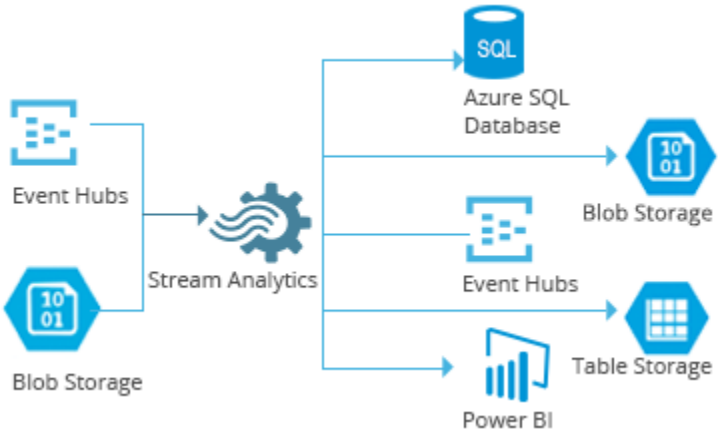
ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

BUSINESS ANALYTICS
Master of Science

Modern Data Management & Business Intelligence
Assignment 3: Azure Streams Analytics



Dimitrios-Gerasimos Anastasatos, AM: BAFT1702

Athens, January 2018

- # Introduction

Nowadays, there is the need to monitor / analyze data in near real time, as we live in a fast paced world (Social Media, Internet of Things etc.) and vast amounts of data are flowing at high velocity over the wire. Capturing and storing event data for later analysis is no longer enough. Organizations that can process and act on this streaming data in real time can dramatically improve efficiencies and differentiate themselves in the market.

Azure Stream Analytics is a fully managed, cost effective real-time event processing engine that helps to unlock deep insights from data. A user can author a Stream Analytics job by specifying the input source of the streaming data, the output sink for the results of your job, and a data transformation expressed in a SQL-like language.

- # Motivation

We have access to a data stream of vehicle observations. The data stream is 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. We will create an Azure Analytics solution to answer some stream queries.

- # Setup a Stream Analytics Job

In Figure 1, the stream input in the Event Hub along with the input reference data, saved as blob storage, are shown. Figure 2 depicts the corresponding reference json files uploaded.

