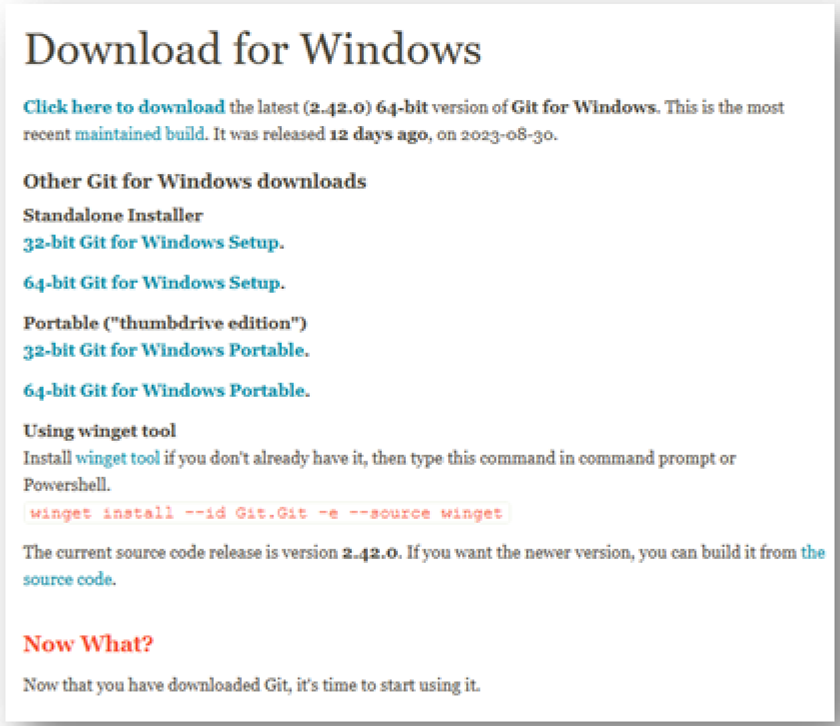
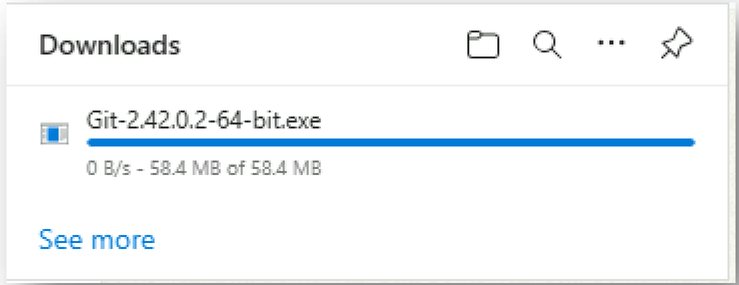
The environment has been provisioned for you upfront for this workshop. If you would like to manually set up the environment on your local machine, you can follow the instructions:

