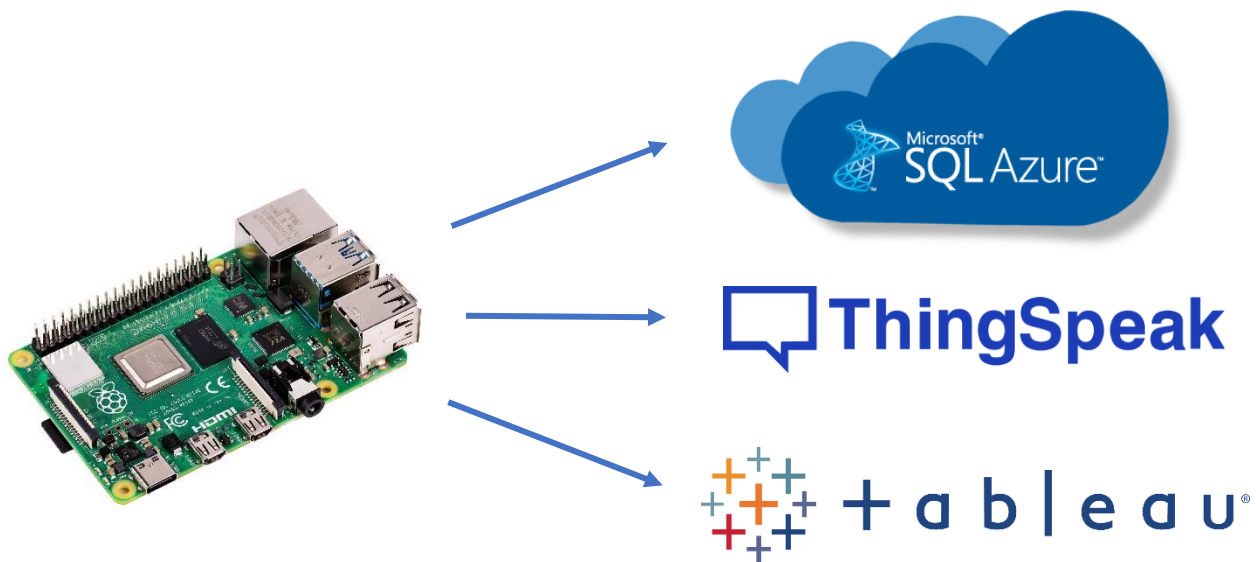
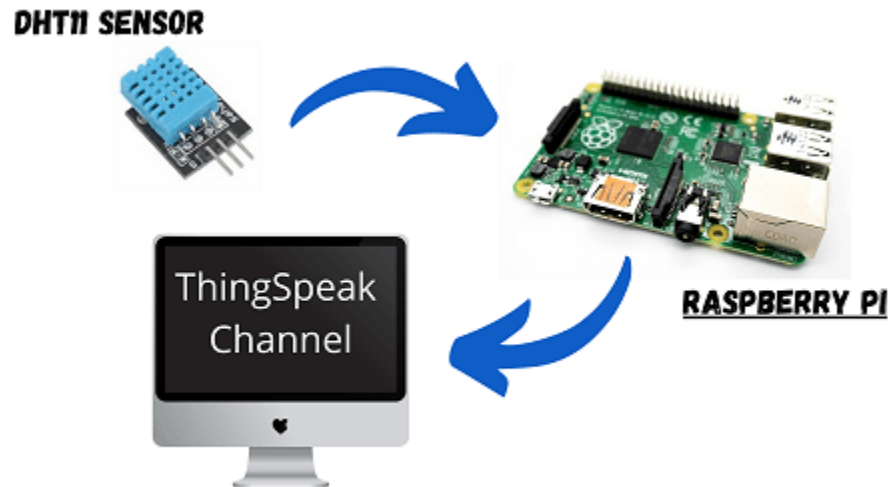

Raspberry PI connected to Azure database, Thingspeak and Tableau Software

Internet of Things, Web & Mobile



TP 08: Save the data from the Raspberry Pi sensor in a database and then create a dashboard to make it an insight of your data.