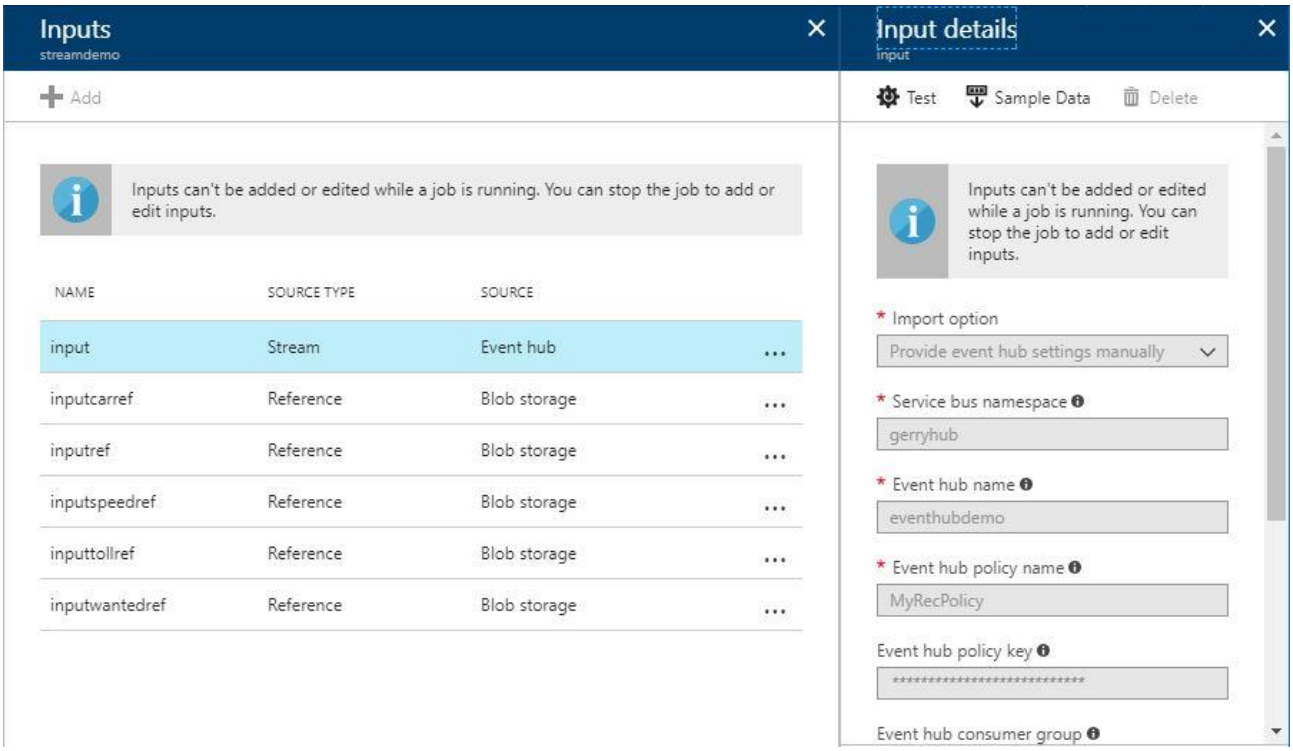


Figure 1

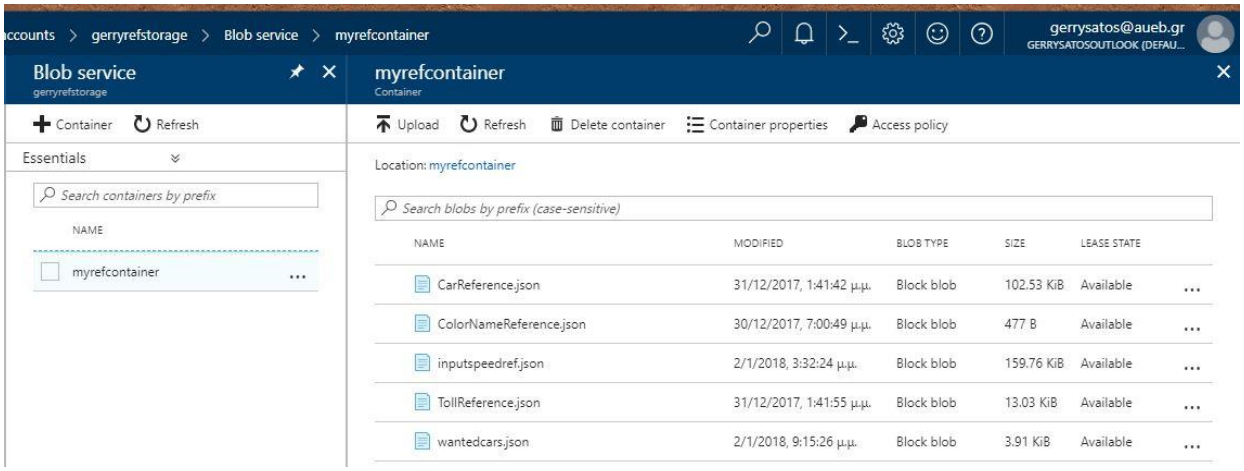


Figure 2

Next in Figure 3, we see on the left all the necessary components for a Stream Analytics Job: the storage account, the event hub, the reference storage, and the stream demo. In the same figure, we also notice an actual Job running. The same job is shown in Figure 4, as well, along with all the inputs (stream input and reference) and the outputs (stream output, Power BI output and some test outputs).

