



Modern Data Management & Business Intelligence

Assignment #3: ATM Data Streams

Tutor: Chatziantoniou Damianos



Students

Despotis Spyridon: p2822111

Papailiou Thanasis: p2822128

2022

Contents

| | |
|--|----|
| 1. Introduction | 3 |
| 2. Process of setting up Azure Analytics Stream Flow | 4 |
| 3. Queries..... | 15 |

1. Introduction

Nowadays, event processing engines are becoming more and more popular due to their effectiveness in capturing and analysing complex streams of data with automatic actions. Specifically, in Banking sector event processing engines, can provide numerous opportunities for stronger risk management and empower intelligent banking. For example, in Greece during the year of 2021, the Greek National Bank utilized SAS Viya through Microsoft Azure in order to adopt advanced analytics and modernize and improve its profitability.

In this assignment we wanted to explore further the capabilities and the variety of elements of Azure Stream Analytics Platform. The given reference data files, contain data related to transactions in ATM machines. This report is presenting all the process we followed of setting up the account and executing queries to ensure that our flow is working property.

| <i>Reference Data</i> | <i>Explanation</i> |
|-----------------------|---|
| <i>AREA.json</i> | Geographical Information / Connects each area_code with a city and a country / The “area_code” of this dataset can be joined with the “area_code” of ATM.json |
| <i>ATM.json</i> | Information about the ATM / Connects each ATM with an area / The “atm_code” of this dataset can be joined with the input’s “ATMCode” section |
| <i>Customer.json</i> | Information about each customer / Provides demographic information about each customer / The “card_number” of this dataset can be joined with the input’s “CardNumber” section. |

Table 1-1 Reference Data Explanation

2. Process of setting up Azure Analytics Stream Flow

Firstly, we will create a **student account** in to **Azure Analytics Platform** with our university emails (AUEB).

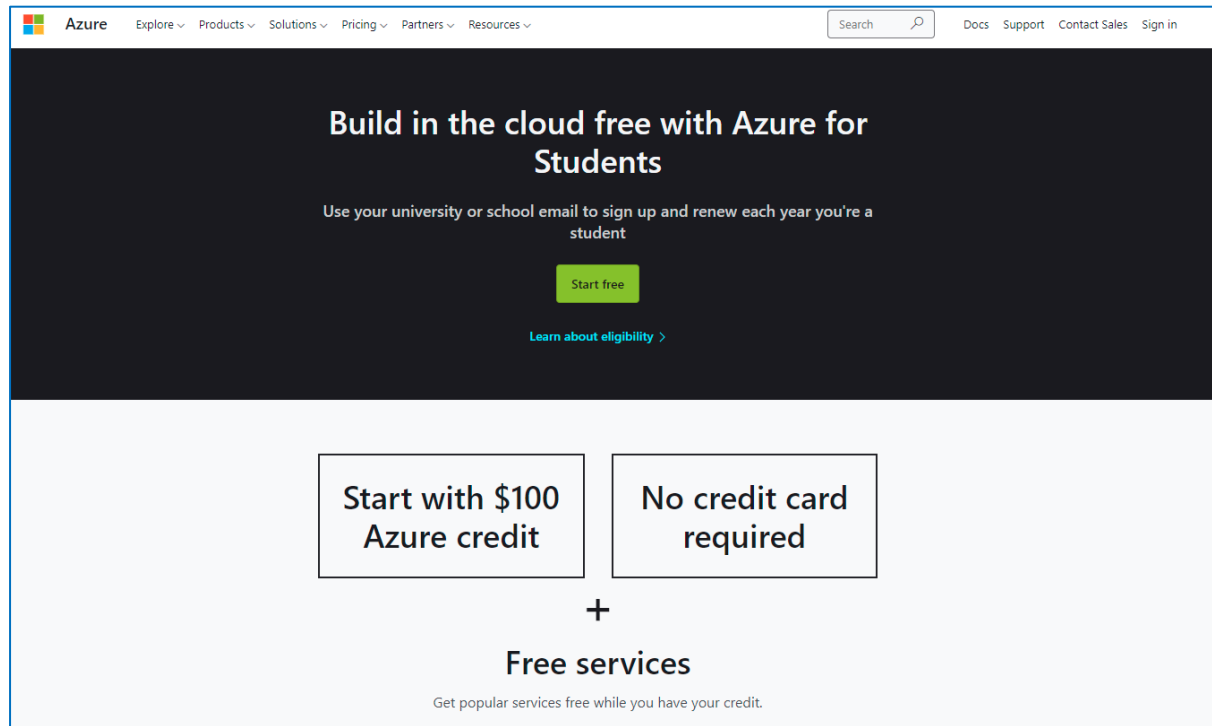


Figure 2-1 Microsoft Azure Sign Up Page for Students

Secondly, we will build a **Namespace** for our project, with the following details:

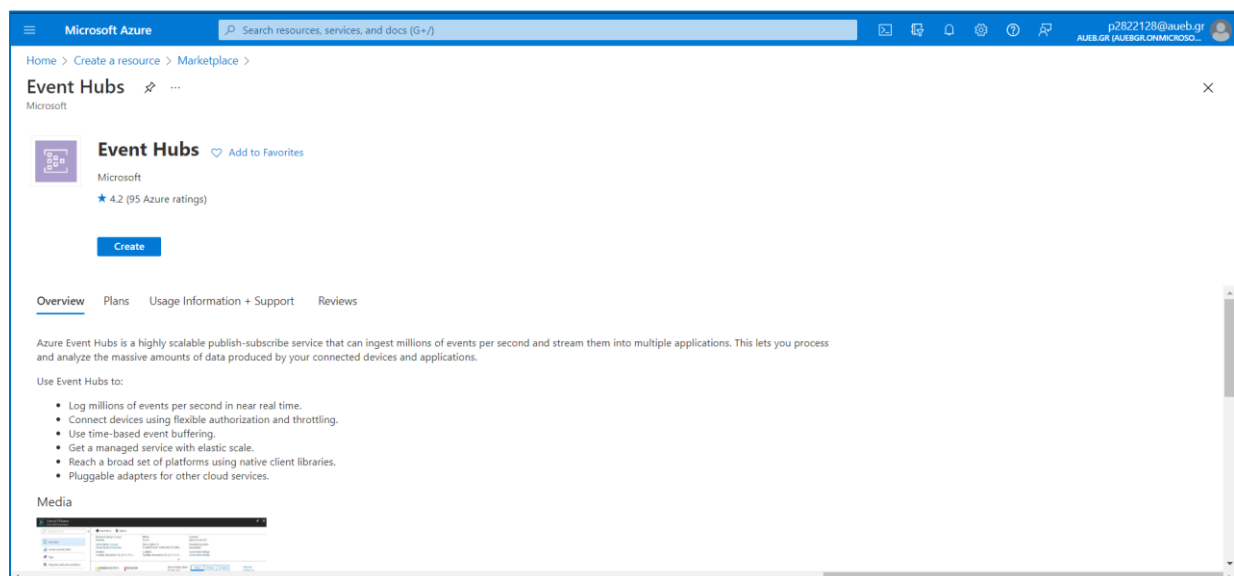


Figure 2-2 Event Hubs Service in Azure Marketplace

Create Namespace ...

Event Hubs

Basics Tags Review + create

Project Details

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

Subscription * Azure for Students

Resource group * (New) MyRG [Create new](#)

Instance Details

Enter required settings for this namespace, including a price tier and configuring the number of units (capacity).

Namespace name * mscba-aueb-streamanalytics ✓ .servicebus.windows.net

Location * West Europe

The region selected supports Availability zones. Your namespace will have Availability Zones enabled. [Learn more.](#)

Pricing tier (View full pricing details) * Standard (20 Consumer groups, 1000 Brokered connections)

Throughput Units * 1

Enable Auto-Inflate ☐

[Review + create](#) < Previous Next: Tags >

Figure 2-3 Creating Namespace for Event Hubs

Create Namespace ...

Event Hubs

Basics Tags Review + create

Project Details

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

Subscription * Azure for Students

Resource group * (New) MyRG [Create new](#)

Instance Details

Enter required settings for this namespace, including a price tier and configuring the number of units (capacity).

