**Question 1: Setting up Sensors (15 points)**

You are given two sensors

Write a Python script that:

* Collects data from both sensors every 5 seconds.
* Prints the data in the terminal using the following format:

Temperature: XX.X°C, Humidity: YY.Y%

Proof: screenshot of your terminal

A screen shot of a computer

Description automatically generated

**Question 2: Connecting to AWS IoT Core (15 points)**

Using the Python AWSIoTPythonSDK library, write a script that:

* Connects the Raspberry Pi to AWS IoT Core using your device's Thing certificate and private key.
* Publishes the temperature and humidity data from Question 1 to the topic champlain/sensor/102/data in JSON format, like this:

{

"temperature": XX.X,

"humidity": YY.Y

}

Proof: Screenshot of the AWS MQTT client with messages arriving to the required topic and with the proper format (json)

A screenshot of a computer

Description automatically generated

**Question 3: Storing Data in DynamoDB (15 points)**

You need to store the data from the two sensors in a DynamoDB table using AWS IoT Rules

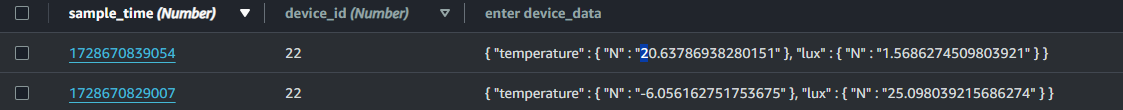
Write an AWS IoT Rule that:

* Triggers when data is published to the topic.
* Inserts the sensors values and timestamp into the DynamoDB table.
* Include the SQL query used to select the data from the incoming message.

Proof: your SQL statement and data inserted in the Dynamo table

A black rectangular object with a white line

Description automatically generated



A screenshot of a computer

Description automatically generated

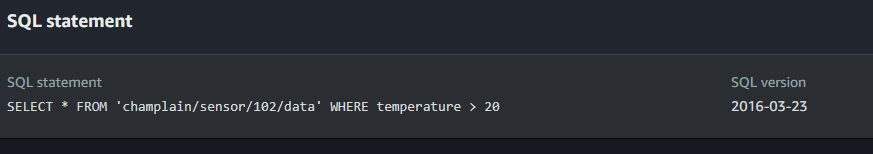
**Question 4: Republish Rule for threshold value (15 Marks)**

Create a separate AWS IoT Rule that:

* Monitor one of your sensors.
* Republishes the data to the topic champlain/republish if the threshold exceeds a given value.
* Write the SQL query for this rule and explain how you would configure it in the AWS Console.

I would set the action to Republish a message to an IoT topic and lab to LabRole and then create the rule. From there, I I would subscribe to the champlain/republish topic in the MQTT test client and then run my code. If everything below 20 temp is excluded from being published in this topic, then it works.

**Proof: Your SQL query and the messages arriving to the RPi only if the threshold exceeded**



A screenshot of a computer

Description automatically generated

**Question 5: Subscribing to the /republish Topic (15 points)**

Write a Python script for the Raspberry Pi that:

* Subscribes to the champlain/republish topic.
* When a message arrives (temperature > 30°C), the Raspberry Pi turns on an LED connected to a GPIO pin.
* Prints the received message in the terminal

Proof the received message in the terminal and the demo with the LED

A computer screen shot of a black screen

Description automatically generated

At the end of the exam you should demo your result with the teacher and submit the link to your repository with the code.