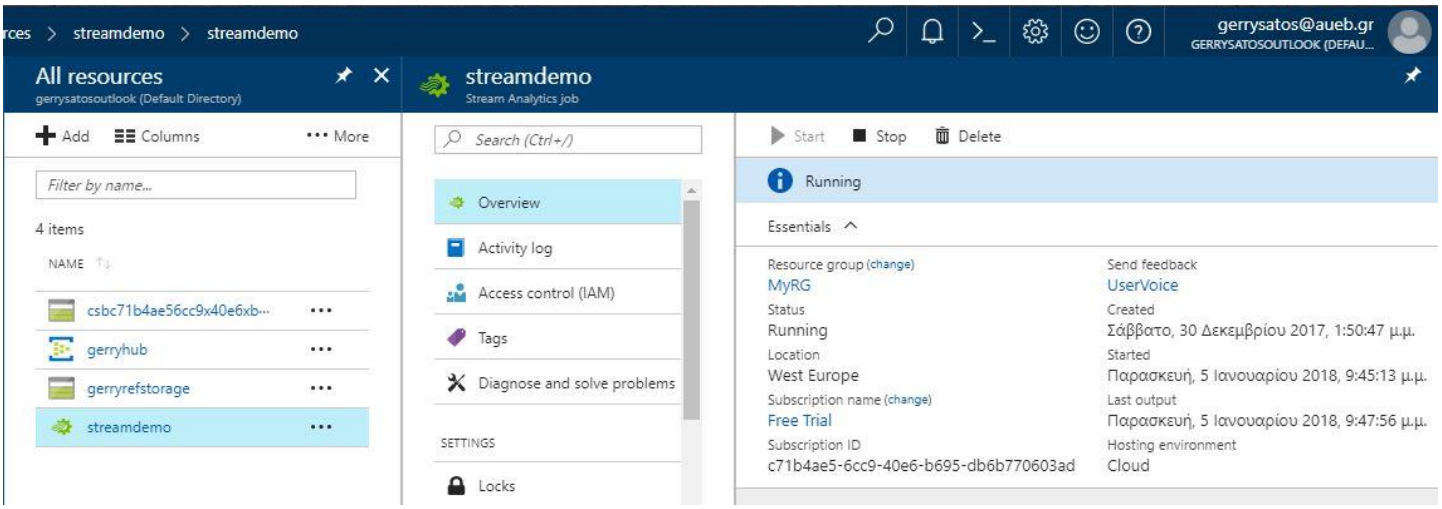


Figure 3

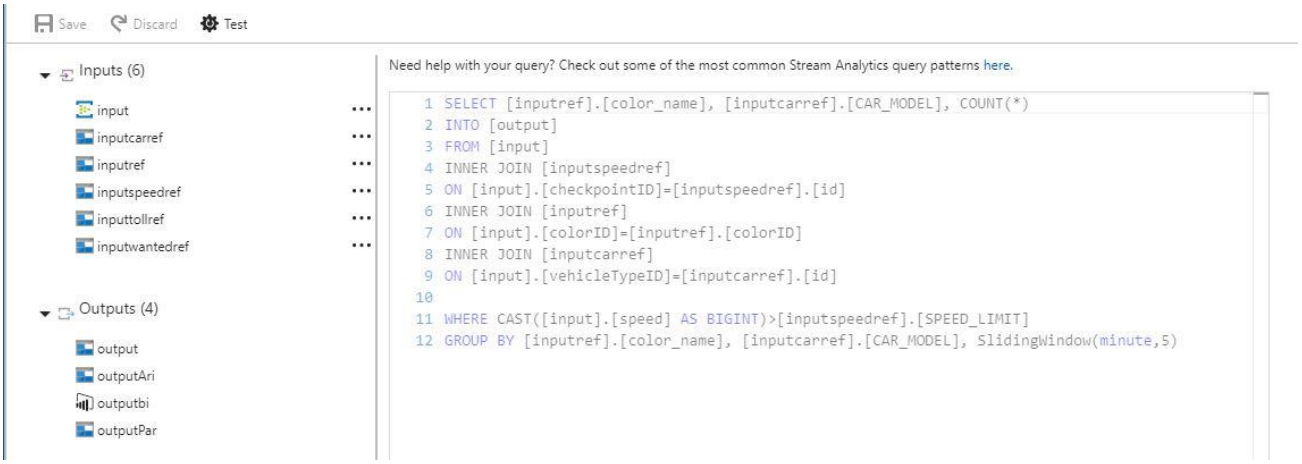


Figure 4

Finally, Figure 5 shows the data generator.

