Module 2 - Get data insights using Azure OpenAI

# Environment Setup

**Prerequisites**

Basis <https://github.com/microsoft/OpenAIWorkshop/blob/main/scenarios/incubations/automating_analytics/README.md>

## Visual Studio

**Download Visual Studio Code**

<https://code.visualstudio.com/download>

A screenshot of a computer

Description automatically generated

A black text on a white background

Description automatically generated

**Install Visual Studio Code**

A screenshot of a computer screen

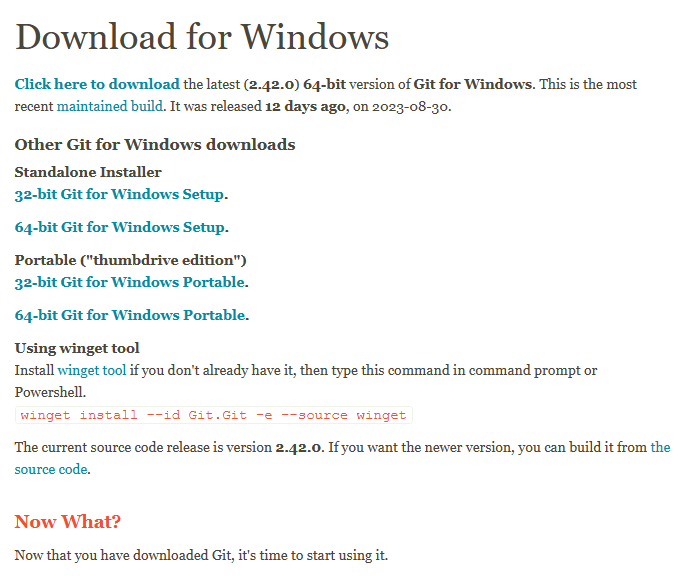
Description automatically generated

A screenshot of a computer program

Description automatically generated

## Git

**Download Git:** <https://git-scm.com/download/win>



A screenshot of a computer

Description automatically generatedA screenshot of a phone

Description automatically generated

**Install GIT**

A screenshot of a computer error

Description automatically generatedA screenshot of a computer program

Description automatically generated

## Azure Open AI setup

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.

See documentation for instructions: [How-to: Create and deploy an Azure OpenAI Service resource - Azure OpenAI | Microsoft Learn](https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/create-resource?pivots=web-portal)

## Install Application locally

**Install the application locally**

Clone the repo (e.g. *git clone https://github.com/microsoft/OpenAIWorkshop.git* or download). Then navigate to: *cd scenarios/incubations/automating\_analytics*

A screenshot of a computer program

Description automatically generated

**Set Git Variables**

A screenshot of a computer screen

Description automatically generated

(Optional) Provide settings for Open AI and Database.You can either create a secrets.env file in the root of this folder (scenarios/incubations/automating\_analytics) as below or do it using the app's UI later on.

**Option 1:** use built-in SQLITE. Then you don't need to install SQL Server.

AZURE\_OPENAI\_API\_KEY="9999999999999999999999999"

AZURE\_OPENAI\_GPT4\_DEPLOYMENT="NAME\_OF\_GPT\_4\_DEPLOYMENT"

AZURE\_OPENAI\_CHATGPT\_DEPLOYMENT="NAME\_OF\_CHATGPT\_4\_DEPLOYMENT"

AZURE\_OPENAI\_ENDPOINT=https://openairesourcename.openai.azure.com/

SQL\_ENGINE = "sqlite"

**Option 2:** use your own SQL Server

AZURE\_OPENAI\_API\_KEY="9999999999999999999999999"

AZURE\_OPENAI\_ENDPOINT="https://openairesourcename.openai.azure.com/"

AZURE\_OPENAI\_GPT4\_DEPLOYMENT="NAME\_OF\_GPT\_4\_DEPLOYMENT"

AZURE\_OPENAI\_CHATGPT\_DEPLOYMENT="NAME\_OF\_CHATGPT\_4\_DEPLOYMENT"

SQL\_USER="sqluserid"

SQL\_PASSWORD="sqlpassword"

SQL\_DATABASE="WideWorldImportersDW"

SQL\_SERVER="sqlservername.database.windows.net"