After doing the TP7, which have been able to capture temperature and humidity from a humidity Sensor and send it to Thingspeak in real-time.



Now I will proceed to the next step with store all collecting data local or cloud using a Microsoft SQL Server.

Step 1:

There are several items that you may retrieve before you connect Python to MS SQL Server (Portal Azure <https://portal.azure.com/>), including the:

- Server name
- Database name
- Table name
- User Id
- Password

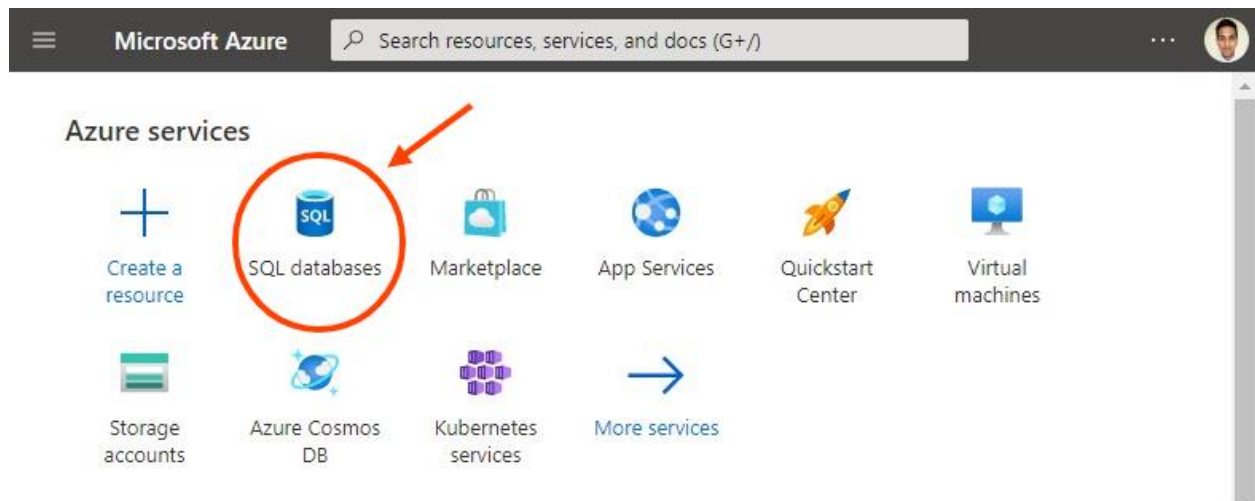


Fig 1. Select option SQL Database

Let create a database using Microsoft SQL Server Management Studio 18.

- Server name: **iot-hramphul.database.windows.net**
- Database name: **IoT_UDM_Database**
- Table name: **HumitureSensor**
- User Id: **hramphul** / Password: **{your_password}**

Microsoft Azure

Search resources, services, and docs (G+)

Home >

SQL databases

CEGID (cegidgroup.onmicrosoft.com)

+ Create ⌚ Reservations ⚙️ Manage view ↻ Refresh ⬇️ Export to CSV 🔗 Open query ⋮

Filter for any field... Subscription equals all Add filter More (2)

No grouping List view

Name ↑↓ Server ↑↓ Replica type ↑↓ Pricing tier ↑↓ Location ↑↓ Sul

No SQL databases to display

Try changing or clearing your filters.

Create SQL database

Learn more ↗️

Give feedback

Fig 2. Create SQL Database

Microsoft Azure

Search resources, services, and docs (G+)

Home > iot-hramphul

iot-hramphul | SQL databases

SQL server

Search

Search to filter databases...