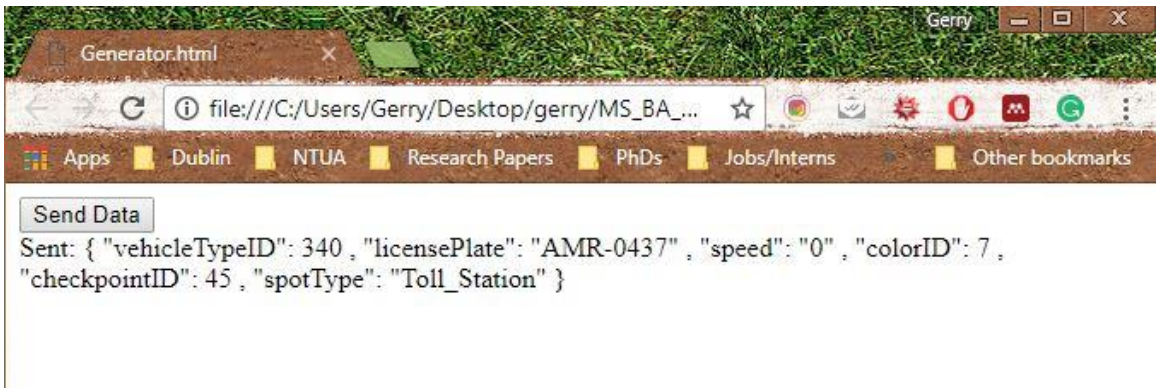


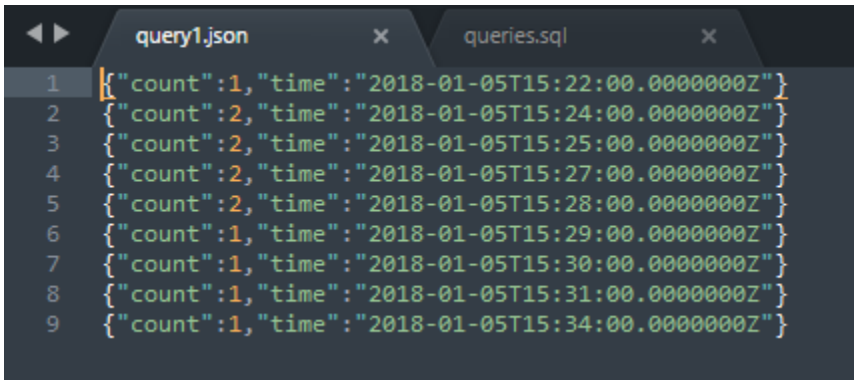
Figure 5

- Query 1

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

```
SELECT COUNT(*)
INTO [output]
    FROM [input]
    INNER JOIN [inputcarref] -- CAR_DATA.csv
    ON [input].[vehicleTypeID]=[inputcarref].[id]
WHERE [input].[spotType]='Toll_Station' AND
[inputcarref].[CAR_MAKE]='Audi'
GROUP BY [inputcarref].[CAR_MAKE],TumblingWindow(second,60)
```

Some results from the stream analytics job are following:



- 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.

```
SELECT [inputref].[color_name],COUNT(*), System.TimeStamp as Time

into [output]

    FROM [input]
    INNER JOIN [inputref] -- COLORS.json
    ON [input].[colorID]=[inputref].[colorID]

WHERE [input].[spotType]='Speed_Limit_Camera'
GROUP BY [inputref].[color_name], HoppingWindow(second,180,90)
```

Some results are following:

query2.json | query6.json | query4.json


```
{"color_name":"Blue","count":3,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Yellow","count":3,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Brown","count":4,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Pink","count":4,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Black","count":1,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Green","count":3,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Silver","count":6,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"White","count":3,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Red","count":1,"time":"2018-01-05T16:42:00.0000000Z"}
{"color_name":"Blue","count":8,"time":"2018-01-05T16:43:30.0000000Z"}
{"color_name":"Yellow","count":6,"time":"2018-01-05T16:43:30.0000000Z"}
{"color_name":"Brown","count":8,"time":"2018-01-05T16:43:30.0000000Z"}
{"color_name":"Pink","count":8,"time":"2018-01-05T16:43:30.0000000Z"}
```


- Query 3

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

```
WITH View1 AS(
SELECT [inputref].[color_name] AS Color,
MIN([inputcarref].[CAR_MODEL_YEAR]) AS Min_Year, System.TimeStamp as Time
    FROM [input]
    INNER JOIN [inputref]
    ON [input].[colorID]=[inputref].[colorID]
    INNER JOIN [inputcarref]
    ON [input].[vehicleTypeID]=[inputcarref].[id]
WHERE [input].[spotType]='Toll_Station'
GROUP BY [inputref].[color_name], TumblingWindow(second,20)
)

SELECT [inputref].[color_name],[input].[vehicleTypeID]
,[inputcarref].[CAR_MODEL_YEAR], System.TimeStamp as Time
INTO [output]
FROM [input]
INNER JOIN View1
ON DATEDIFF(second,input,View1) BETWEEN 0 AND 20
INNER JOIN [inputcarref]
ON [input].[vehicleTypeID]=[inputcarref].[id]
INNER JOIN [inputref]
ON [input].[colorID]=[inputref].[colorID]

WHERE [inputcarref].[CAR_MODEL_YEAR]=[View1].[Min_Year]
AND [inputref].[color_name]=[View1].[Color]
```