Navigate to cd scenarios/incubations/automating\_analytics

## Python environment

Create a python environment with **version from 3.7 and 3.10**

**Download Python:** <https://www.python.org/downloads/>

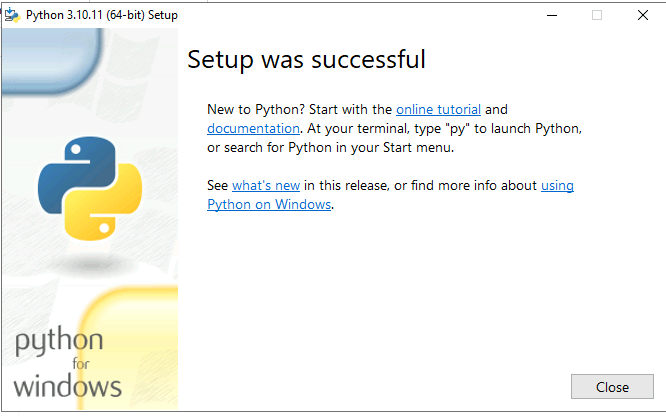


A screenshot of a computer

Description automatically generated

**Installation Python**A screenshot of a computer

Description automatically generated



**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.

A screenshot of a computer program

Description automatically generated

Import the requirements.txt: *pip install -r requirements.txt*

Cd\

A screenshot of a computer program

Description automatically generated

A screen shot of a computer screen

Description automatically generated

Place the **secrets.env** in the path (this should be replaced with details from Hack subscriptions)

A screenshot of a computer code

Description automatically generated

AZURE\_OPENAI\_API\_KEY="4be92cb8a2a444eb806a0215436d1489"

AZURE\_OPENAI\_ENDPOINT="https://oaibpa2txjd3xq5sbd2.openai.azure.com/"

AZURE\_OPENAI\_GPT4\_DEPLOYMENT="chatgpt"

AZURE\_OPENAI\_CHATGPT\_DEPLOYMENT="chatgpt"

SQL\_USER="DemoUser"

SQL\_PASSWORD="Demo@pass1234567"

SQL\_DATABASE="AdventureWorks2019"

SQL\_SERVER="sqlhackmi-vxrtdxsgia4gc.public.12be5ca35652.database.windows.net,3342"

C:\\_OpenAI-SQL\_\OpenAIWorkshop\scenarios\incubations\automating\_analytics

A screenshot of a computer

Description automatically generated

Open Visual Studio code and import the project

A screenshot of a computer

Description automatically generated

Navigate to path C:\\_OpenAI-SQL\_\OpenAIWorkshop

A screenshot of a computer

Description automatically generated

Navigate to app.py

A screenshot of a computer

Description automatically generated

Open Terminal Window

A screenshot of a computer program

Description automatically generated

Navigate to path: .\scenarios\incubations\automating\_analytics\

A screenshot of a computer

Description automatically generated

# Test application & assistants

To run the application from the command line: *streamlit run app.py*

py -m streamlit run app.py

Edge Browser opens with setting retrieved from secrets:

A screenshot of a computer

Description automatically generated

Sample: SQL Query writing Assistant

A screenshot of a computer

Description automatically generated

Sample Data Analysis Assistant

A screenshot of a computer

Description automatically generated

# Resolve Data quality issues

Storyline:

**1) Rename tables / columns to not self-explanatory name**

e.g. rename:

* customer to cst
* Address to Addr
* Employee to Emp
* ProductCategory to ProdCat
* ProductSubcategory to ProdSubCat
* SalesOrderDetail to SOD

**2) Run natural language query and hit some errors**

<enter samples here >

* Show me top 10 customers
* List all product categories
* Get a list of all employees and their contact information
* Sort products by price

**3) Create database synonyms / views to resolve the problems**

You can work with so-called synonyms for table names which Azure OpenAI can work with, without having to change the actual table names. Other applications could rely on the actual table names.

See documentation for more information: [Synonyms (Database Engine) - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/relational-databases/synonyms/synonyms-database-engine?view=sql-server-ver16)

IN SSMS, connect to your Azure MI and navigate to the AdventureWorks2019 database. You can either create synonyms via T-SQL or via the GUI (right-click on synonyms).

Example T-SQL statement:

CREATE SYNONYM [Sales].[Customer] FOR [AdventureWorks2019].[Sales].[CST]

You can check the created synonyms:

A screenshot of a computer

Description automatically generated

For the columns, you can create views.

**4) Rerun the queries in natural language**