

Objective: In this tutorial, you will learn how to re-build the database with Azure and visualize it using PowerBI.

Required Setup: Azure set-up

Parts:

- RPi 3 B
- GrovePi+ board
- Two Grove connection wires
- 1 x Grove Temperature and Humidity Sensor
- 1 x Grove LCD RGB Backlight v2.0

Part A: Set up a SQL database on Azure

Step 1. Sign into the Azure portal using <https://portal.azure.com/#home>. You should see your dashboard.

Step 2. Click on **'+Create a resource'** in the main navigation menu. In the search box, type **'SQL database'**, find the service and click on **'Create'** at the bottom.

Step 3. For 'Database name', you have to pick a name that is globally unique (mine is testdatabase); for 'Server', click on **'Create new'** and a side panel like the image on the right will prompt from the right. Pick your own 'Server name', 'Server admin login', and 'Password'. Click on the **'Select'** at the bottom of the side panel. Leave other fields as they were and click on **'Review+Create'** at the bottom. Click on **'Create'**.

Create SQL Database
Microsoft

⚠ Changing basic options may reset selections you have made. Please review all options prior to creating the database.

Basics • Additional settings • Tags • Review + Create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription Free Trial

* Resource group test-group
[Create new](#)

DATABASE DETAILS

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

* Database name testdatabase

* Server
The value should not be empty.

* Want to use SQL elastic pool? ☐ Yes ☒ No

* Compute + storage Please select a server first.
[Configure database](#)

New server

* Server name raspberrypitest
.database.windows.net

* Server admin login testrole

* Password

* Confirm password

* Location West US

☒ Allow Azure services to access server

Step 4. Click on '**go to resource**', you will now see the main dashboard for Azure SQL database.

Step 5. In the left navigation menu for the database, find and choose '**Query editor(preview)**'. Put in the password you just used to create the SQL database. Then click '**OK**'.

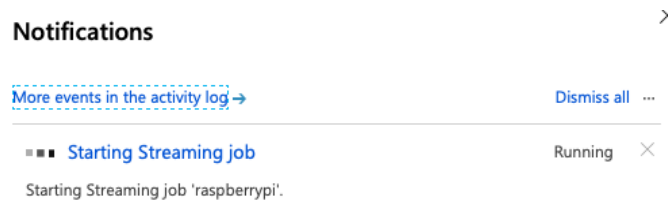
Step 6. You will now see a blank section called '**Query 1**'. Copy paste the following SQL commands into it.

```
CREATE TABLE Temp_HUM_Sound(
Event_data Datetime NULL,
temp float NULL,
humandity float NULL,
sound float NULL);
```

Step 7. Click '**Run**'.

Step 8. Click on the '**Tables**' from the list on the left to check if the table is created successfully.

Step 12. Back to Azure Stream Analytics job. Click '**Overview**' from the navigation menu, then click '**Start**' and '**Start**'. It should take a while. When it finishes, a notification will prompt.



Step 13. Let it run for several minutes. Now you may wonder how to check the data generated in the table. Go back to the SQL database tab (if you closed it, click on '**Home**' from the main navigation menu, you will find the SQL database from the '**recent resource**' section). In the left navigation menu for the database, find and choose '**Query editor(preview)**'. Put in the password to login the SQL database. Then click '**OK**'.

Step 14. You will see a blank section called '**Query 1**'. Type the following SQL command in the query window, then click '**Run**'.

```
SELECT * FROM Temp_Hum_Sound;
```

Step 15. If everything works, you should see the following in the '**Results**' section below the query window. If nothing shows up, go back to Raspberry Pi, use 'ctrl+c' (control + c on Mac) to terminate the script then run it again. After that, Stop the Stream Analytics job and restart it again.