Some results are following:

query3.json | query1.json | query8a.json | queries.sql


```
{"color_name":"Yellow","vehicletypeid":429,"car_model_year":1972,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"White","vehicletypeid":419,"car_model_year":1987,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Black","vehicletypeid":723,"car_model_year":1991,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Blue","vehicletypeid":90,"car_model_year":1998,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Red","vehicletypeid":268,"car_model_year":2011,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Pink","vehicletypeid":565,"car_model_year":1997,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Brown","vehicletypeid":45,"car_model_year":1966,"time":"2018-01-05T15:50:00.0000000Z"}
{"color_name":"Green","vehicletypeid":639,"car_model_year":1981,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"Silver","vehicletypeid":817,"car_model_year":2012,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"Yellow","vehicletypeid":365,"car_model_year":1993,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"Black","vehicletypeid":160,"car_model_year":2002,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"White","vehicletypeid":381,"car_model_year":2004,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"Brown","vehicletypeid":51,"car_model_year":1994,"time":"2018-01-05T15:50:20.0000000Z"}
{"color_name":"Green","vehicletypeid":173,"car_model_year":2008,"time":"2018-01-05T15:50:40.0000000Z"}
{"color_name":"Silver","vehicletypeid":896,"car_model_year":1999,"time":"2018-01-05T15:50:40.0000000Z"}
{"color_name":"Yellow","vehicletypeid":10,"car_model_year":2003,"time":"2018-01-05T15:50:40.0000000Z"}
```