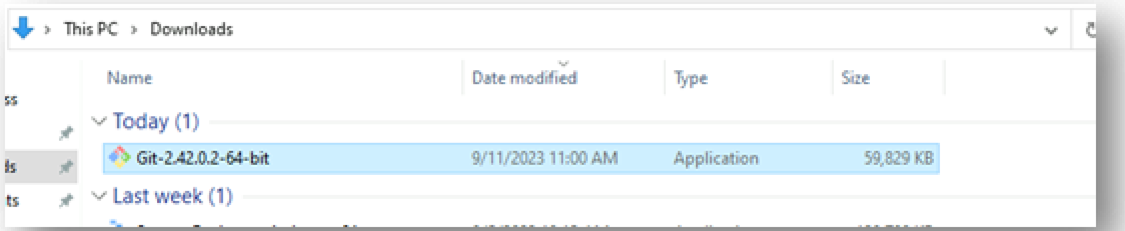
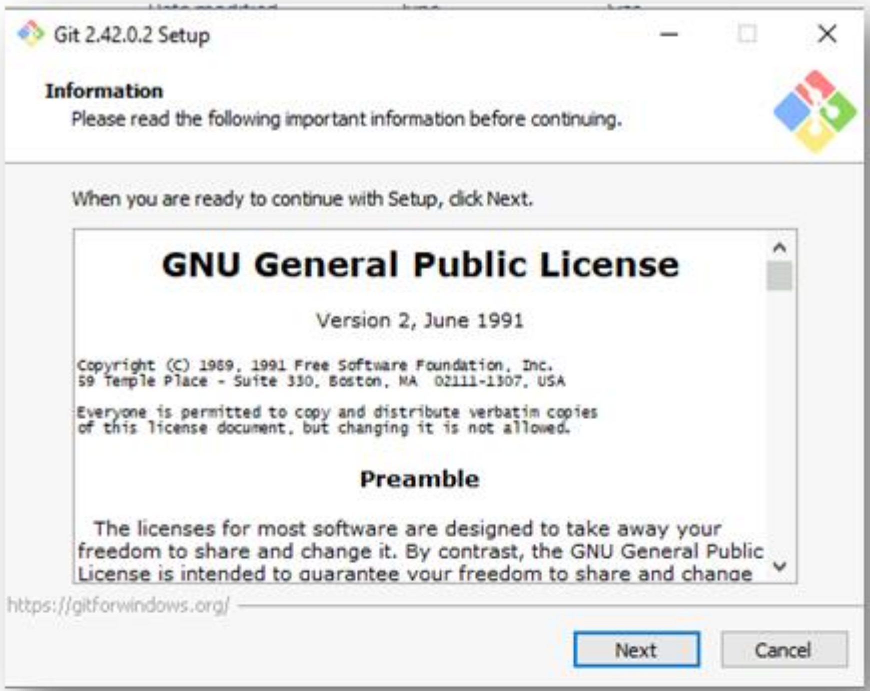
Narrative	Screenshot	Notes
Download Visual Studio Code https://code.visualstudio.com/download	 Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs Download Download Visual Studio Code Free and built on open source. Integrated Git, debugging and extensions. Windows Windows 10, 11 .deb Debian, Ubuntu .rpm Red Hat, Fedora, SUSE Mac macOS 10.11+ User Installer x64 x86 Arm64 System Installer x64 x86 Arm64 .zip x64 x86 Arm64 CLI x64 x86 Arm64 .deb x64 Arm32 Arm64 .rpm x64 Arm32 Arm64 .tar.gz x64 Arm32 Arm64 Snap Snap Store CLI x64 Arm32 Arm64 .zip Intel chip Apple silicon Universal CLI Intel chip Apple silicon	

	 Downloads VSCodeUserSetup-x64-1.82.0.exe Open file	
Install Visual Studio Code	 Setup - Microsoft Visual Studio Code (User) **License Agreement** Please read the following important information before continuing. Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation. *This license applies to the Visual Studio Code product. Source Code for Visual Studio Code is available at https://github.com/Microsoft/vscode under the MIT license agreement at https://github.com/microsoft/vscode/blob/main/LICENSE.txt. Additional license information can be found in our FAQ at https://code.visualstudio.com/docs/supporting/faq.* **MICROSOFT SOFTWARE LICENSE TERMS** **MICROSOFT VISUAL STUDIO CODE** [✓ I accept the agreement] I do not accept the agreement Next > Cancel	