Database	↑↓	Status
IoT_UDM_Database		Online

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start

Settings

Azure Active Directory
SQL databases
SQL elastic pools

Fig 3. After create database IoT_UDM_Database

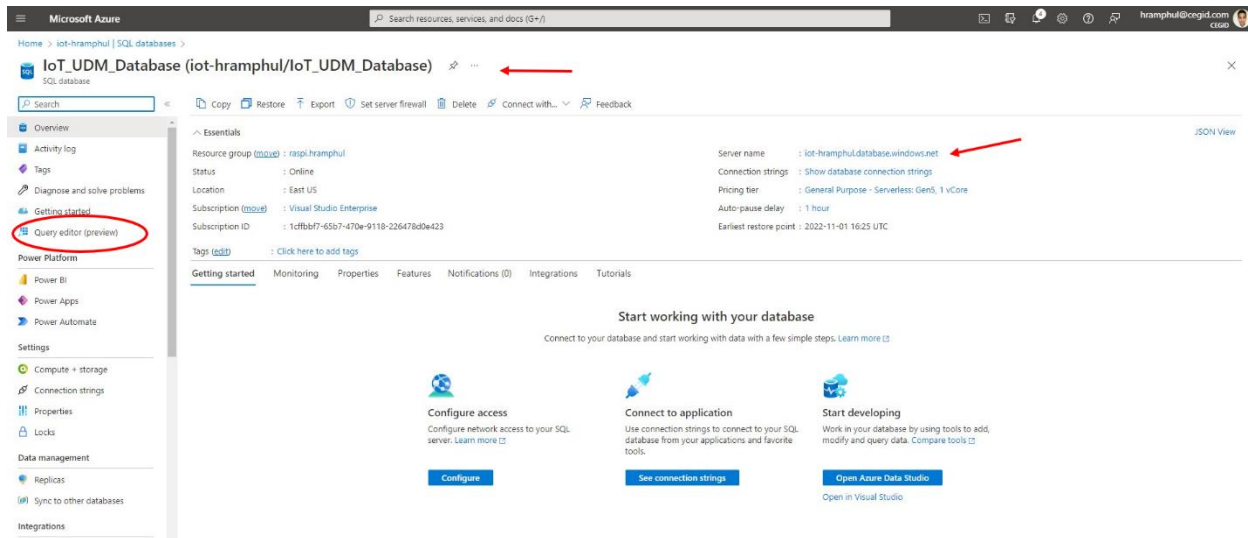


Fig 4. Connect to database IoT_UDM_Database

Structure of table ***HumitureSensor*** in script version.

```
CREATE TABLE [dbo].[HumitureSensor](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Temperature] [int] NULL,
    [Humidity] [int] NULL
) ON [PRIMARY]
GO
```

Step 2:

To insert values into MS SQL Server table using Python, we must use module **pyodbc**. To use it, we must install it first by using the command below:

```
pip3 install pyodbc
```

Brief note:

Pyodbc is an open source Python module that makes accessing ODBC databases simple. It implements the DB API 2.0 specification but is packed with even more Pythonic convenience.

Step 3:

After **pyodbc** installed successfull. Now let's insert our collecting data into the *HumitureSensor* table.

Here is the complete Python code how to insert those collecting data:

```
# Define a function that save data to table
# @Temperature as parameter: _temperature
# @Humidity as parameter: _humidity
def saveData(_temperature, _humidity):
    # Map to ODBC
    cursor = DB_CONNECTION.cursor()
    # Query to execute
    cursor.execute('INSERT INTO HumitureSensor (Temperature, Humidity) VALUES
    (?, ?)', (_temperature, _humidity))
    # Execute my request
    DB_CONNECTION.commit()
```

Download source code from (Github) 100% executable and each line is explained.