Namespace name * mscba-aueb-streamanalytics ✓ .servicebus.windows.net

Location * West Europe

The region selected supports Availability zones. Your namespace will have Availability Zones enabled. [Learn more.](#)

Pricing tier (View full pricing details) * Standard (20 Consumer groups, 1000 Brokered connections)

Throughput Units * 1

Enable Auto-Inflate ☐

[Review + create](#) < Previous Next: Tags >

A resource group is a container that holds related resources for an Azure solution.

Name * MyRG ✓

OK Cancel

Figure 2-4 Resource Group named “MyRG”

mscba-aueb-streamanalytics | Overview ...

Deployment

Search (Ctrl+F) Delete Cancel Redeploy Refresh

Your deployment is complete

Deployment name: mscba-aueb-streamanalytics

Subscription: Azure for Students

Resource group: MyRG

Start time: 1/11/2022, 7:53:23 PM

Correlation ID: 469e1573-7115-413e-9b8b-0beef4ba1df

Deployment details (Download)

Next steps

[Go to resource](#)

Microsoft Defender for Cloud

Secure your apps and infrastructure

[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials

Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

[Find an Azure expert >](#)

Figure 2-5 Successful Deployment Message

The we created an **Event hub**, which would receive and process our ATM events per second:

Dashboard > mscba-aueb-streamanalytics >

Create Event Hub

Event Hubs

Name

Partition Count

Message Retention

Capture

Figure 2-6 New Event Hub named "eventhubemo"

Microsoft Azure

Search resources, services, and docs (G+/I)

Dashboard > mscba-aueb-streamanalytics

Event Hubs Namespace

Search (Ctrl+/)

+ Event Hub Delete Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Shared access policies

Scale

Geo-Recovery

Networking

Encryption

Properties

Locks

Entities

Event Hubs

Essentials

Resource group (Move): MyRG

Status: Active

Location: West Europe

Subscription (Move): Azure for Students

Subscription ID: 50a3dde7-ca7e-4c76-860f-cade090c1199

Host name: mscba-aueb-streamanalytics.servicebus.windows.net

Tags (Edit): Click here to add tags

NAMESPACE CONTENTS

EVENT HUB

KAFKA SURFACE ENABLED

ZONE REDUNDANCY ENABLED

Show data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

Requests

Messages

Notifications

More events in the activity log → Dismiss all

Event Hub creation

Successfully created Event Hub eventhubdemo

a few seconds ago

Deployment succeeded

Deployment 'mscba-aueb-streamanalytics' to resource group 'MyRG' was successful.

5 minutes ago

Optimize your cloud workloads with personalized recommendations

With your Azure account, you get free, personalized recommendations to help you optimize your cloud workloads. Start with Azure Advisor recommendations—based on an analysis of your Azure usage—to improve cost-efficiency, security, reliability, performance, and operational excellence. [Learn more](#)

See recommendations >

21 minutes ago

Figure 2-7 Successful Creation

Microsoft Azure

Search resources, services, and docs (G+/I)

Dashboard > mscba-aueb-streamanalytics

mscba-aueb-streamanalytics | Event Hubs

Event Hubs Namespace

Search (Ctrl+/)

+ Event Hub Refresh

Search to filter items...

| Name | Status | Message Retention | Partition Count |
|--------------|--------|-------------------|-----------------|
| eventhubdemo | Active | 1 day | 3 |

Tags

Diagnose and solve problems

Events

Settings

Shared access policies

Scale

Geo-Recovery

Networking

Encryption

Properties

Locks

Entities

Event Hubs

Schema Registry

Monitoring

Alerts

Figure 2-8 Active Status of "eventhubdemo" Event Hub

The next step is to create the **Shared Access Policies** of the Event Hub, named “MySendPolicy” and “MyRecPolicy”:

The image shows two side-by-side windows for creating Shared Access Policies (SAS Policies) for an Event Hub. The left window is titled "SAS Policy: MySendPolicy" and the right window is titled "SAS Policy: MyRecPolicy". Both windows have a top bar with "Save", "Discard", and "Delete" buttons. Below the title bar, there are three checkboxes: "Manage", "Send", and "Listen". In the "MySendPolicy" window, "Send" is checked. In the "MyRecPolicy" window, "Listen" is checked. Below the checkboxes, there are four text boxes for keys and connection strings. The "Primary key" and "Secondary key" boxes contain long alphanumeric strings. The "Connection string-primary key" and "Connection string-secondary key" boxes contain the endpoint URL "Endpoint=sb://mscbaueb.servicebus.windows.n...".

Figure 2-9 Shared Access Policies Creation

To generate a **Security Access Signature**, we download the Events Hub Signature Generator from the above link (<https://github.com/sandrinodimattia/RedDog/releases>) and then through “RedDog.ServiceBus.EventHubs.SignatureGenerator.exe” we create our signature:

The image shows a screenshot of the "Event Hubs - Signature Generator" application. The application has a title bar with the text "Event Hubs - Signature Generator". It is divided into two main sections: "Hub" and "Credentials". The "Hub" section contains four text boxes: "Namespace" (mscba-ueb-streamanalytics), "Hub Name" (eventhubdemo), "Publisher" (Laptop), and "Mode" (Http). The "Credentials" section contains three text boxes: "Sender Key Name" (MySendPolicy), "Sender Key" (3hJ8p4gC6qlmisA1lNZy+h6UEY+flloER07Qjk=), and "Token TTL (minutes)" (8640). Below these sections is a large text box labeled "Signature" containing a long alphanumeric string. A "Generate" button is located at the bottom right of the application.

Figure 2-10 Security Access Signature Generator

Next, we edit the Generator.html file, in order to **update the CONFIG variables**:

```
/*  
*** CONFIG ***  
*/  
  
//Use the signature generator: https://github.com/sandrinodimattia/RedDog/releases  
var sas = "SharedAccessSignature sr=https%3a%2f%2fmscba-aueb-streamanalytics.servicebus.windows.net%2feventhubdemo%2fpublishers%2flaptop%2fmessages&sig=GbP3jYBxTL1GqULSWYn7DxTexkpBzsHeZ9fmJFdVFjE%3d&se=1642442790&skn=MySendPolicy";  
var serviceNamespace = "mscba-aueb-streamanalytics";  
var hubName = "eventhubdemo";  
var deviceName = "Laptop";
```

Figure 2-11 Editing Config Variables in Generator.html file

We are now ready to **feed our Event Hub** with the use of **Generator.html** in Chrome browser. We press “Send Data” to start the process:

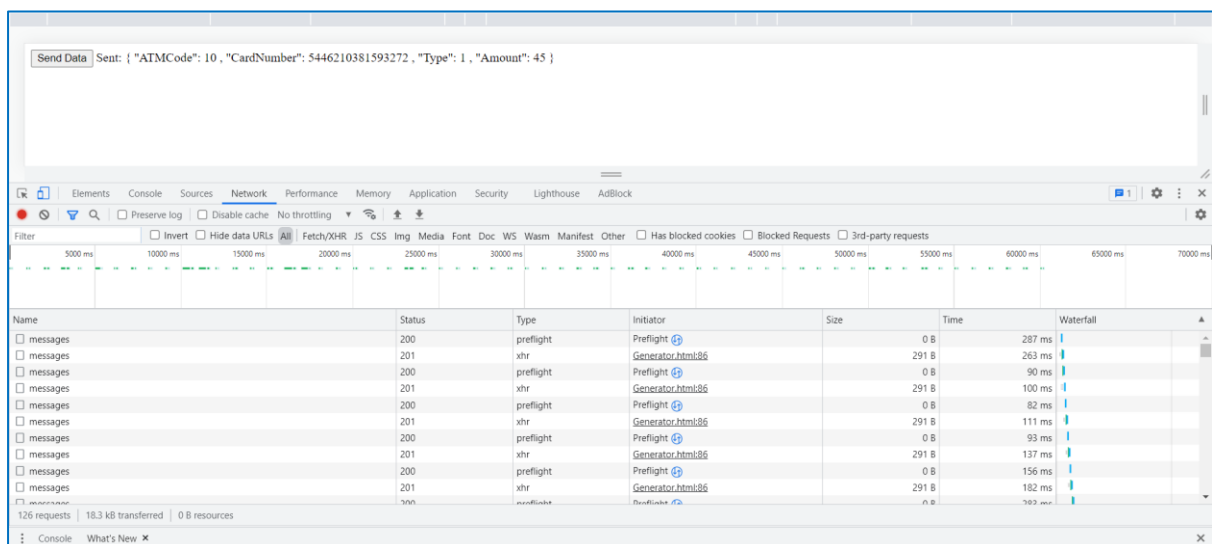


Figure 2-12 Feeding with Data the Event Hub

Afterwards, we will create an **Azure Blob Storage Account** to take advantage of cloud storage:

Microsoft Azure

Dashboard > Storage accounts >

Create a storage account

Basics Advanced Networking Data protection Encryption Tags Review + create

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription * Azure for Students

Resource group * MyRG [Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name * mscbastreamstorage

Region * (Europe) North Europe

Performance * ☒ Standard: Recommended for most scenarios (general-purpose v2 account)
☐ Premium: Recommended for scenarios that require low latency.

Redundancy * ☒ Geo-redundant storage (GRS)
☒ Make read access to data available in the event of regional unavailability.

[Review + create](#) < Previous Next: Advanced >

https://portal.azure.com/?Microsoft_Azure_Education_correlationId=995a024cab42f2acce7...

Figure 2-13 New Blob Storage Account Creation

Microsoft Azure

Dashboard >

mscbastreamstorage_1641927187509 | Overview

Deployment

Search (Ctrl+/) << Delete Cancel Redeploy Refresh

We'd love your feedback! →

✓ Your deployment is complete

Deployment name: mscbastreamstorage_1641927187509
Subscription: Azure for Students
Resource group: MyRG

Start time: 1/11/2022, 8:53:09 PM
Correlation ID: 19758de0-62e2-4d8d-ba16-feebda976fe6

Deployment details (Download)
Next steps
[Go to resource](#)

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Figure 2-14 Successful Deployment Message

Inside Blob Storage environment, we create **2 new containers**. One to upload the given **reference data files** (area.json, atm.json and customer.json) and the other one to upload **the output of the stream**:

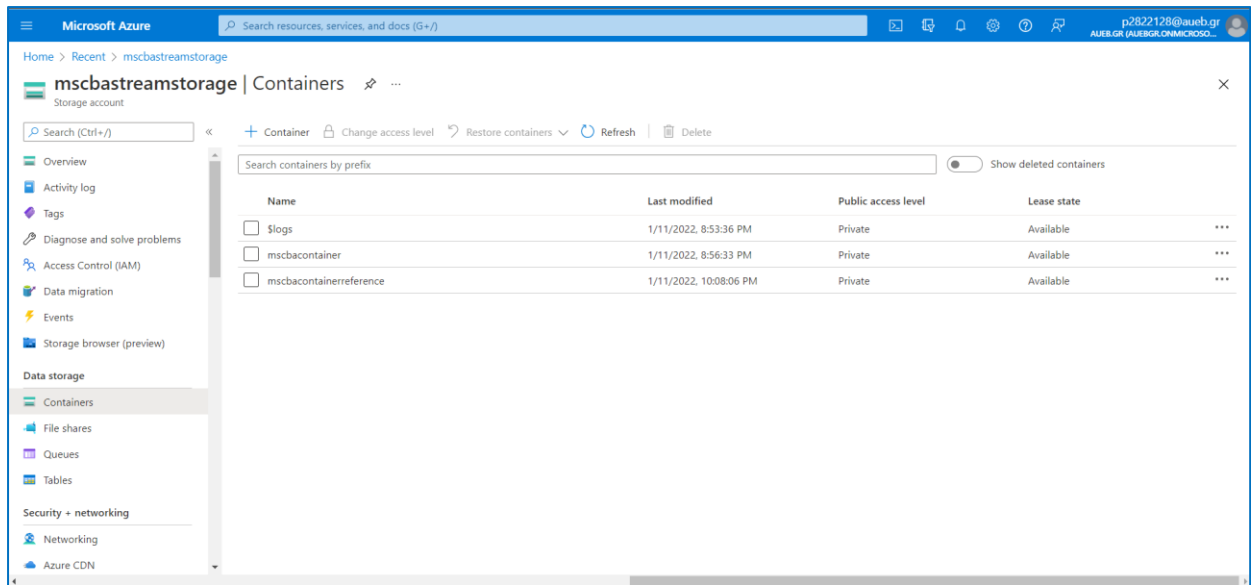


Figure 2-15 Containers Creation in Storage Account

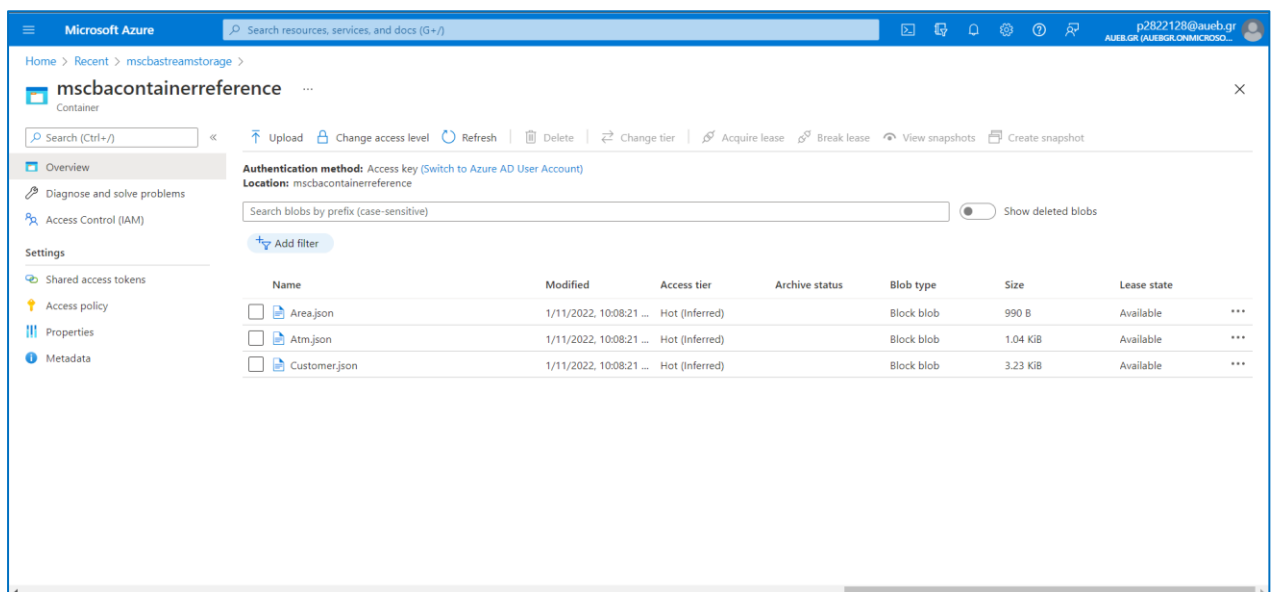


Figure 2-16 Reference Files Inside Container

We are now all set to create a **new Stream Analytics Job**. Specifically, we create a new job named “streamdemo “:

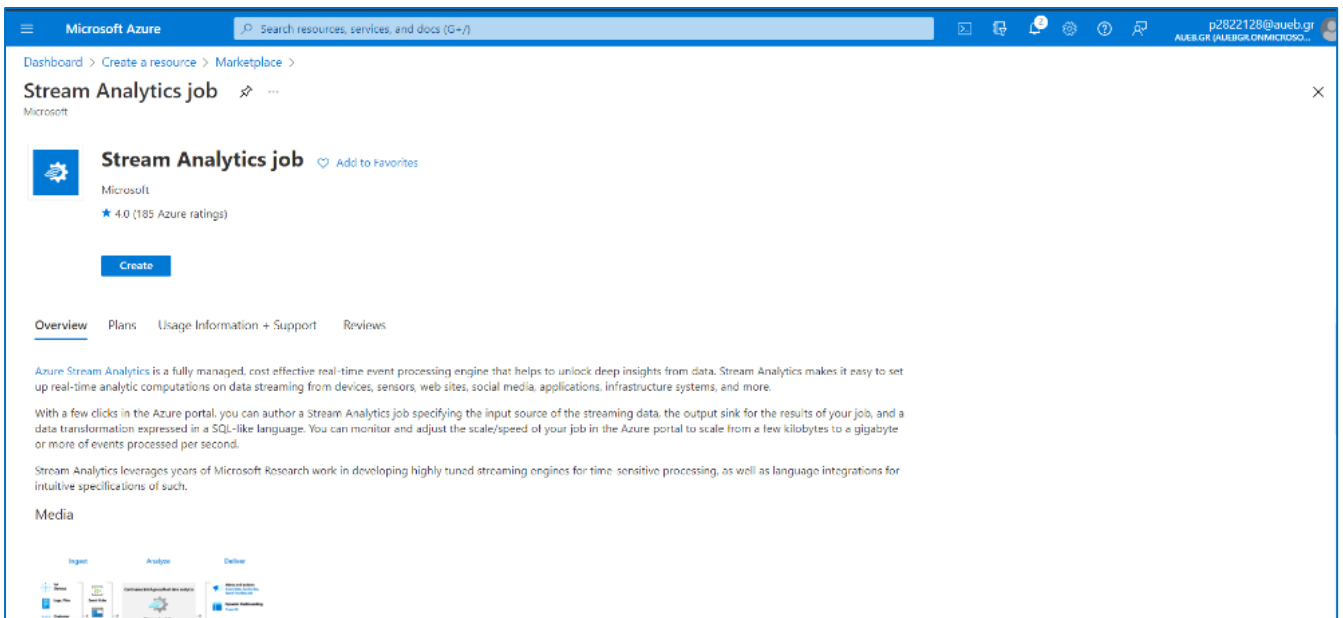


Figure 2-17 Stream Analytics job Service in Azure Marketplace

The screenshot shows the 'New Stream Analytics job' creation form in the Azure portal. The form includes several fields: 'Job name' (streamdemo), 'Subscription' (Azure for Students), 'Resource group' (MyRG), 'Location' (West Europe), 'Hosting environment' (Cloud), and 'Streaming units' (3). There is also a checkbox for 'Secure all private data assets needed by this job in my Storage account.' and a 'Create' button at the bottom.

Figure 2-18 Creation of New Stream Analytics Job

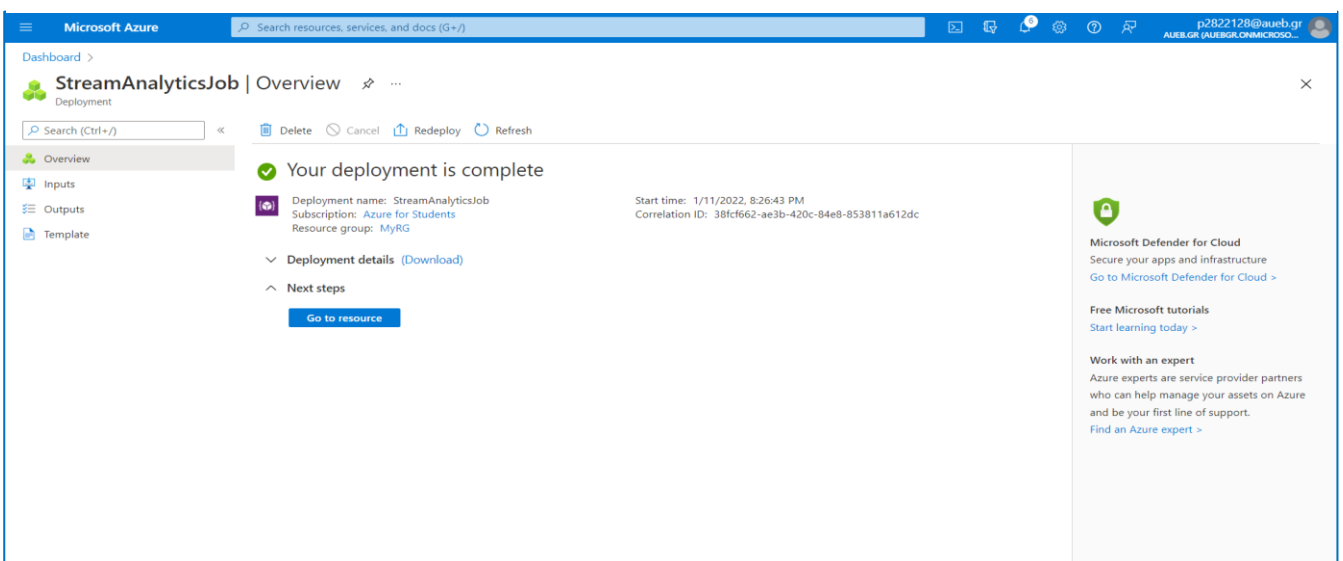


Figure 2-19 Successful Message of Deployment

Next, we **define the inputs and outputs** for our Streaming Analytics Job. At first, we add a **stream input**, using the event hub that we have created and the “MyRecPolicy” access policy:

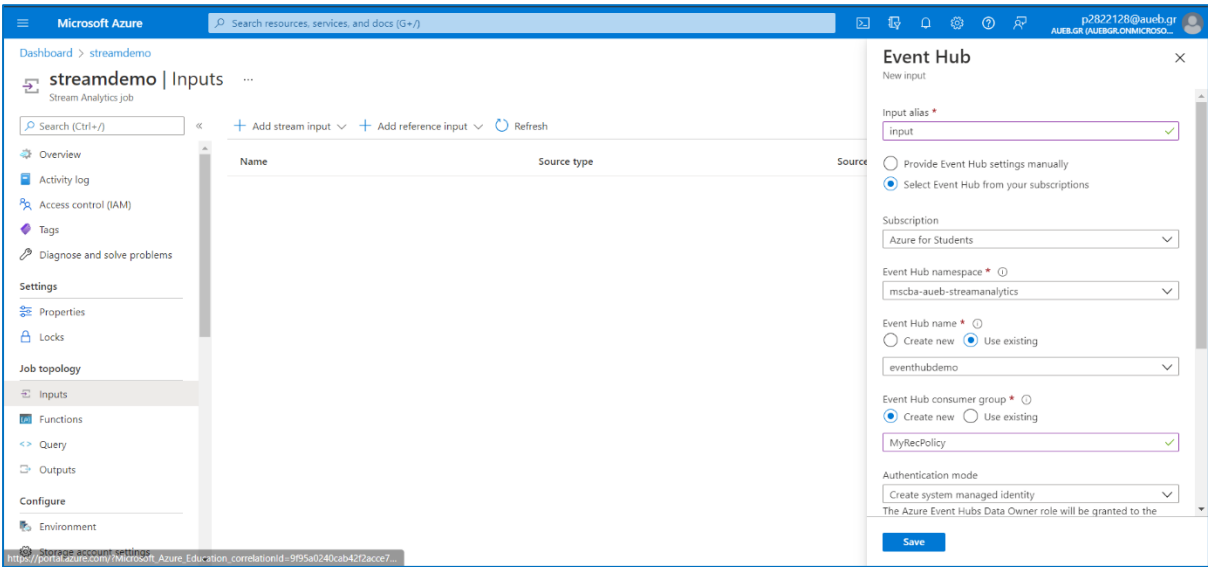


Figure 2-20 Defining Stream Input

Then we add as **reference inputs** the Area.json, Atm.json, and Customer.json files that we have uploaded in the first Blob's storage container:

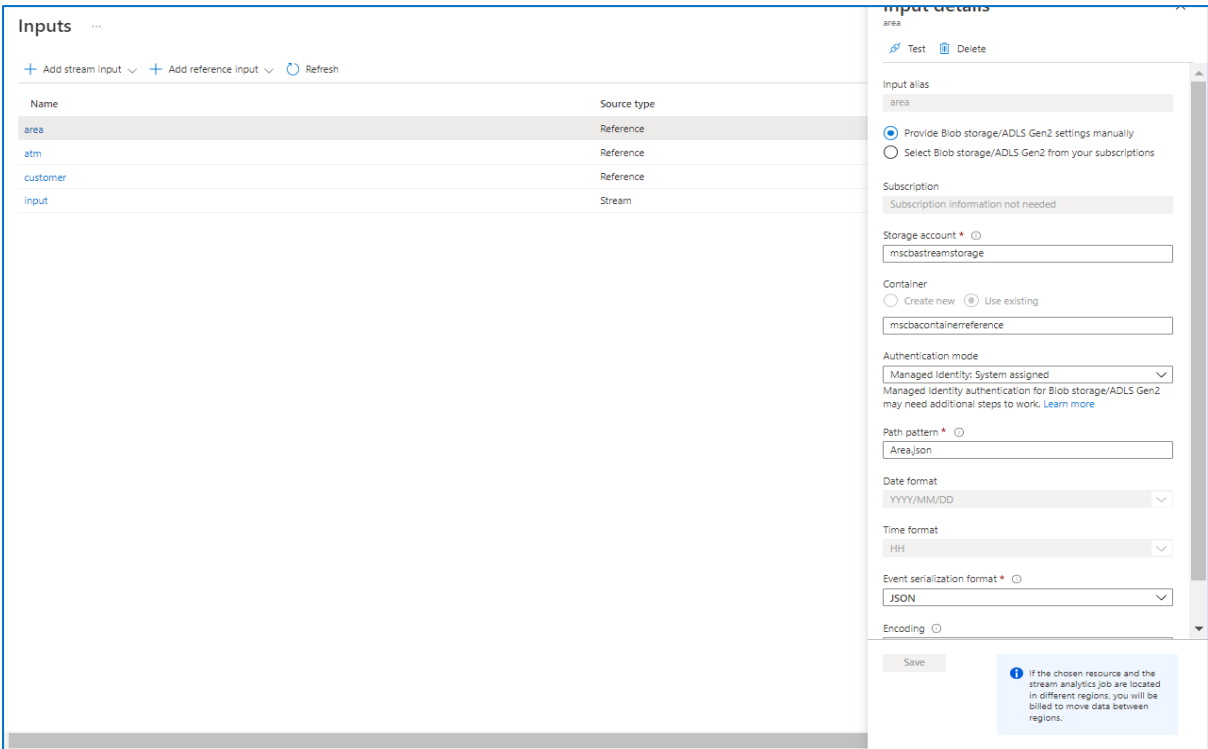


Figure 2-21 Adding Reference Inputs

Finally, we set the second container as the **stream's output**:

Outputs ...

+ Add ▾ Refresh

| Name | Sink |
|------|------|
|------|------|

Blob storage/ADLS Gen2

New output

Output alias *
output ✓

☐ Provide Blob storage/ADLS Gen2 settings manually
☒ Select Blob storage/ADLS Gen2 from your subscriptions

Subscription
Azure for Students ▾

Storage account *
mscbastreamstorage ▾

Container * ☐ Create new ☐ Use existing
msccontainer ✓

Authentication mode
Managed Identity: System assigned ▾
The Storage Blob Data Contributor role will be granted to the Managed Identity for this Stream Analytics job when you click Save. If grant fails follow the manual grant steps [here](#).

Path pattern

Date format
YYYY/MM/DD ▾

Time format
HH ▾

Save

i The selected resource and the stream analytics job are located in different regions. You will be billed to move data between regions.

Figure 2-22 Defining Stream Input

Summing up, we have 4 inputs and 1 output:

Microsoft Azure

Search resources, services, and docs (G+)

Home > streamdemo Stream Analytics job

Search (Ctrl+/)

Start Stop Delete Refresh

Stopped

Location : West europe Output watermark : Tuesday, January 11, 2022, 10:28:30 PM

Subscription (Move) : Azure for Students Cluster : Shared

Subscription ID : 50a3dde7-ca7e-4c76-860f-cade090c1199 Hosting environment : Cloud

Overview

Inputs
4
area Blob storage
atm Blob storage
atm
atm

Outputs
1
output Blob storage

Query

```
1 SELECT
2   atm.area_code, sum(input.Amount)
3 INTO
4   [output]
5 FROM
6   [input]
7 JOIN atm ON input.ATMCode = atm.atm_code
8 WHERE input.Type=1
9 GROUP BY atm.area_code , TumblingWindow(minute,1)
```

Resource utilization

Figure 2-23 Dashboard of “streamdemo” Stream Analytics Job With Inputs and Outputs

At last, in order to run all the queries above, we will create 8 outputs, and we will store the results of each query in a different output:

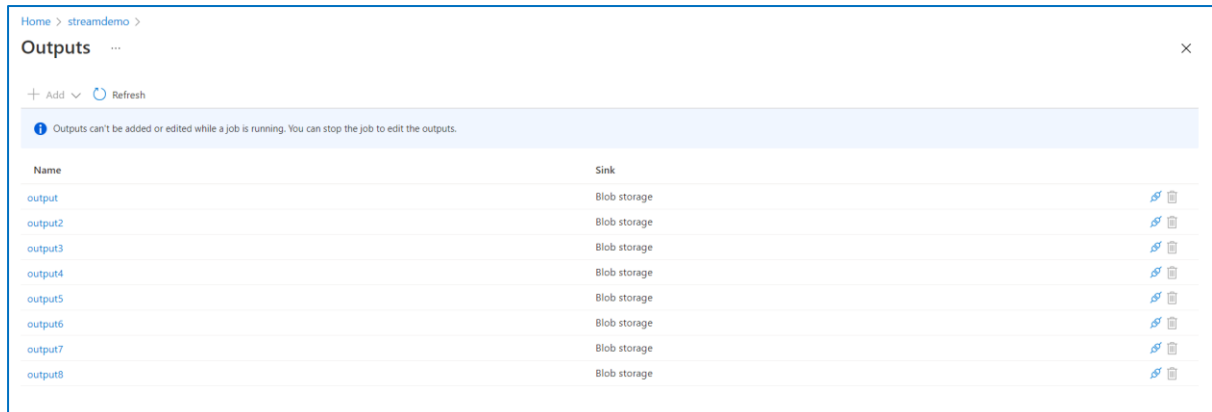


Figure 2-24 Creating Outputs in order to Store the Results

3. Queries

1. Show the total “Amount” of “Type = 0” transactions at “ATM Code = 21” of the last 10 minutes. Repeat as new events keep flowing in (use a sliding window).

➤ **Input**

```
SELECT
sum(input.Amount) AS TotalAmount,
System.Timestamp AS Time
INTO
[output]
FROM
[input]
JOIN atm ON input.ATMCode = atm.atm_code
WHERE input.Type=0 and atm.atm_code = 21
GROUP BY SlidingWindow(minute,10);
```

➤ **Output**

```
{"TotalAmount":10.0,"Time":"2022-01-16T09:27:39.4400000Z"}
{"TotalAmount":49.0,"Time":"2022-01-16T09:27:52.5030000Z"}
{"TotalAmount":77.0,"Time":"2022-01-16T09:28:02.4570000Z"}
{"TotalAmount":126.0,"Time":"2022-01-16T09:28:25.4670000Z"}
{"TotalAmount":139.0,"Time":"2022-01-16T09:29:04.4910000Z"}
{"TotalAmount":168.0,"Time":"2022-01-16T09:29:19.4780000Z"}
{"TotalAmount":201.0,"Time":"2022-01-16T09:30:23.9240000Z"}
{"TotalAmount":215.0,"Time":"2022-01-16T09:30:35.7560000Z"}
{"TotalAmount":260.0,"Time":"2022-01-16T09:30:38.5380000Z"}
{"TotalAmount":287.0,"Time":"2022-01-16T09:31:51.6160000Z"}
{"TotalAmount":306.0,"Time":"2022-01-16T09:32:08.1970000Z"}
{"TotalAmount":317.0,"Time":"2022-01-16T09:32:43.6900000Z"}
{"TotalAmount":348.0,"Time":"2022-01-16T09:33:37.0560000Z"}
{"TotalAmount":390.0,"Time":"2022-01-16T09:34:29.5690000Z"}
{"TotalAmount":414.0,"Time":"2022-01-16T09:37:22.6160000Z"}
{"TotalAmount":404.0,"Time":"2022-01-16T09:37:39.4400000Z"}
{"TotalAmount":365.0,"Time":"2022-01-16T09:37:52.5030000Z"}
{"TotalAmount":337.0,"Time":"2022-01-16T09:38:02.4570000Z"}
{"TotalAmount":288.0,"Time":"2022-01-16T09:38:25.4670000Z"}
{"TotalAmount":275.0,"Time":"2022-01-16T09:39:04.4910000Z"}
{"TotalAmount":246.0,"Time":"2022-01-16T09:39:19.4780000Z"}
{"TotalAmount":213.0,"Time":"2022-01-16T09:40:23.9240000Z"}
{"TotalAmount":199.0,"Time":"2022-01-16T09:40:35.7560000Z"}
{"TotalAmount":154.0,"Time":"2022-01-16T09:40:38.5380000Z"}
{"TotalAmount":127.0,"Time":"2022-01-16T09:41:51.6160000Z"}
{"TotalAmount":108.0,"Time":"2022-01-16T09:42:08.1970000Z"}
{"TotalAmount":97.0,"Time":"2022-01-16T09:42:43.6900000Z"}
{"TotalAmount":66.0,"Time":"2022-01-16T09:43:37.0560000Z"}
{"TotalAmount":24.0,"Time":"2022-01-16T09:44:29.5690000Z"}
{"TotalAmount":18.0,"Time":"2022-01-16T09:57:04.4990000Z"}
```

```
{"TotalAmount":46.0,"Time":"2022-01-16T09:58:09.6640000Z"}
{"TotalAmount":58.0,"Time":"2022-01-16T09:58:11.5690000Z"}
{"TotalAmount":77.0,"Time":"2022-01-16T09:58:48.6650000Z"}
{"TotalAmount":106.0,"Time":"2022-01-16T09:58:52.6070000Z"}
{"TotalAmount":118.0,"Time":"2022-01-16T09:59:09.6710000Z"}
{"TotalAmount":142.0,"Time":"2022-01-16T09:59:21.7410000Z"}
{"TotalAmount":188.0,"Time":"2022-01-16T10:00:10.5310000Z"}
{"TotalAmount":170.0,"Time":"2022-01-16T10:07:04.4990000Z"}
{"TotalAmount":142.0,"Time":"2022-01-16T10:08:09.6640000Z"}
{"TotalAmount":130.0,"Time":"2022-01-16T10:08:11.5690000Z"}
{"TotalAmount":111.0,"Time":"2022-01-16T10:08:48.6650000Z"}
{"TotalAmount":82.0,"Time":"2022-01-16T10:08:52.6070000Z"}
{"TotalAmount":70.0,"Time":"2022-01-16T10:09:09.6710000Z"}
{"TotalAmount":46.0,"Time":"2022-01-16T10:09:21.7410000Z"}
{"TotalAmount":10.0,"Time":"2022-01-16T10:12:22.5100000Z"}
{"TotalAmount":41.0,"Time":"2022-01-16T10:20:22.4670000Z"}
{"TotalAmount":31.0,"Time":"2022-01-16T10:22:22.5100000Z"}
{"TotalAmount":45.0,"Time":"2022-01-16T10:44:39.4950000Z"}
{"TotalAmount":92.0,"Time":"2022-01-16T10:44:52.4950000Z"}
{"TotalAmount":119.0,"Time":"2022-01-16T10:44:55.4640000Z"}
{"TotalAmount":157.0,"Time":"2022-01-16T10:44:58.4490000Z"}
{"TotalAmount":176.0,"Time":"2022-01-16T10:44:59.4800000Z"}
{"TotalAmount":211.0,"Time":"2022-01-16T10:46:23.5290000Z"}
{"TotalAmount":252.0,"Time":"2022-01-16T10:46:25.8730000Z"}
{"TotalAmount":283.0,"Time":"2022-01-16T10:47:26.5120000Z"}
{"TotalAmount":309.0,"Time":"2022-01-16T10:48:01.5140000Z"}
{"TotalAmount":343.0,"Time":"2022-01-16T10:48:08.8440000Z"}
{"TotalAmount":298.0,"Time":"2022-01-16T10:54:39.4950000Z"}
{"TotalAmount":251.0,"Time":"2022-01-16T10:54:52.4950000Z"}
{"TotalAmount":224.0,"Time":"2022-01-16T10:54:55.4640000Z"}
{"TotalAmount":186.0,"Time":"2022-01-16T10:54:58.4490000Z"}
{"TotalAmount":167.0,"Time":"2022-01-16T10:54:59.4800000Z"}
{"TotalAmount":132.0,"Time":"2022-01-16T10:56:23.5290000Z"}
{"TotalAmount":91.0,"Time":"2022-01-16T10:56:25.8730000Z"}
{"TotalAmount":60.0,"Time":"2022-01-16T10:57:26.5120000Z"}
{"TotalAmount":34.0,"Time":"2022-01-16T10:58:01.5140000Z"}
{"TotalAmount":43.0,"Time":"2022-01-16T11:04:17.5610000Z"}
{"TotalAmount":67.0,"Time":"2022-01-16T11:04:28.6730000Z"}
{"TotalAmount":93.0,"Time":"2022-01-16T11:04:47.0680000Z"}
{"TotalAmount":126.0,"Time":"2022-01-16T11:05:42.6220000Z"}
{"TotalAmount":156.0,"Time":"2022-01-16T11:05:53.6870000Z"}
{"TotalAmount":178.0,"Time":"2022-01-16T11:06:16.6880000Z"}
{"TotalAmount":223.0,"Time":"2022-01-16T11:06:26.0800000Z"}
{"TotalAmount":180.0,"Time":"2022-01-16T11:14:17.5610000Z"}
```


2. Show the total “Amount” of “Type = 1” transactions at “ATM Code = 21” of the last hour. Repeat once every hour (use a tumbling window).
-

➤ **Input**

```
SELECT
    sum(input.Amount) AS TotalAmount,
    System.Timestamp AS Time
INTO
    [output2]
FROM
    [input]
JOIN atm ON input.ATMCode = atm.atm_code
WHERE input.Type=1 and atm.atm_code = 21
GROUP BY TumblingWindow(hour,1);
```

➤ **Output**

```
{"TotalAmount":587.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"TotalAmount":286.0,"Time":"2022-01-16T11:00:00.0000000Z"}
```

3. Show the total “Amount” of “Type = 1” transactions at “ATM Code = 21” of the last hour. Repeat once every 30 minutes (use a hopping window).
-

➤ **Input**

```
SELECT
    sum(input.Amount) AS TotalAmount,
    System.Timestamp AS Time
INTO
    [output3]
FROM
    [input]
WHERE input.Type=1 and input.ATMCode = 21
GROUP BY HoppingWindow(minute,60,30);
```

➤ **Output**

```
{"TotalAmount":488.0,"Time":"2022-01-16T11:30:00.0000000Z"}
{"TotalAmount":302.0,"Time":"2022-01-16T12:00:00.0000000Z"}
{"TotalAmount":272.0,"Time":"2022-01-16T12:30:00.0000000Z"}
{"TotalAmount":267.0,"Time":"2022-01-16T13:00:00.0000000Z"}
```

4. Show the total “Amount” of “Type = 1” transactions per “ATM Code” of the last one hour (use a sliding window).
-

➤ **Input**

```
SELECT
    input.ATMCode AS AtmCode,
    sum(input.Amount) AS TotalAmount,
    System.Timestamp AS Time
INTO
    [output4]
FROM
    [input]
WHERE input.Type=1
GROUP BY input.ATMCode , SlidingWindow(hour,1);
```

➤ **Output**

```
{"AtmCode":21,"TotalAmount":32.0,"Time":"2022-01-16T09:25:50.5410000Z"}
{"AtmCode":21,"TotalAmount":42.0,"Time":"2022-01-16T09:25:51.8220000Z"}
{"AtmCode":19,"TotalAmount":23.0,"Time":"2022-01-16T09:25:52.5410000Z"}
{"AtmCode":15,"TotalAmount":44.0,"Time":"2022-01-16T09:25:53.6350000Z"}
{"AtmCode":10,"TotalAmount":39.0,"Time":"2022-01-16T09:25:54.6040000Z"}
{"AtmCode":19,"TotalAmount":71.0,"Time":"2022-01-16T09:25:55.6670000Z"}
{"AtmCode":18,"TotalAmount":50.0,"Time":"2022-01-16T09:25:57.5100000Z"}
{"AtmCode":19,"TotalAmount":97.0,"Time":"2022-01-16T09:25:58.5260000Z"}
{"AtmCode":10,"TotalAmount":65.0,"Time":"2022-01-16T09:25:59.4790000Z"}
{"AtmCode":10,"TotalAmount":98.0,"Time":"2022-01-16T09:26:00.5100000Z"}
{"AtmCode":19,"TotalAmount":119.0,"Time":"2022-01-16T09:26:01.5110000Z"}
{"AtmCode":20,"TotalAmount":22.0,"Time":"2022-01-16T09:26:02.5260000Z"}
{"AtmCode":15,"TotalAmount":85.0,"Time":"2022-01-16T09:26:03.4640000Z"}
{"AtmCode":20,"TotalAmount":56.0,"Time":"2022-01-16T09:26:07.4170000Z"}
{"AtmCode":20,"TotalAmount":91.0,"Time":"2022-01-16T09:26:10.5270000Z"}
{"AtmCode":20,"TotalAmount":117.0,"Time":"2022-01-16T09:26:11.4490000Z"}
{"AtmCode":19,"TotalAmount":139.0,"Time":"2022-01-
```

```
16T09:26:12.6830000Z"}
{"AtmCode":10,"TotalAmount":133.0,"Time":"2022-01-
16T09:26:17.4800000Z"}
{"AtmCode":15,"TotalAmount":101.0,"Time":"2022-01-
16T09:26:20.4800000Z"}
{"AtmCode":10,"TotalAmount":181.0,"Time":"2022-01-
16T09:26:23.5150000Z"}
{"AtmCode":15,"TotalAmount":149.0,"Time":"2022-01-
16T09:26:24.4690000Z"}
{"AtmCode":18,"TotalAmount":63.0,"Time":"2022-01-
16T09:26:26.4850000Z"}
{"AtmCode":13,"TotalAmount":48.0,"Time":"2022-01-
16T09:26:30.5010000Z"}
{"AtmCode":19,"TotalAmount":181.0,"Time":"2022-01-
16T09:26:31.4690000Z"}
{"AtmCode":19,"TotalAmount":202.0,"Time":"2022-01-
16T09:26:32.5160000Z"}
{"AtmCode":15,"TotalAmount":197.0,"Time":"2022-01-
16T09:26:39.5180000Z"}
{"AtmCode":20,"TotalAmount":158.0,"Time":"2022-01-
16T09:26:39.5490000Z"}
{"AtmCode":20,"TotalAmount":170.0,"Time":"2022-01-
16T09:26:39.5640000Z"}
{"AtmCode":15,"TotalAmount":247.0,"Time":"2022-01-
16T09:26:41.4710000Z"}
{"AtmCode":10,"TotalAmount":201.0,"Time":"2022-01-
16T09:26:42.4710000Z"}
{"AtmCode":17,"TotalAmount":10.0,"Time":"2022-01-
16T09:26:44.5020000Z"}
{"AtmCode":15,"TotalAmount":292.0,"Time":"2022-01-
16T09:26:45.4550000Z"}
{"AtmCode":13,"TotalAmount":84.0,"Time":"2022-01-
16T09:26:46.5650000Z"}
{"AtmCode":19,"TotalAmount":224.0,"Time":"2022-01-
16T09:26:47.7210000Z"}
{"AtmCode":15,"TotalAmount":332.0,"Time":"2022-01-
16T09:26:49.9260000Z"}
{"AtmCode":19,"TotalAmount":242.0,"Time":"2022-01-
16T09:26:56.4920000Z"}
{"AtmCode":15,"TotalAmount":361.0,"Time":"2022-01-
16T09:26:58.5390000Z"}
{"AtmCode":20,"TotalAmount":186.0,"Time":"2022-01-
16T09:26:59.4610000Z"}
{"AtmCode":18,"TotalAmount":106.0,"Time":"2022-01-
16T09:27:00.4820000Z"}
{"AtmCode":2,"TotalAmount":23.0,"Time":"2022-01-
16T09:27:03.7640000Z"}
{"AtmCode":13,"TotalAmount":101.0,"Time":"2022-01-
16T09:27:05.4990000Z"}
{"AtmCode":10,"TotalAmount":215.0,"Time":"2022-01-
16T09:27:07.4680000Z"}
```

```
{"AtmCode":15,"TotalAmount":372.0,"Time":"2022-01-16T09:27:10.5300000Z"}
```

5. Show the total “Amount” of “Type = 1” transactions per “Area Code” of the last hour. Repeat once every hour (use a tumbling window).

➤ Input

```
SELECT
    atm.area_code AS AreaCode,
    sum(input.Amount) AS TotalAmount,
    System.Timestamp AS Time
INTO
    [output5]
FROM
    [input]
JOIN atm ON input.ATMCode = atm.atm_code
WHERE input.Type=1
GROUP BY atm.area_code , TumblingWindow(hour,1);
```

➤ Output

```
{"AreaCode":18,"TotalAmount":25.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":3,"TotalAmount":709.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":11,"TotalAmount":1879.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":5,"TotalAmount":2620.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":10,"TotalAmount":555.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":12,"TotalAmount":36.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":2,"TotalAmount":1740.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":7,"TotalAmount":694.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":15,"TotalAmount":48.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":20,"TotalAmount":42.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":14,"TotalAmount":36.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":6,"TotalAmount":53.0,"Time":"2022-01-16T10:00:00.0000000Z"}
{"AreaCode":1,"TotalAmount":2031.0,"Time":"2022-01-16T10:00:00.0000000Z"}
```

```

16T10:00:00.0000000Z"}
{"AreaCode":17,"TotalAmount":23.0,"Time":"2022-01-
16T10:00:00.0000000Z"}
{"AreaCode":9,"TotalAmount":451.0,"Time":"2022-01-
16T10:00:00.0000000Z"}
{"AreaCode":4,"TotalAmount":1913.0,"Time":"2022-01-
16T10:00:00.0000000Z"}
{"AreaCode":3,"TotalAmount":164.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":11,"TotalAmount":1404.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":5,"TotalAmount":907.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":19,"TotalAmount":26.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":10,"TotalAmount":304.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":12,"TotalAmount":30.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":2,"TotalAmount":780.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":7,"TotalAmount":183.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":1,"TotalAmount":1249.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":9,"TotalAmount":303.0,"Time":"2022-01-
16T11:00:00.0000000Z"}
{"AreaCode":4,"TotalAmount":1047.0,"Time":"2022-01-
16T11:00:00.0000000Z"}

```

6. Show the total “Amount” per ATM’s “City” and Customer’s “Gender” of the last hour. Repeat once every hour (use a tumbling window)

➤ Input

```

SELECT
    area.area_city AS City,
    customer.gender AS Gender,
    sum(input.Amount) AS TotalAmount,
    System.Timestamp AS Time
INTO
    [output6]
FROM
    [input]
JOIN atm ON input.ATMCode = atm.atm_code
JOIN customer ON input.CardNumber = customer.card_number
JOIN area ON customer.area_code = area.area_code
GROUP BY area.area_city , customer.gender ,TumblingWindow(hour,1);

```

➤ Output

```
{ "City": "Vancouver", "Gender": "Female", "TotalAmount": 444.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Dayton", "Gender": "Female", "TotalAmount": 528.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Greeley", "Gender": "Male", "TotalAmount": 458.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Baltimore", "Gender": "Male", "TotalAmount": 589.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Memphis", "Gender": "Male", "TotalAmount": 431.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Greeley", "Gender": "Female", "TotalAmount": 1088.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Springfield", "Gender": "Female", "TotalAmount": 621.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Omaha", "Gender": "Male", "TotalAmount": 989.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Tacoma", "Gender": "Male", "TotalAmount": 475.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Springfield", "Gender": "Male", "TotalAmount": 2342.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Dayton", "Gender": "Male", "TotalAmount": 490.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Tacoma", "Gender": "Female", "TotalAmount": 1469.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Omaha", "Gender": "Female", "TotalAmount": 499.0, "Time": "2022-01-16T12:00:00.0000000Z" }
{ "City": "Vancouver", "Gender": "Female", "TotalAmount": 755.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Dayton", "Gender": "Female", "TotalAmount": 522.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Greeley", "Gender": "Male", "TotalAmount": 281.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Baltimore", "Gender": "Male", "TotalAmount": 526.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Memphis", "Gender": "Male", "TotalAmount": 474.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Greeley", "Gender": "Female", "TotalAmount": 1136.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Springfield", "Gender": "Female", "TotalAmount": 426.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Omaha", "Gender": "Male", "TotalAmount": 1243.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Springfield", "Gender": "Male", "TotalAmount": 2311.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Tacoma", "Gender": "Male", "TotalAmount": 587.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Dayton", "Gender": "Male", "TotalAmount": 458.0, "Time": "2022-01-16T13:00:00.0000000Z" }
{ "City": "Tacoma", "Gender": "Female", "TotalAmount": 1484.0, "Time": "2022-01-16T13:00:00.0000000Z" }
```

```
{"City":"Omaha","Gender":"Female","TotalAmount":577.0,"Time":"2022-01-16T13:00:00.0000000Z"}
```

7. Alert (Do a simple SELECT “1”) if a Customer has performed two transactions of “Type = 1” in a window of an hour (use a sliding window).
-

➤ **Input**

```
SELECT 1 AS Alert,  
customer.last_name AS Surname,  
System.Timestamp AS Time  
INTO  
    [output7]  
FROM input  
TIMESTAMP BY input.EventEnqueuedUtcTime  
JOIN customer  
ON customer.card_number = input.CardNumber  
GROUP BY input.Type, customer.last_name, SlidingWindow(hour, 1)  
HAVING input.Type=1 AND COUNT(*)=2
```

➤ **Output**

```
{"Alert":1,"Surname":"Jordan","Time":"2022-01-16T13:35:11.0090000Z"}  
{"Alert":1,"Surname":"Snyder","Time":"2022-01-16T13:35:13.3370000Z"}  
{"Alert":1,"Surname":"Fuller","Time":"2022-01-16T13:35:19.4930000Z"}  
{"Alert":1,"Surname":"Young","Time":"2022-01-16T13:35:27.4940000Z"}  
{"Alert":1,"Surname":"Stone","Time":"2022-01-16T13:35:28.6810000Z"}  
{"Alert":1,"Surname":"Perry","Time":"2022-01-16T13:35:31.4800000Z"}  
{"Alert":1,"Surname":"Russell","Time":"2022-01-16T13:35:48.4870000Z"}  
{"Alert":1,"Surname":"Day","Time":"2022-01-16T13:36:19.4850000Z"}  
{"Alert":1,"Surname":"Perez","Time":"2022-01-16T13:36:21.5790000Z"}  
{"Alert":1,"Surname":"Carroll","Time":"2022-01-16T13:36:28.5340000Z"}  
{"Alert":1,"Surname":"Moreno","Time":"2022-01-16T13:36:40.4670000Z"}  
{"Alert":1,"Surname":"Bradley","Time":"2022-01-16T13:37:05.4760000Z"}  
{"Alert":1,"Surname":"Cooper","Time":"2022-01-16T13:37:32.5150000Z"}  
{"Alert":1,"Surname":"Mason","Time":"2022-01-16T13:38:01.6150000Z"}
```

8. Alert (Do a simple SELECT “1”) if the “Area Code” of the ATM of the transaction is not the same as the “Area Code” of the “Card Number” (Customer’s Area Code) - (use a sliding window)
-

➤ **Input**

```

SELECT
atm.area_code AS AtmAreaCode,
customer.area_code AS CustomerAreaCode,
COUNT (*),
System.Timestamp AS Time
INTO [output8]
FROM [input]
INNER JOIN customer ON customer.card_number = input.CardNumber INNER
JOIN atm
ON atm.atm_code = input.ATMCode
WHERE atm.area_code != customer.area_code
GROUP BY atm.area_code,
customer.area_code,
SlidingWindow(hour, 1)

```

➤ Output

```

{"AtmAreaCode":10,"CustomerAreaCode":6,"COUNT":1,"Time":"2022-01-16T09:25:50.0420000Z"}
{"AtmAreaCode":1,"CustomerAreaCode":6,"COUNT":1,"Time":"2022-01-16T09:25:50.5410000Z"}
{"AtmAreaCode":1,"CustomerAreaCode":6,"COUNT":2,"Time":"2022-01-16T09:25:51.8220000Z"}
{"AtmAreaCode":2,"CustomerAreaCode":1,"COUNT":1,"Time":"2022-01-16T09:25:52.5410000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":1,"Time":"2022-01-16T09:25:53.6350000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":1,"Time":"2022-01-16T09:25:54.6040000Z"}
{"AtmAreaCode":2,"CustomerAreaCode":1,"COUNT":2,"Time":"2022-01-16T09:25:55.6670000Z"}
{"AtmAreaCode":4,"CustomerAreaCode":2,"COUNT":1,"Time":"2022-01-16T09:25:57.5100000Z"}
{"AtmAreaCode":2,"CustomerAreaCode":1,"COUNT":3,"Time":"2022-01-16T09:25:58.5260000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":2,"Time":"2022-01-16T09:25:59.4790000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":3,"Time":"2022-01-16T09:26:00.5100000Z"}
{"AtmAreaCode":2,"CustomerAreaCode":1,"COUNT":4,"Time":"2022-01-16T09:26:01.5110000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":2,"Time":"2022-01-16T09:26:03.4640000Z"}
{"AtmAreaCode":4,"CustomerAreaCode":2,"COUNT":2,"Time":"2022-01-16T09:26:04.4790000Z"}
{"AtmAreaCode":3,"CustomerAreaCode":4,"COUNT":1,"Time":"2022-01-16T09:26:05.5260000Z"}
{"AtmAreaCode":9,"CustomerAreaCode":10,"COUNT":1,"Time":"2022-01-16T09:26:09.4800000Z"}
{"AtmAreaCode":2,"CustomerAreaCode":1,"COUNT":5,"Time":"2022-01-16T09:26:10.5260000Z"}

```



```
16T09:26:12.6830000Z"}
{"AtmAreaCode":10,"CustomerAreaCode":6,"COUNT":2,"Time":"2022-01-
16T09:26:13.6990000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":4,"Time":"2022-01-
16T09:26:14.5110000Z"}
{"AtmAreaCode":4,"CustomerAreaCode":2,"COUNT":3,"Time":"2022-01-
16T09:26:15.5270000Z"}
{"AtmAreaCode":4,"CustomerAreaCode":2,"COUNT":4,"Time":"2022-01-
16T09:26:16.5580000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":5,"Time":"2022-01-
16T09:26:17.4800000Z"}
{"AtmAreaCode":7,"CustomerAreaCode":3,"COUNT":1,"Time":"2022-01-
16T09:26:18.4960000Z"}
{"AtmAreaCode":9,"CustomerAreaCode":10,"COUNT":2,"Time":"2022-01-
16T09:26:19.4640000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":3,"Time":"2022-01-
16T09:26:20.4800000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":6,"Time":"2022-01-
16T09:26:21.4490000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":4,"Time":"2022-01-
16T09:26:22.5120000Z"}
{"AtmAreaCode":11,"CustomerAreaCode":8,"COUNT":7,"Time":"2022-01-
16T09:26:23.5150000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":5,"Time":"2022-01-
16T09:26:24.4690000Z"}
{"AtmAreaCode":5,"CustomerAreaCode":7,"COUNT":6,"Time":"2022-01-
16T09:26:25.4850000Z"}
```