MSc in Business Analytics – DMBI
Assignment #3
Due Date: 07 Jan 2018

INSTRUCTIONS

You are going to use Azure Stream Analytics to process a data stream of vehicle observations and answer stream queries.

1. Create a trial account at: https://azure.microsoft.com/en-us/
2. Setup an Event Hub.
3. Generate a Security Access Signature: https://github.com/sandrinodimattia/RedDog/releases
4. Edit Generator.html (open with a text editor eg: Sublime or Notepad++) and update the CONFIG variables. **Keep the "js" folder in the same folder as the Generator.html file.**
5. Feed the Event Hub with the use of Generator.html (In order to start the Stream Generator, open the Generator.html with a web browser (eg: Chrome) and press the "Send Data" button.)
6. Setup a Storage account.
7. Upload the Reference Data files to your storage account. Make sure that you transform the data in a format that can be used as a data source.
8. Setup a Stream Analytics Job.
9. Use the Event Hub + Reference Data Files as Input.
10. Create a Blob Storage Output.

SCENARIO

You have access to a data stream that's generated from sensors placed in some checkpoints (toll stations and speed cameras). Each time a car passes by one checkpoint, an event is generated. All cars are equipped with tags that provide the vehicleTypeID and colorID of each car. Tag readers are capable of reading this information. In addition, a camera reads the license plate and completed the event's data. You are asked to create an Azure Analytics solution for the tasks listed in the "QUERIES" section.

REFERENCE DATA

GENERAL NOTES

- Use only the parts of the datasets that you need for the queries.
- Use Sublime or Notepad++ to have a proper view of the datasets.
- Use any type of software you like to transform the reference datasets (or write your own scripts). **HINT: Use colors.json as a reference.**

CAR_DATA.csv

- Information about the Make, Model and Model Year of all the cars.
- It's a comma separated csv file.
- The "id" of this dataset can be joined with the input's "vehicleTypeID" section.

COLORS.json

- Information about the color of each car.
- It's a json file.
- The "color_code" of this dataset can be joined with the input's "colorID" section.

SPEED_CAMERA_SPOTS.csv

- Information about each speed camera that generates events.
- It's a comma delimited csv file.
- The "id" of this dataset can be joined with the input's "checkpointID" section.

TOLL_STATIONS.txt

- Information about each toll station that generates events.
- It's a tab delimited txt file
- The "id" of this dataset can be joined with the input's "checkpointID" section.

WANTED_CARS.txt

- Information about the cars of police's most wanted criminals.
- A list of license plates of cars that belong to criminals.
- This dataset can be joined with the input's "licensePlate" section.

DATA STREAM INPUT

GENERAL NOTES

Events generated have the following format:


```
{
  "vehicleTypeID": 319 ,
  "licensePlate": "GJC-6886" ,
  "speed": "31" ,
  "colorID": 2 ,
  "checkpointID": 832 ,
  "spotType": "Speed_Limit_Camera"
}
```


FIELDS DESCRIPTION

- **vehicleTypeID:** Information about the type of the car. Can be joined with CAR_DATA.csv.
- **licensePlate:** The license plate of the car. Can be joined with WANTED_CARS.txt.
- **speed:** The speed of the car at the time the event was generated. All cars need to stop at the toll station, so their speed is zero.
- **colorID:** Information about the color of the car. Can be joined with COLORS.json.
- **checkpointID:** Information about the checkpoint that produced the event. Can be joined with SPEED_CAMERA_SPOTS.csv or TOLL_STATIONS.txt (depending on the "spotType" value, explained below)
- **spotType:** Describes the type of the check point. Can take the values "Toll_Station" and "Speed_Limit_Camera".

SAMPLE

The file "GeneratorDataSample.txt" contains more samples of events generated by the event generator. This doesn't have to be used anywhere else in the assignment.

QUERIES

Query 1:

In a **tumbling window** of 1 minute count the number of Audis that passed through a toll station.

Query 2:

In a **hopping window** of 3 minutes, for each color, calculate the total number of cars that passed through a police speed limit camera. Repeat every 90 seconds.

Query 3:

In a **tumbling window** of 20 seconds, for each color, find the oldest car that passed through a toll station.

Query 4:

In a **sliding window** of 60 seconds, calculate the speed limit camera spots where the most violations happened.

Query 5:

In a **sliding window** of five minutes, for each color and car model, display the total number of cars that break the speed limit.

Query 6:

You have been given a list of the license plates of police's most wanted criminals. In a **sliding window** of 1 minute, display a list of all the cars that you spotted at any checkpoint.

Query 7:

In a **sliding window** of 1 minute, display a list of fake license plates. Check if the same license plate has passed through any type of checkpoint twice in the same time window.

Query 8:

In a **tumbling window** of 2 minutes, calculate the percentage of BMW drivers that break the speed limit. (eg Out of all the BMW drivers that were identified in the last 2 minutes, 80% broke the speed limit).