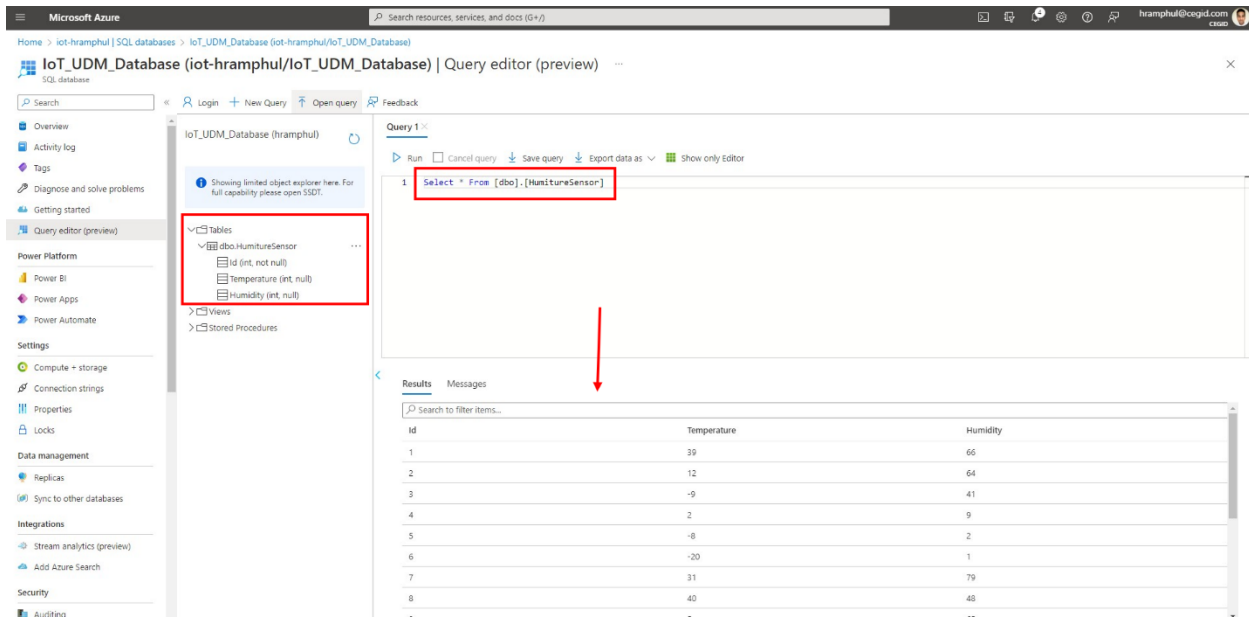
Important: Don't forget to add **conn.commit()** to ensure that the insert command would get executed.

Step 4:

Finally, we can verify that all data were inserted into the *HumitureSensor* table by running the following SELECT query in MS SQL Server:

SELECT * FROM product

You should now see all additional data at the bottom of the table:



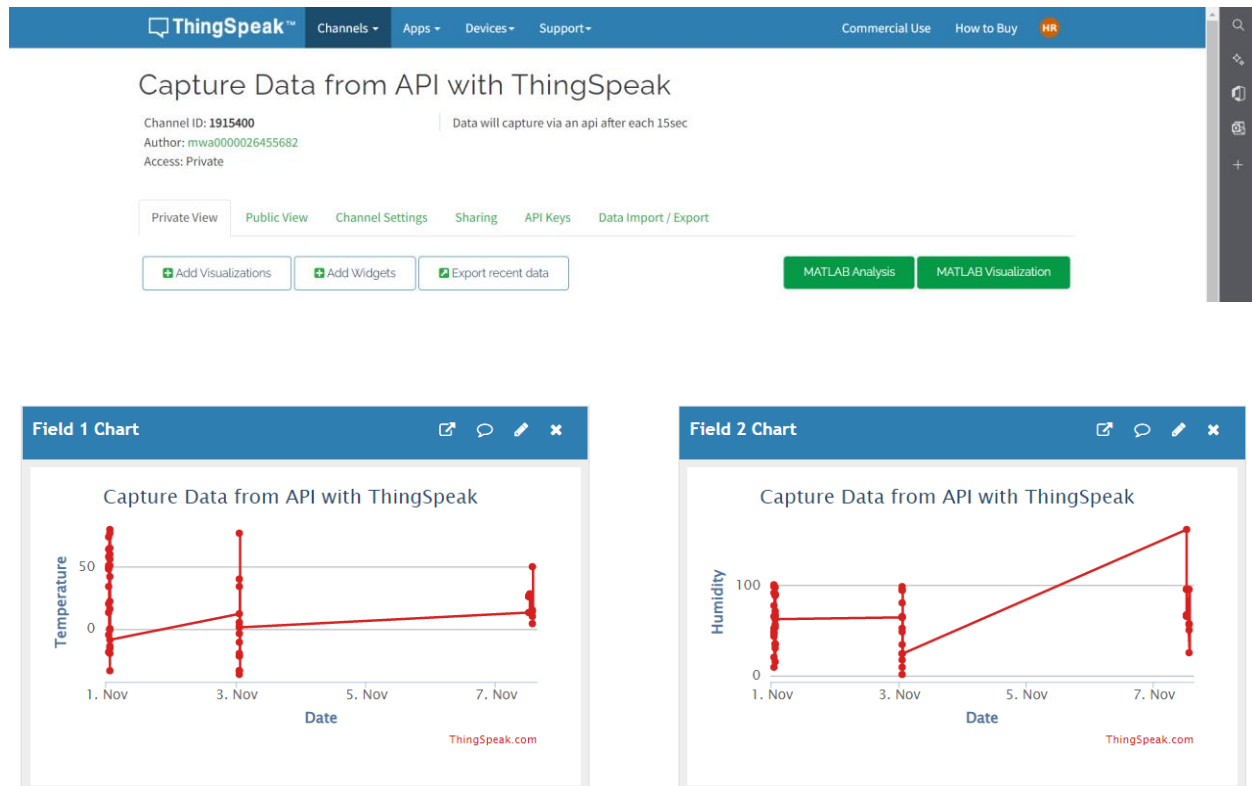
The screenshot shows the Microsoft Azure portal interface for the IoT_UDM_Database (iot-hramphul/IoT_UDM_Database). The Query editor (preview) is open, displaying the query: `Select * From [dbo].[HumitureSensor]`. The results table shows 8 rows of data:

Id	Temperature	Humidity
1	39	66
2	12	64
3	-9	41
4	2	9
5	-8	2
6	-20	1
7	31	79
8	40	48

Fig 5. All data collected save to cloud database.

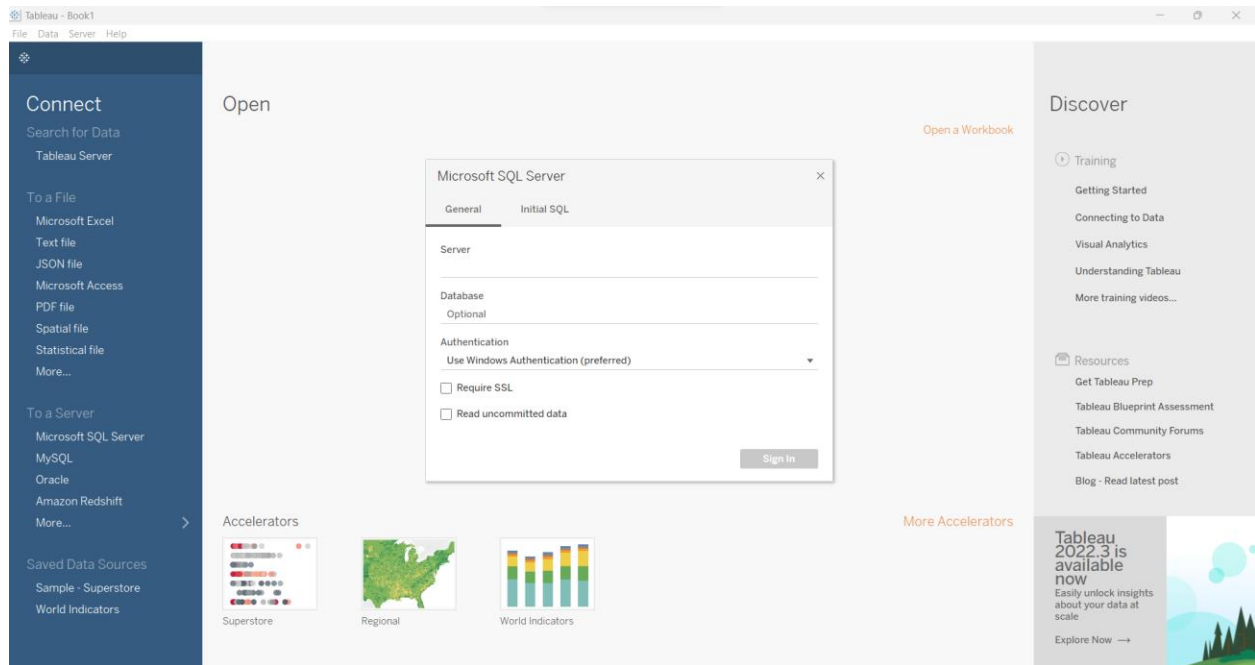
Step 5:

In Thingspeak dashboard you can see the graph and the readings.

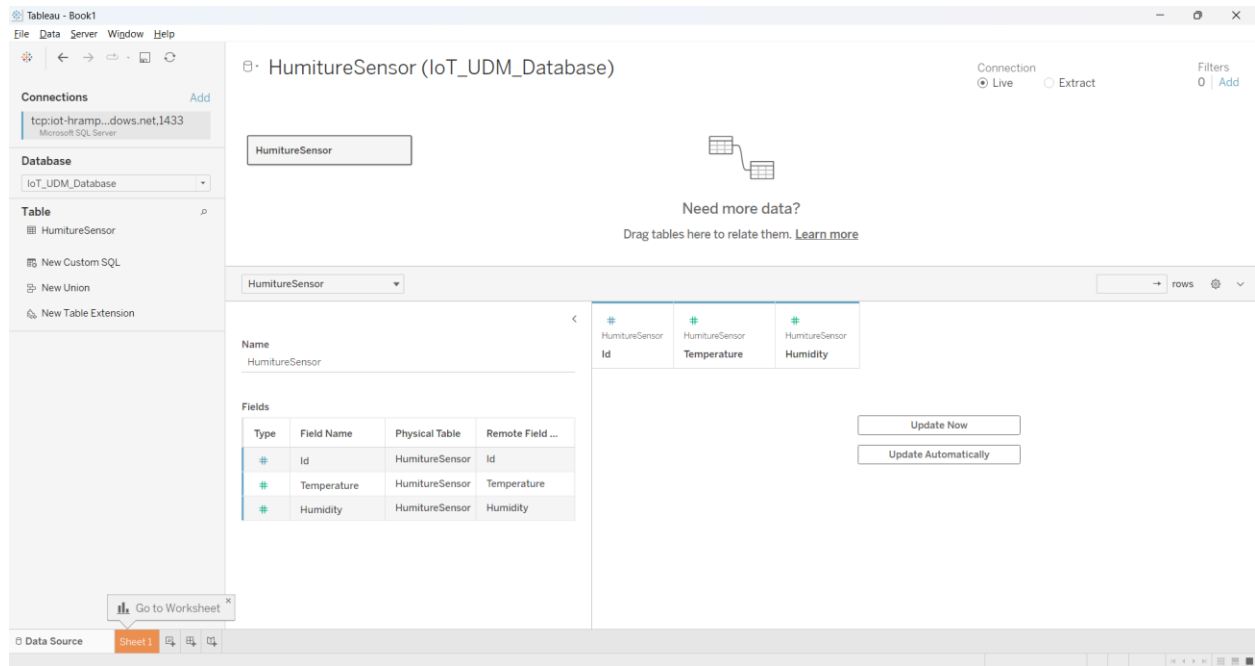


Step 6:

Now let's connect our data to Tableau to have a readable graph. First connect the Tableau to the MS SQL Server.



Connected successfully to MS SQL Server via Tableau. And can see live data entered after each 15 second.



Step 7:

Generate a live readable graph with Tableau by connection to the MS SQL Server.