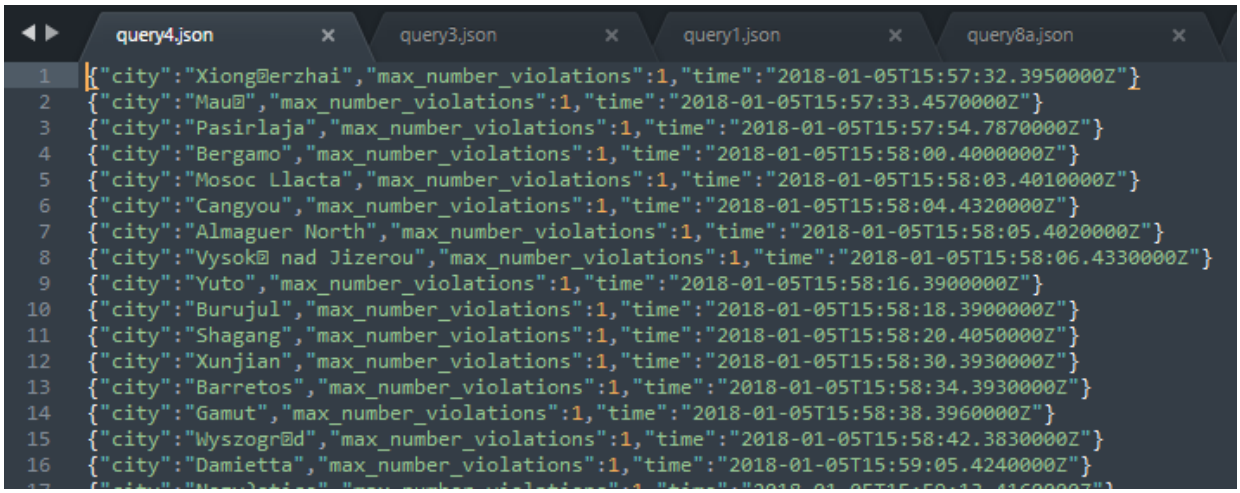

- Query 4

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

```
WITH View1 AS (
    SELECT [inputspeedref].[id] AS violationID ,COUNT(*) AS
ViolationsPerSpot
    FROM [input]
    INNER JOIN [inputspeedref]
    ON [input].[checkpointID]=[inputspeedref].[id]
    WHERE [input].[spotType]='Speed_Limit_Camera' AND CAST([input].[speed]
AS BIGINT)>[inputspeedref].[SPEED_LIMIT]
    GROUP BY [inputspeedref].[id],SlidingWindow(second,60)
)

SELECT [inputspeedref].[CITY], MAX([View1].[ViolationsPerSpot]) As
Max_Number_Violations, System.Timestamp AS Time
INTO [output]
FROM [View1]
INNER JOIN [inputspeedref]
ON [View1].[violationID]=[inputspeedref].[id]
GROUP BY [inputspeedref].[CITY], SlidingWindow(second,60)
```

Some results:



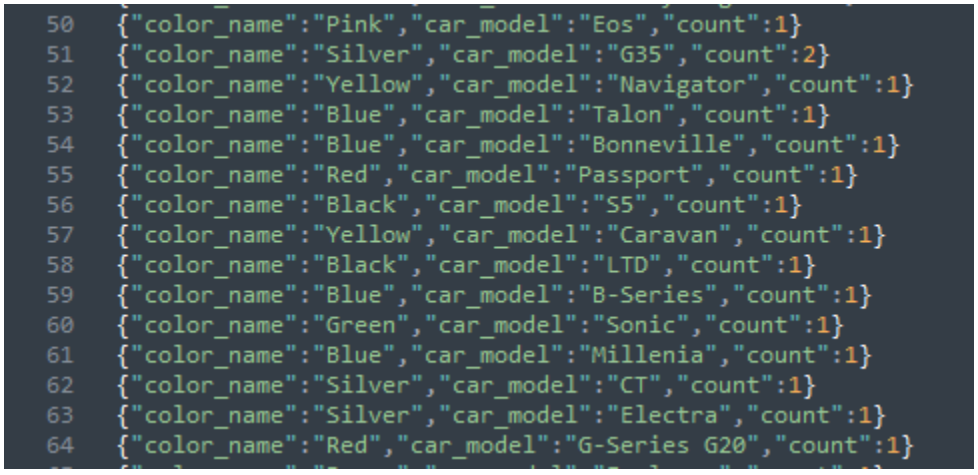
- Query 5

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

```
SELECT [inputref].[color_name], [inputcarref].[CAR_MODEL], COUNT(*)
INTO [output]
FROM [input]
INNER JOIN [inputspeedref]
ON [input].[checkpointID]=[inputspeedref].[id]
INNER JOIN [inputref]
ON [input].[colorID]=[inputref].[colorID]
INNER JOIN [inputcarref]
ON [input].[vehicleTypeID]=[inputcarref].[id]

WHERE CAST([input].[speed] AS BIGINT)>[inputspeedref].[SPEED_LIMIT]
GROUP BY [inputref].[color_name], [inputcarref].[CAR_MODEL],
SlidingWindow(minute,5)
```

Some results:

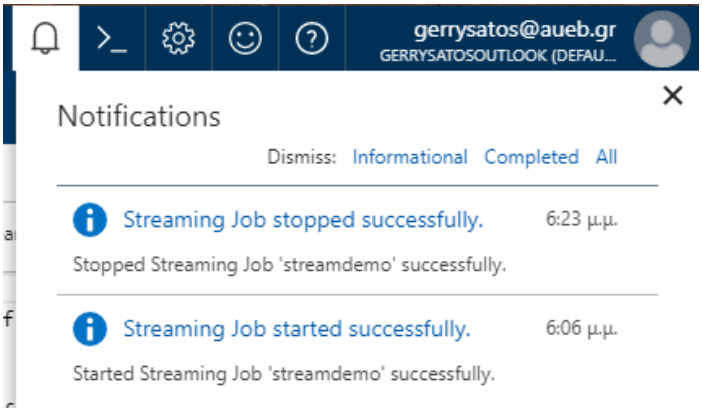


- Query 6

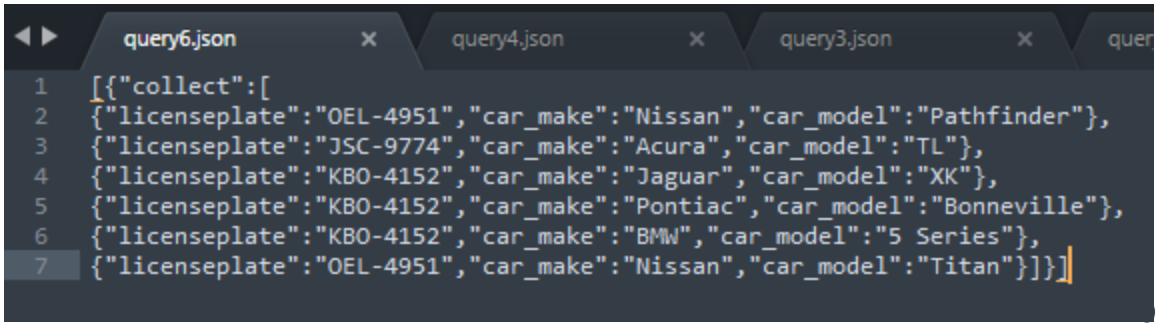
You have been given a list of the license plates of police's most wanted criminals. In a sliding window of 1 minute, display a list of all the cars that you spotted at any checkpoint.

```
WITH V1 AS(
SELECT [input].[licensePlate], [inputcarref].[CAR_MAKE],
[inputcarref].[CAR_MODEL]
FROM [input]
INNER JOIN [inputwantedref]
ON [input].[licensePlate] = [inputwantedref].[licensePlate]
INNER JOIN [inputcarref]
ON [input].[vehicleTypeID] = [inputcarref].[id]
WHERE [input].[licensePlate] = [inputwantedref].[licensePlate]
)
SELECT Collect()
INTO [output]
FROM [V1]
GROUP BY SlidingWindow(second,60)
```

Letting the stream to run for a while, no output was produced. That happens probably because all the combinations of licenses plates are extremely a lot and it didn't manage to spot a wanted license plate.



Testing our query with sample data from file 'stream-demo-input' we see that it works.