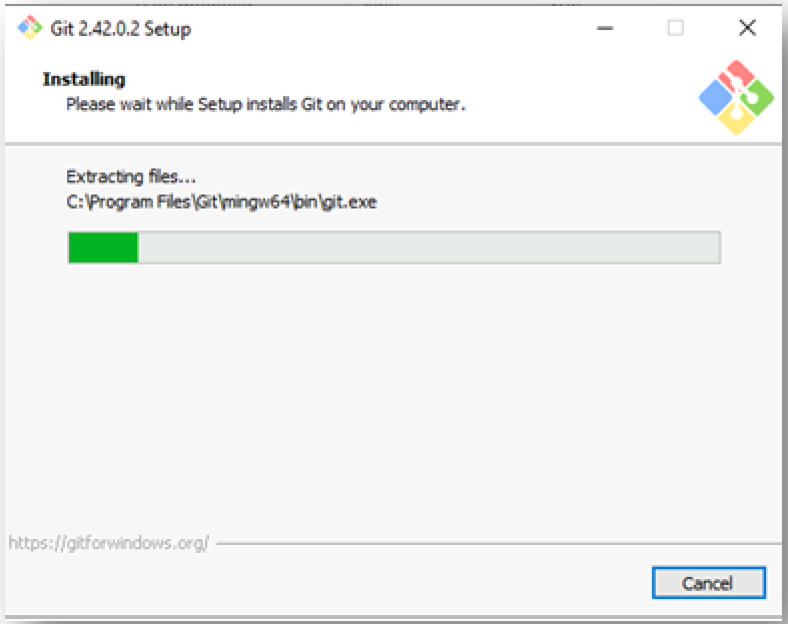
Microsoft

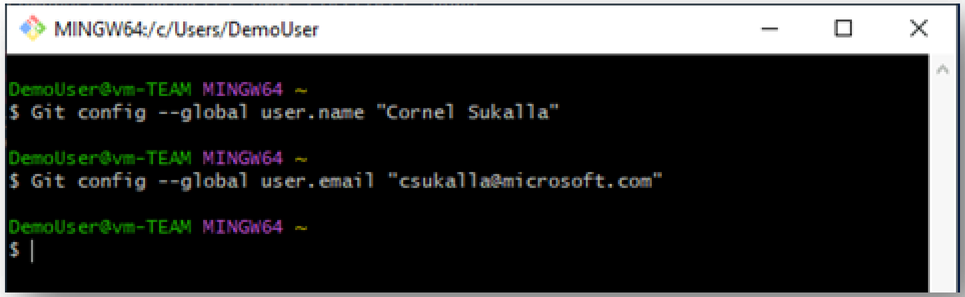
	 Setup - Microsoft Visual Studio Code (User) **Completing the Visual Studio Code Setup Wizard** Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts. Click Finish to exit Setup. ☑ Launch Visual Studio Code	

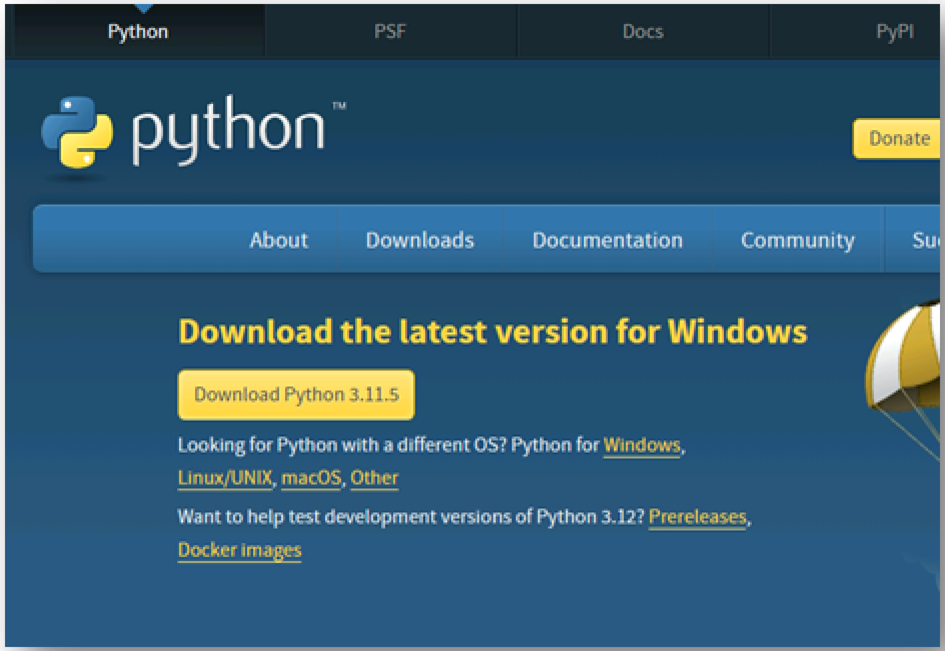
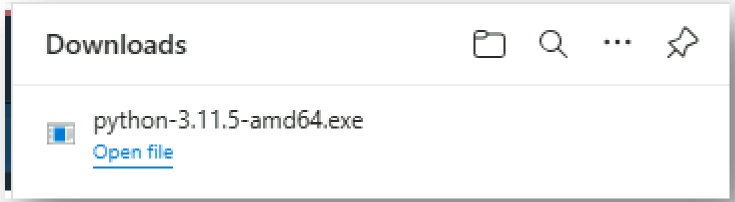
Download Git: https://git-scm.com/download/win	 Download for Windows Click here to download the latest (2.42.0) 64-bit version of Git for Windows. This is the most recent maintained build. It was released 12 days ago, on 2023-08-30. Other Git for Windows downloads Standalone Installer 32-bit Git for Windows Setup. 64-bit Git for Windows Setup. Portable ("thumbdrive edition") 32-bit Git for Windows Portable. 64-bit Git for Windows Portable. Using winget tool Install winget tool if you don't already have it, then type this command in command prompt or Powershell. `winget install --id Git.Git -e --source winget` The current source code release is version 2.42.0. If you want the newer version, you can build it from the source code. Now What? Now that you have downloaded Git, it's time to start using it.  Downloads Git-2.42.0.2-64-bit.exe 0 B/s - 58.4 MB of 58.4 MB See more	

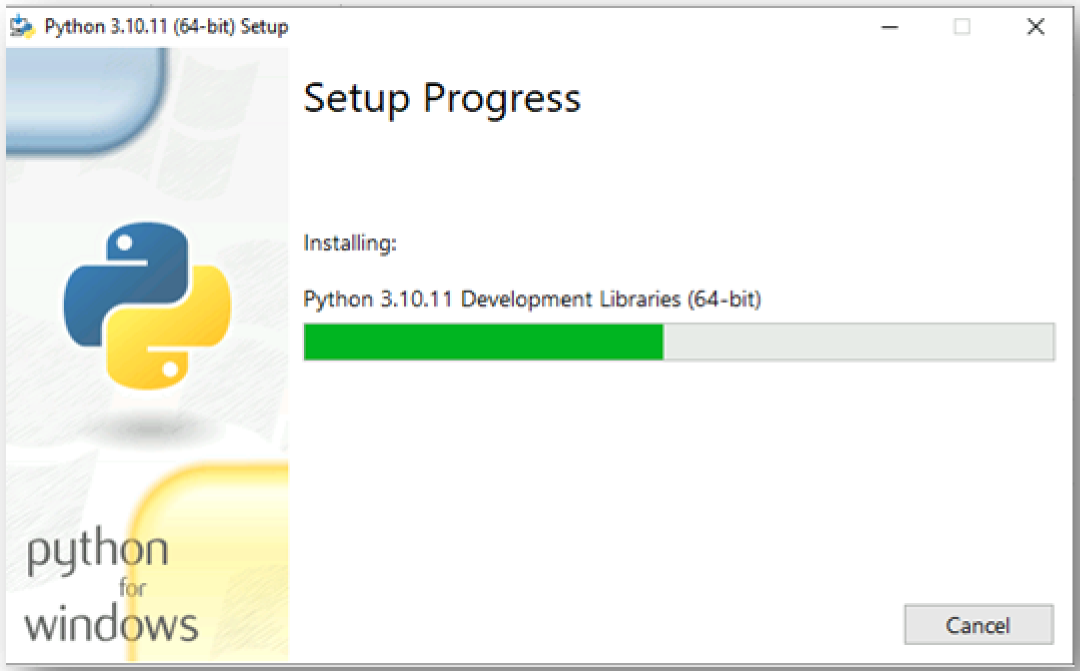
	 This PC › Downloads Name Date modified Type Size Today (1) Git-2.42.0.2-64-bit 9/11/2023 11:00 AM Application 59,829 KB Last week (1)	
Install GIT	 Git 2.42.0.2 Setup **Information** Please read the following important information before continuing. When you are ready to continue with Setup, click Next. **GNU General Public License** Version 2, June 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. **Preamble** The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change https://gitforwindows.org/ Next Cancel	

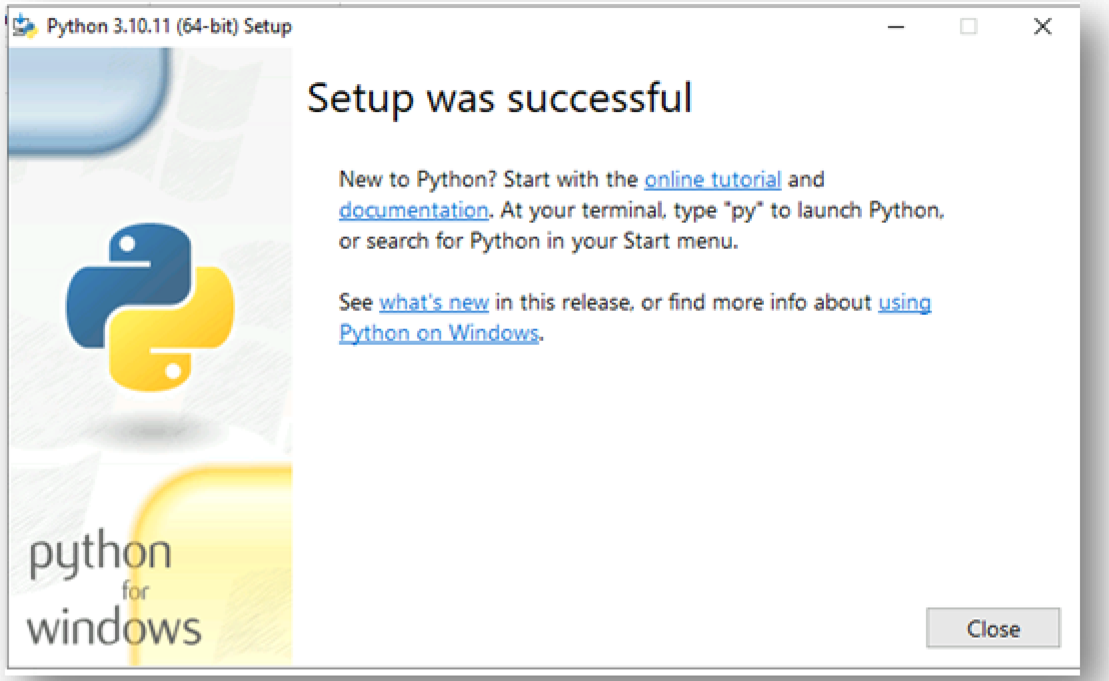
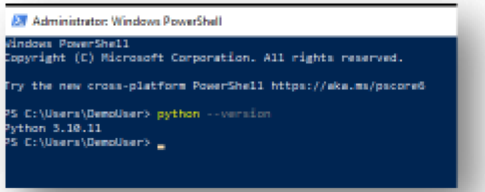
Microsoft

	 Git 2.42.0.2 Setup **Installing** Please wait while Setup installs Git on your computer. Extracting files... C:\Program Files\Git\mingw64\bin\git.exe https://gitforwindows.org/ Cancel	

(Optional) Set Git Variables	 MINGW64:/c/Users/DemoUser
	`DemoUser@vm-TEAM MINGW64 ~`
	`$ Git config --global user.name "Cornel Sukalla"`
	
	`DemoUser@vm-TEAM MINGW64 ~`
	`$ Git config --global user.email "csukalla@microsoft.com"`
	
	`DemoUser@vm-TEAM MINGW64 ~`
	`$ |`	

Create a python environment with **version from 3.7 and 3.10** **Download Python:** https://www.python.org/downloads/	 Python PSF Docs PyPI python™ Donate About Downloads Documentation Community Download Python 3.11.5 Download the latest version for Windows Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other Want to help test development versions of Python 3.12? Prereleases, Docker images  python-3.11.5-amd64.exe	

Installation Python **Important:** • Python and the pip package manager must be in the path in Windows for the setup scripts to work. • Ensure you can run python --version from console. On Ubuntu, you might need to run *sudo apt install python-is-python3* to link python to python3.	 Python 3.10.11 (64-bit) Setup Setup Progress Installing: Python 3.10.11 Development Libraries (64-bit) 	

Python 3.10.11 (64-bit) Setup

Setup was successful

New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See what's new in this release, or find more info about using Python on Windows.

Close

Administrator: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\DemoUser> python --version
Python 3.10.11
PS C:\Users\DemoUser> _
```