| Results Messages | | | |
|-----------------------------|------|----------|-------|
| Search to filter items... | | | |
| EVENT_DATE | TEMP | HUMIDITY | SOUND |
| 2019-03-01T05:54:15.0470000 | 24 | | |
| 2019-03-01T05:54:09.5470000 | 24 | | |
| 2019-03-01T05:54:20.5730000 | 24 | | |
| 2019-03-01T05:54:26.0530000 | 24 | | |
| 2019-03-01T05:54:31.5700000 | 24 | | |
| 2019-03-01T05:54:37.0200000 | 24 | | |

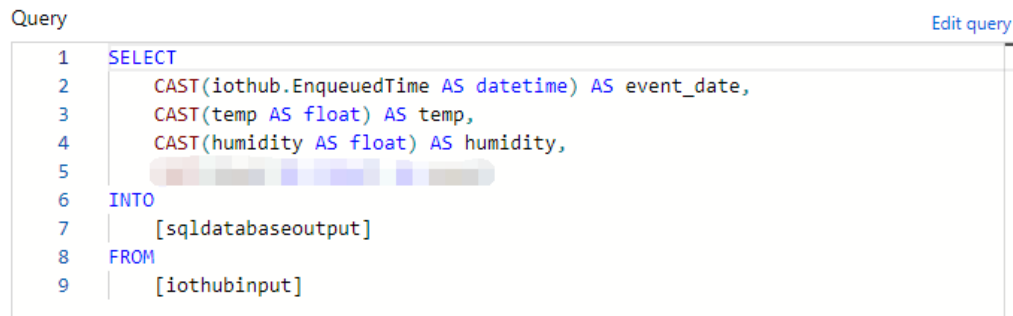
Step 16. Since we did not have the code for Humidity and the sensor for sound, they won't show up in the table. Let's try to add them! Now go back to the python script 'stream.py' on your Raspberry Pi. Try to finish **Task 2**. You only need to modify the code on line 80 to send humidity and sound sensor data. Use the link on line 78 as a reference to study JSON format, and make the script send both humidity and sound data to Azure SQL database.

```
##### Task 2 -- Modify the message
## Now the message only contains information of temperature
## Can you include humidity within the message as well?
## The message is in JSON format
## Read more information from this link: https://www.w3schools.com/js/js_json_syntax.asp
## Use "Humidity" as the name of data
## BONUS: If you want to add sound data, use 'Sound' as the name of data
message = { "temp": str(temp) }
```

Hints: 1. Connect the sound sensor to Port A2

2. Modify the **code** at line 80

3. Modify the Query in Stream Analytics job on Azure, make sure you map the name right. Ex. Humidity and sound. Make them all floating value. The following image is an example for adding humidity.



The screenshot shows the Azure Stream Analytics Query Editor interface. The title bar says 'Query' and there is an 'Edit query' link in the top right. The query text is as follows:

```
1 SELECT
2     CAST(iothub.EnqueueedTime AS datetime) AS event_date,
3     CAST(temp AS float) AS temp,
4     CAST(humidity AS float) AS humidity,
5
6 INTO
7     [sqldatabaseoutput]
8 FROM
9     [iothubinput]
```

4. Make sure you test your query before you save it. If you forget how to test it, go back to handout 2 and check **Step 12, Step 13 Step 14.**

5. There might be a chance that your database did not receive any data from the IoT hub. To solve that, delete the previous database in Azure SQL database by typing:

```
DROP TABLE Temp_Hum_Sound;
```

in the query window of Query editor. Use the SQL commands in **Step 6** to re-create the database.

6. Keep the python script running before you start the Stream Analytics.

7. Let it run for several minutes. Use SQL command in **Step 14** to check the database. If you experience any issues, use 'ctrl+c' to terminate the python script and stop the Stream Analytics job. Then reboot the Raspberry Pi, use terminal to run the script again, then re-create the database and restart the Stream Analytics job again.

Congratulations! You have successfully to re-build the database with Azure and visualize it using PowerBI.

References:

1. <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi>
2. <https://gunnarpeipman.com/iot/beer-iot-stream-analytics-sql/>