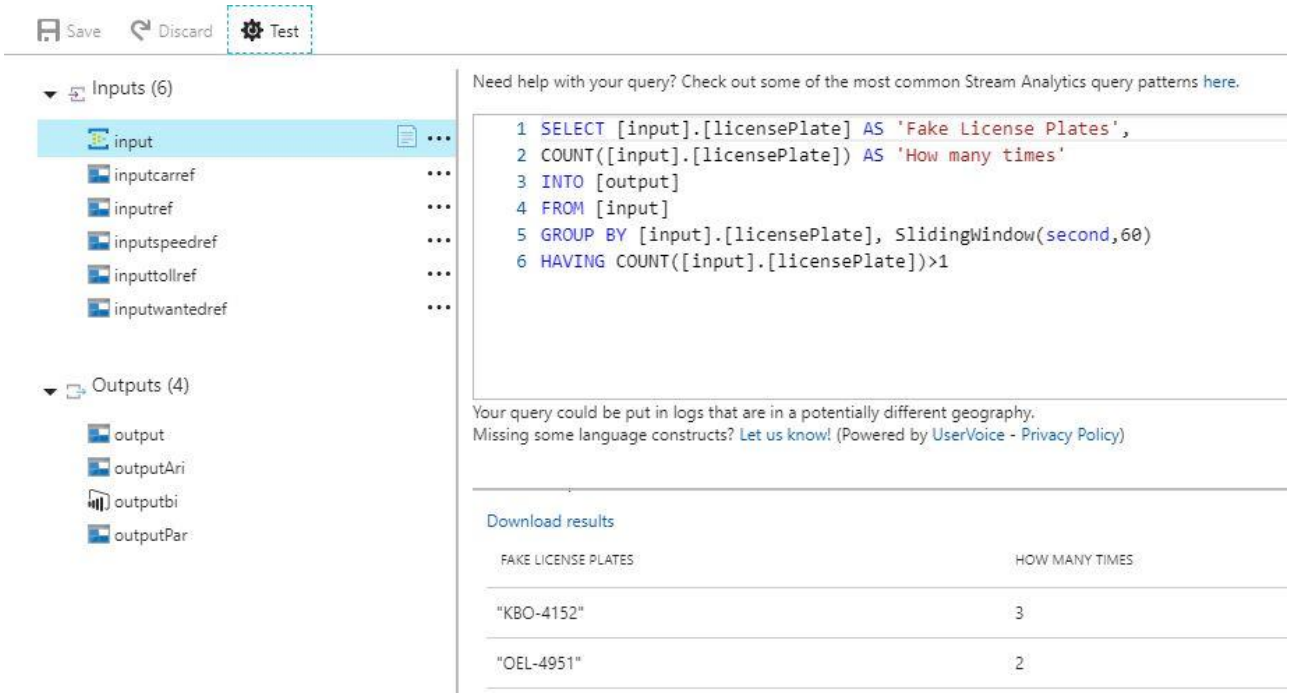


- Query 7

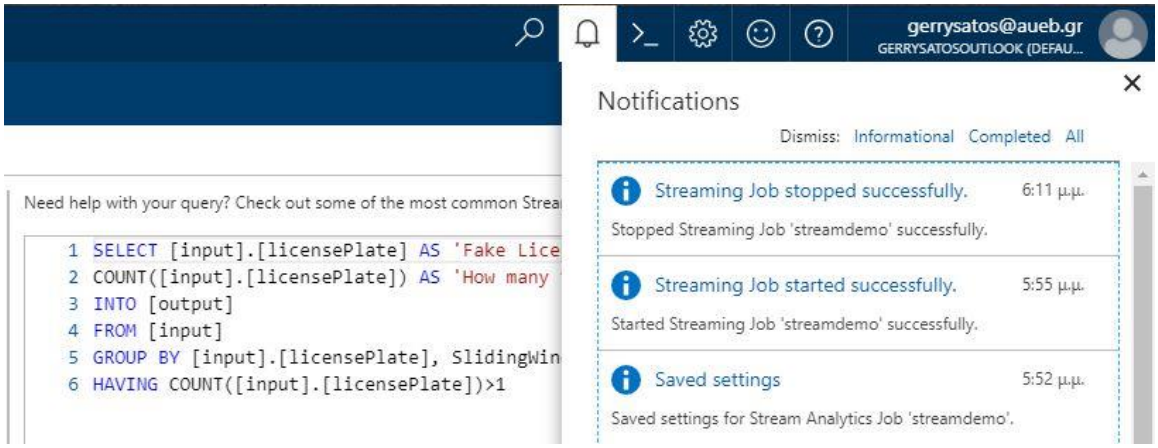
In a sliding window of 1 minute, display a list of fake license plates. Check if the same license plate has passed through any type of checkpoint twice in the same time window.

```
SELECT [input].[licensePlate] AS 'Fake License Plates',
COUNT([input].[licensePlate]) AS 'How many times'
INTO [output]
FROM [input]
GROUP BY [input].[licensePlate], SlidingWindow(second,60)
HAVING COUNT([input].[licensePlate])>1
```

Testing our query with sample data from file 'stream-demo-input' we see that it works.



Letting now the stream to run for 16 minutes, no output was produced. That happens probably because all the combinations of licenses plates are extremely a lot and it didn't manage to find 2 or more identical plates.



- Query 8

  In a tumbling window of 2 minutes, calculate the percentage of BMW drivers that break the speed limit.

```
WITH V1 AS(
SELECT COUNT([input].[vehicleTypeID]) AS Arithmitis
FROM [input]
JOIN [inputcarref]
ON [input].[vehicleTypeID]=[inputcarref].[id]
JOIN [inputspeedref]
ON [input].[checkpointID]=[inputspeedref].[id]
WHERE [inputcarref].[CAR_MAKE]='BMW' AND
        CAST([input].[speed] AS BIGINT)>[inputspeedref].[SPEED_LIMIT]
GROUP BY TumblingWindow(second,120)
), V2 AS(
SELECT COUNT([input].[vehicleTypeID]) AS Paronomastis
FROM [input]
JOIN [inputcarref]
ON [input].[vehicleTypeID]=[inputcarref].[id]
WHERE [inputcarref].[CAR_MAKE]='BMW'
GROUP BY TumblingWindow(second,120)
)

SELECT 100* CAST ([V1].[Arithmitis] AS float)/CAST ([V2].[Paronomastis] AS
float) AS Result
INTO [output]
FROM [V1]
INNER JOIN [V2]
ON DATEDIFF(second,[V1],[V2]) between 0 AND 120
```

Some results are following: