# Query Data Using Azure Stream Analytics

**Reza Salehi**

CLOUD CONSULTANT

@zaalion linkedin.com/in/rezasalehi2008

# Overview

**Supported input formats for Azure Stream Analytics**

**How is the input mapped to the output?**

**Introducing Stream Analytics Query Language**

- Data types, language elements, and built-in functions

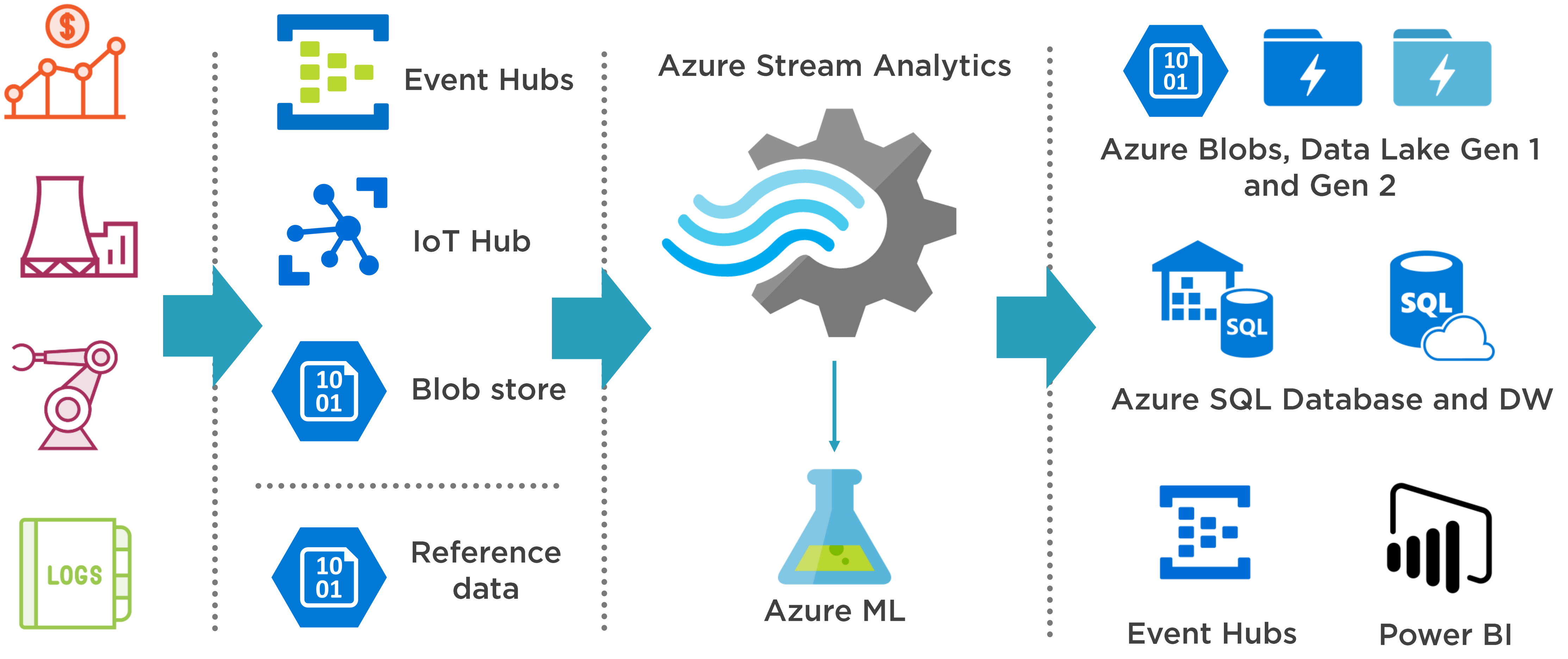
**More on data stream event timestamps**

- Event time vs. arrival time
- Event ordering policies

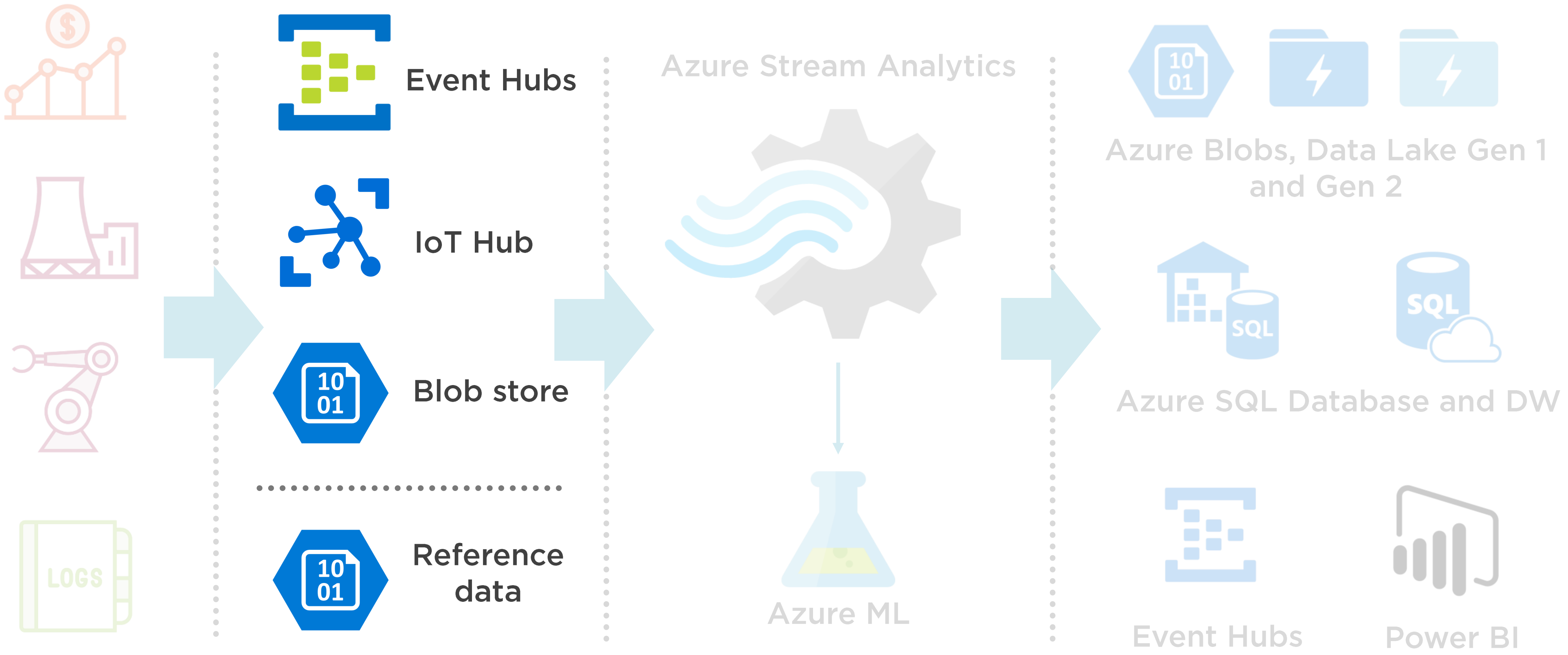
**Demo:**

- Stream Analytics Query Language
- Event ordering policies

# Azure Stream Analytics Data Flow



# Azure Stream Analytics Inputs



Event Hubs

IoT Hub

Blob store

Reference data

Azure Stream Analytics

Azure ML

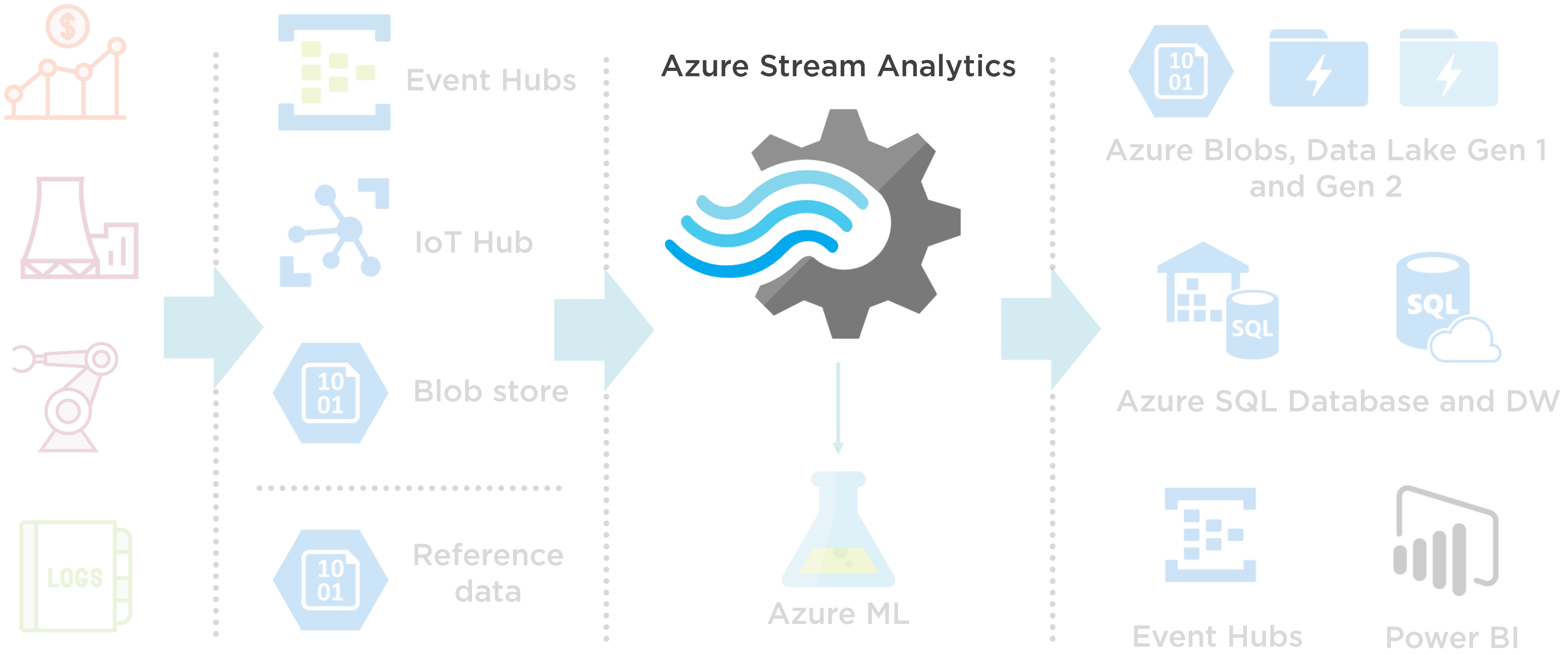
Azure Blobs, Data Lake Gen 1 and Gen 2

Azure SQL Database and DW

Event Hubs

Power BI

# Azure Stream Analytics Inputs



# Data Stream Inputs

Azure Event Hubs

Azure IoT Hub

Azure Blob storage

Which data formats are supported as input data streams?

# Data Stream Inputs

CSV

{JSON}

AVRO

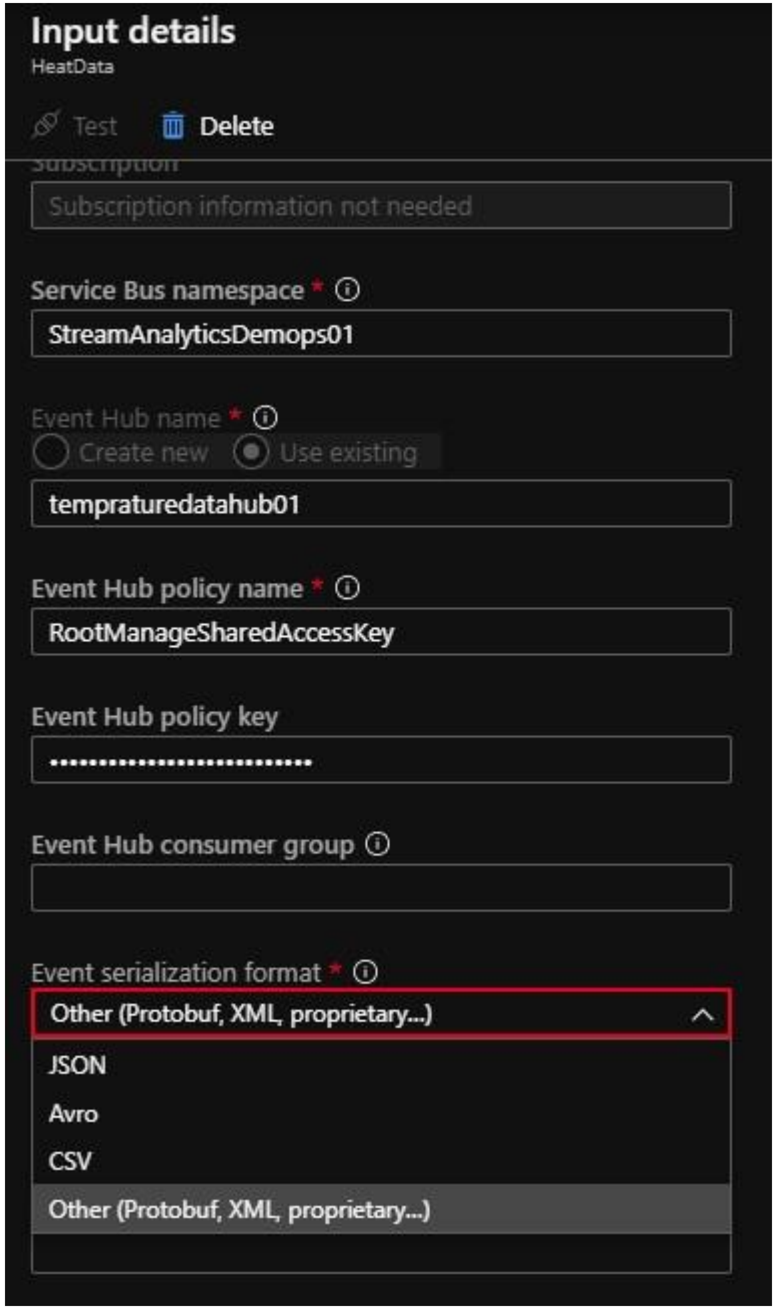
# Apache Avro

Data serialization framework

Developed within Apache's Hadoop project

It uses JSON for defining schema

Serializes data in binary format



## Input details

HeatData

Test Delete

Subscription

Subscription information not needed

Service Bus namespace * ⓘ

StreamAnalyticsDemops01

Event Hub name * ⓘ

○ Create new ● Use existing

tempraturedatahub01

Event Hub policy policy name * ⓘ

RootManageSharedAccessKey

Event Hub policy key

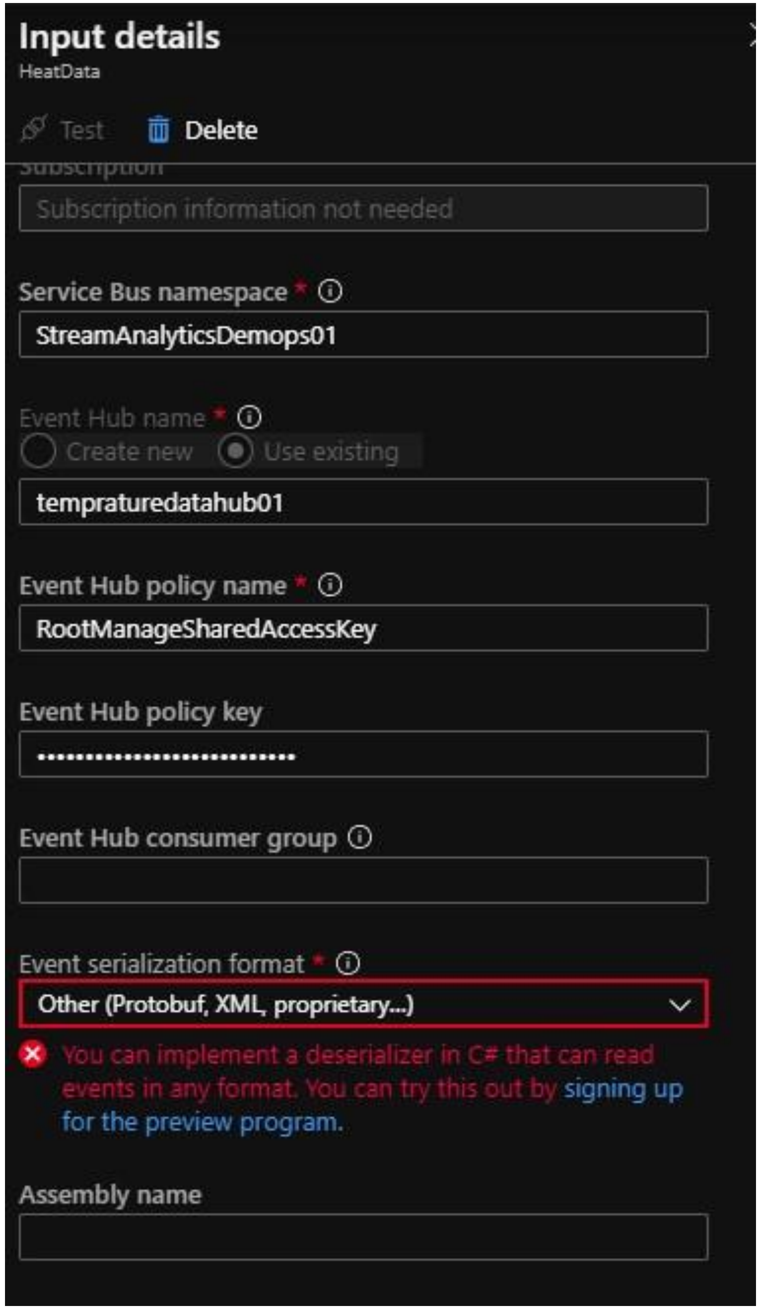
••••••••••••••••••••••••••••

Event Hub consumer group ⓘ

Event serialization format * ⓘ

Other (Protobuf, XML, proprietary...)

- JSON
- Avro
- CSV
- Other (Protobuf, XML, proprietary...)



## Input details

HeatData

Test Delete

Subscription

Subscription information not needed

Service Bus namespace * ⓘ

StreamAnalyticsDemops01

Event Hub name * ⓘ

○ Create new ● Use existing

tempraturedatahub01

Event Hub policy name * ⓘ

RootManageSharedAccessKey

Event Hub policy key

••••••••••••••••••••••••••••

Event Hub consumer group ⓘ

Event serialization format * ⓘ

Other (Protobuf, XML, proprietary...)

✖ You can implement a deserializer in C# that can read events in any format. You can try this out by signing up for the preview program.

Assembly name

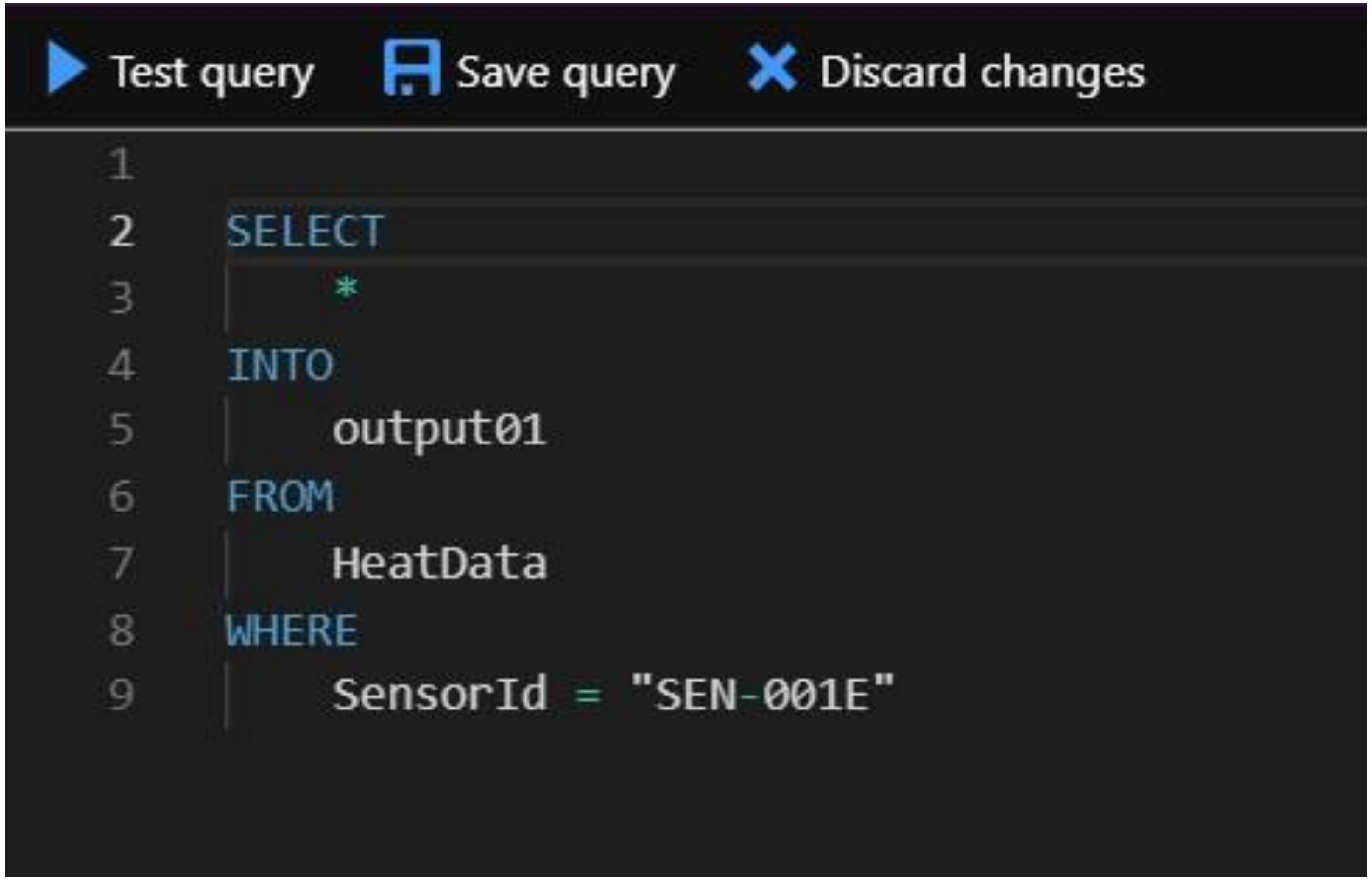
In this course,
we are using JSON.

# Stream Analytics Query Language

# Stream Analytics Query Language

Is a T-SQL like language which maps the input stream(s) into the Stream Analytics output.

# Stream Analytics Query Language



```

SELECT
    *
INTO
    output01
FROM
    HeatData
WHERE
    SensorId = "SEN-001E"
```

You almost know this language if you’re familiar with T-SQL.

# Stream Analytics Query Language

**Data Types**

Nvarchar, bit, float, etc.

**Language Elements**

Select, where, join, etc.

**Built-in Functions**

Avg, count, max, etc.

# Stream Analytics Supported Data Types

Float and bigint

Nvarchar(max)

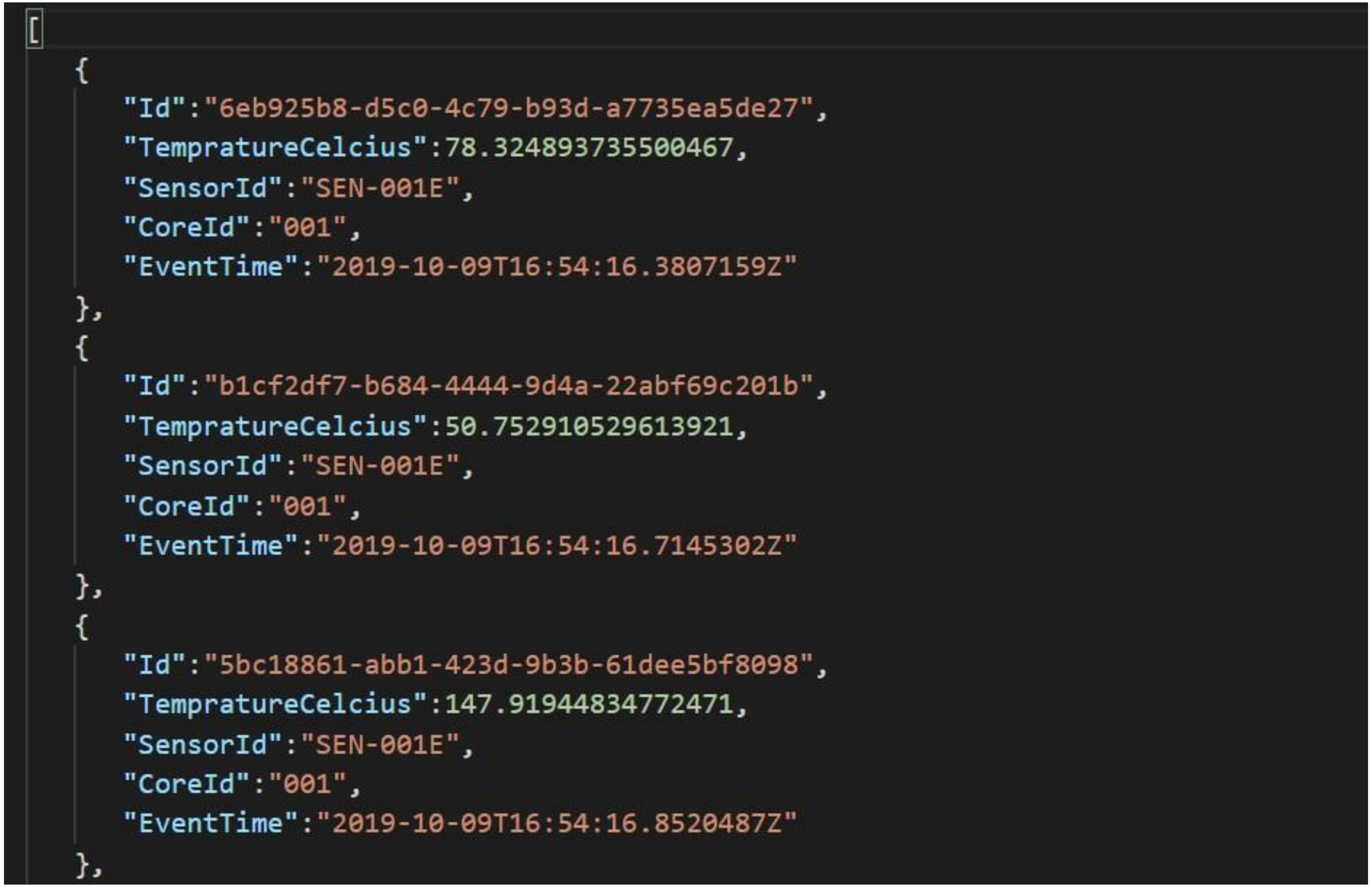
Record

Bit

Datetime

Array

# Stream Analytics Query Language



```
[
   {
      "Id":"6eb925b8-d5c0-4c79-b93d-a7735ea5de27",
      "TempratureCelcius":78.324893735500467,
      "SensorId":"SEN-001E",
      "CoreId":"001",
      "EventTime":"2019-10-09T16:54:16.3807159Z"
   },
   {
      "Id":"b1cf2df7-b684-4444-9d4a-22abf69c201b",
      "TempratureCelcius":50.752910529613921,
      "SensorId":"SEN-001E",
      "CoreId":"001",
      "EventTime":"2019-10-09T16:54:16.7145302Z"
   },
   {
      "Id":"5bc18861-abb1-423d-9b3b-61dee5bf8098",
      "TempratureCelcius":147.91944834772471,
      "SensorId":"SEN-001E",
      "CoreId":"001",
      "EventTime":"2019-10-09T16:54:16.8520487Z"
   },
```

# Type Conversions

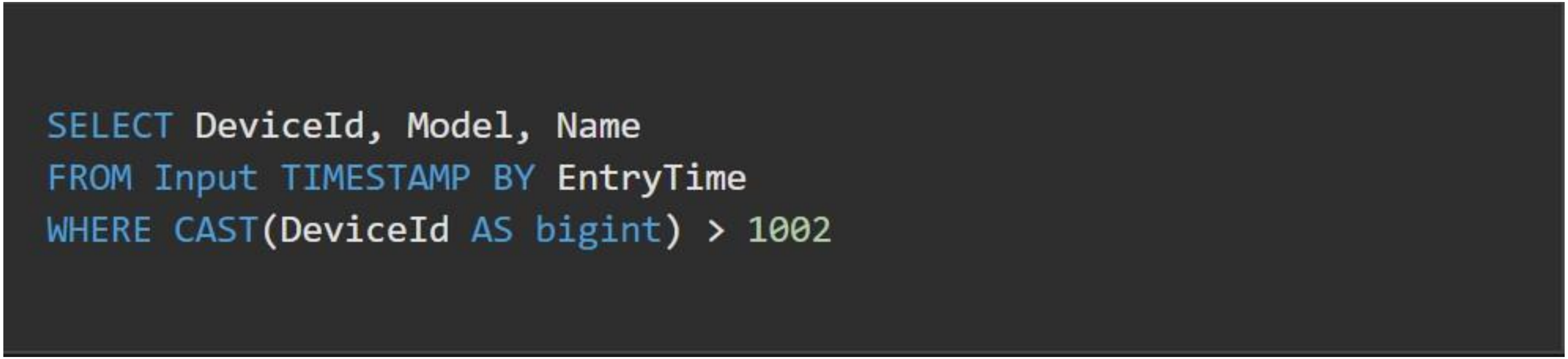
## Type casting functions

CAST(), TRY_CAST(), GetType()

## Type casting errors

Can happen during input read or output write

# Stream Analytics Query Language



```
SELECT DeviceId, Model, Name
FROM Input TIMESTAMP BY EntryTime
WHERE CAST(DeviceId AS bigint) > 1002
```

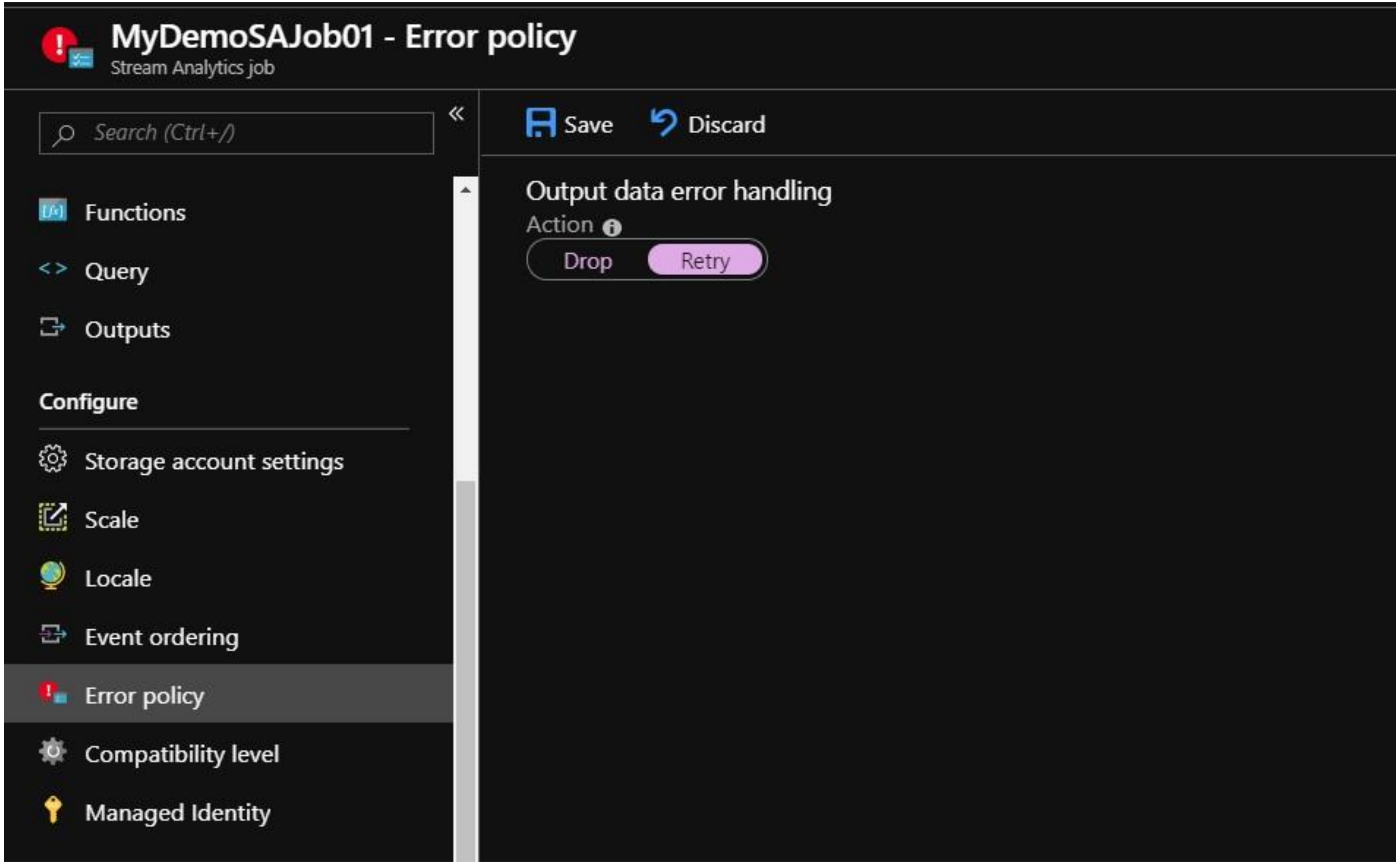
# Type Casting Errors

Type conversion errors happen during the input read, will cause the job to drop the event

Type conversion errors happen during the output write are handled by the error policy

You can set the error policy to drop or retry

# Error Policy



Output data error handling

Action

Drop [✓ Retry]

# Stream Analytics Query Language Elements

Select, from, over where, union, into

Group by, having, join, with, case, coalesce

Into

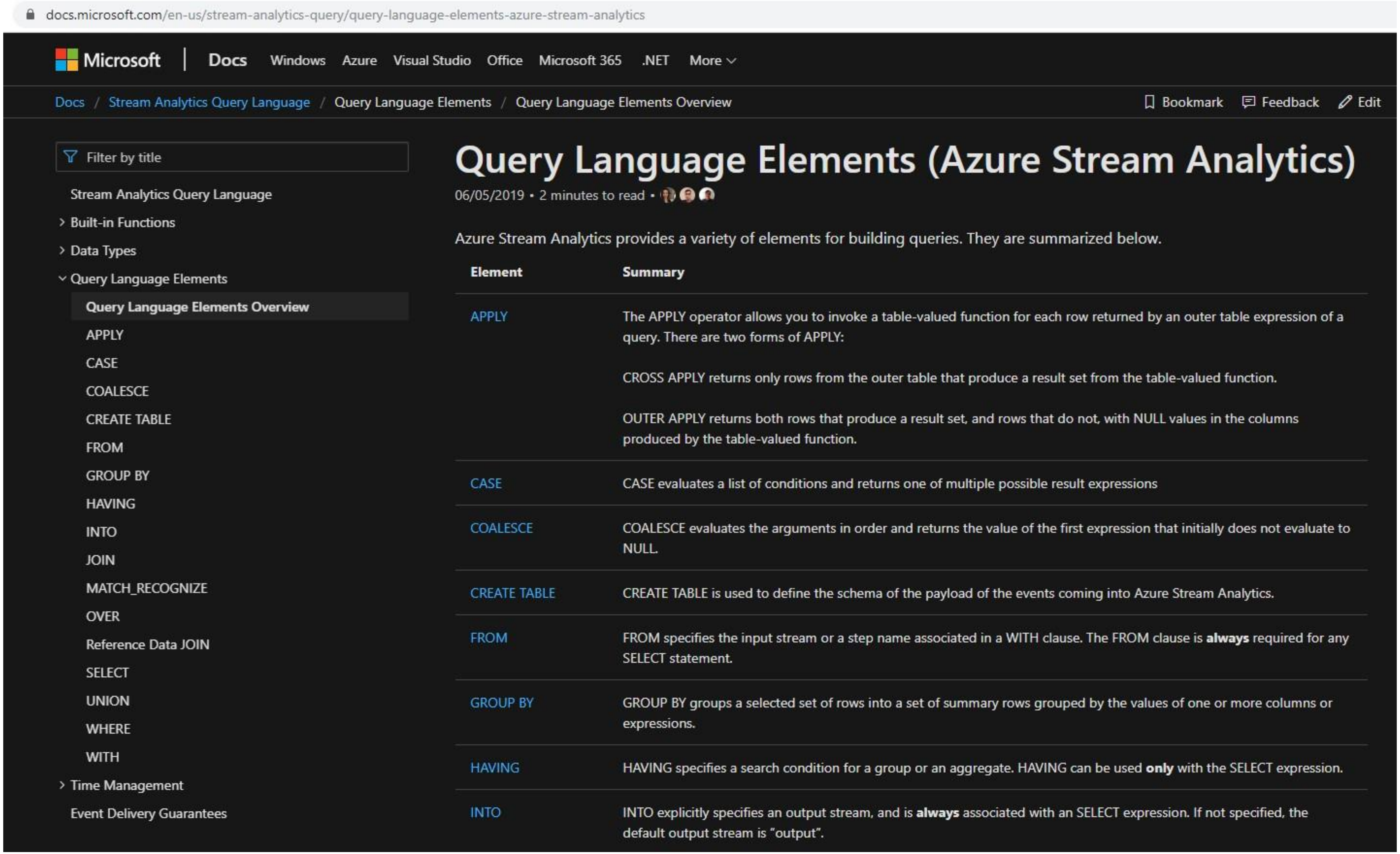
Specifies an output stream

Apply

Invokes a function for each data row

Create table

Defines the schema of the input payloads


docs.microsoft.com/en-us/stream-analytics-query/query-language-elements-azure-stream-analytics
Microsoft
Docs
Windows
Azure
Visual Studio
Office
Microsoft 365
.NET
More
Docs / Stream Analytics Query Language / Query Language Elements / Query Language Elements Overview
Bookmark
Feedback
Edit
Filter by title
Stream Analytics Query Language
Built-in Functions
Data Types
Query Language Elements
Query Language Elements Overview
APPLY
CASE
COALESCE
CREATE TABLE
FROM
GROUP BY
HAVING
INTO
JOIN
MATCH_RECOGNIZE
OVER
Reference Data JOIN
SELECT
UNION
WHERE
WITH
Time Management
Event Delivery Guarantees

# Query Language Elements (Azure Stream Analytics)

06/05/2019 • 2 minutes to read •

Azure Stream Analytics provides a variety of elements for building queries. They are summarized below.

| Element | Summary |
|---|---|
| APPLY | The APPLY operator allows you to invoke a table-valued function for each row returned by an outer table expression of a query. There are two forms of APPLY:<br>CROSS APPLY returns only rows from the outer table that produce a result set from the table-valued function.<br>OUTER APPLY returns both rows that produce a result set, and rows that do not, with NULL values in the columns produced by the table-valued function. |
| CASE | CASE evaluates a list of conditions and returns one of multiple possible result expressions |
| COALESCE | COALESCE evaluates the arguments in order and returns the value of the first expression that initially does not evaluate to NULL. |
| CREATE TABLE | CREATE TABLE is used to define the schema of the payload of the events coming into Azure Stream Analytics. |
| FROM | FROM specifies the input stream or a step name associated in a WITH clause. The FROM clause is **always** required for any SELECT statement. |
| GROUP BY | GROUP BY groups a selected set of rows into a set of summary rows grouped by the values of one or more columns or expressions. |
| HAVING | HAVING specifies a search condition for a group or an aggregate. HAVING can be used **only** with the SELECT expression. |
| INTO | INTO explicitly specifies an output stream, and is **always** associated with an SELECT expression. If not specified, the default output stream is "output". |

# Query Language Built-in Functions

Aggregate functions

Array functions

Input Metadata functions

Analytic functions

GeoSpatial functions

Record functions

# Query Language Built-in Functions

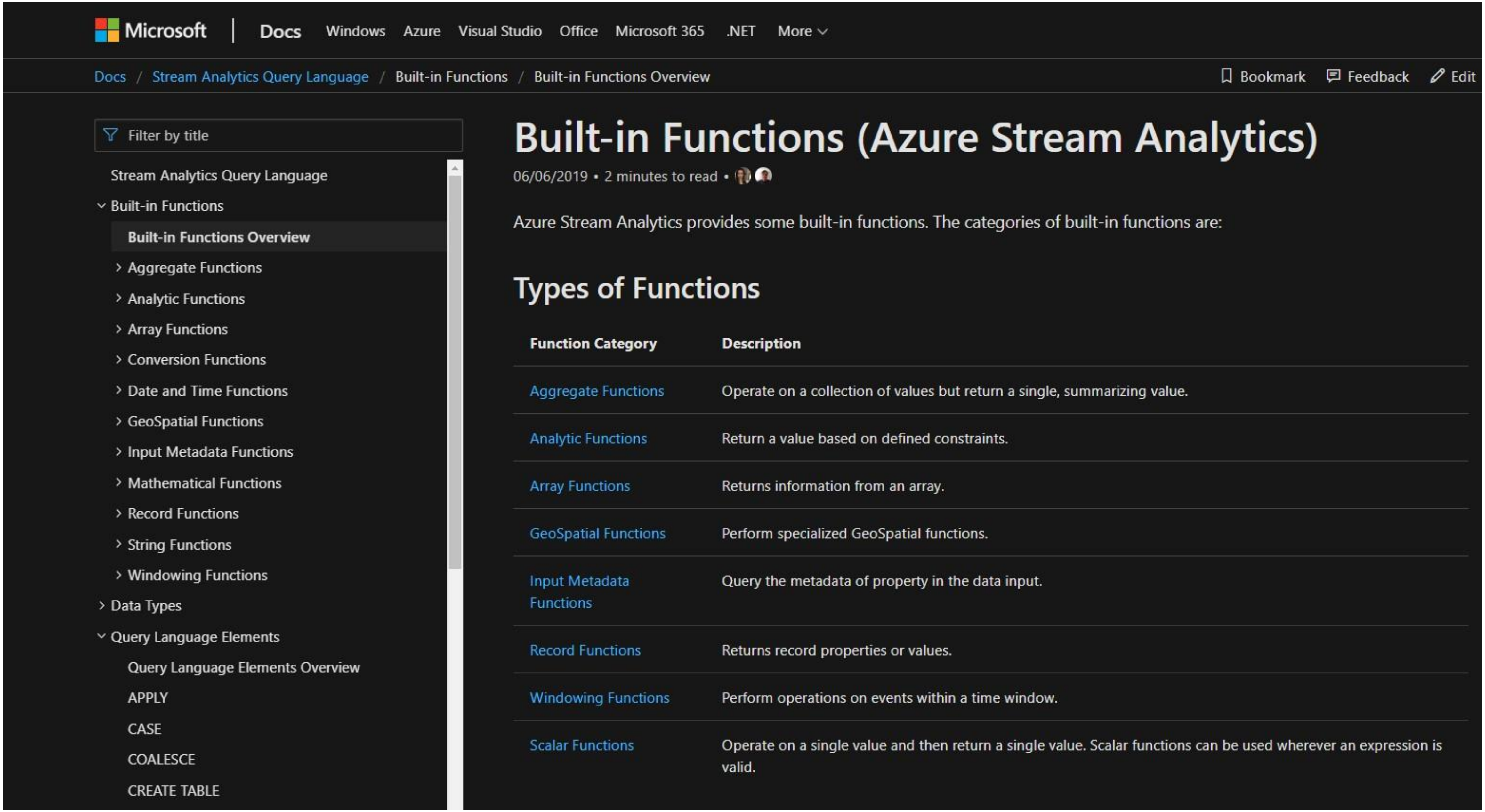
Windowing functions

Conversion functions

Date and Time functions

Mathematical functions

String functions



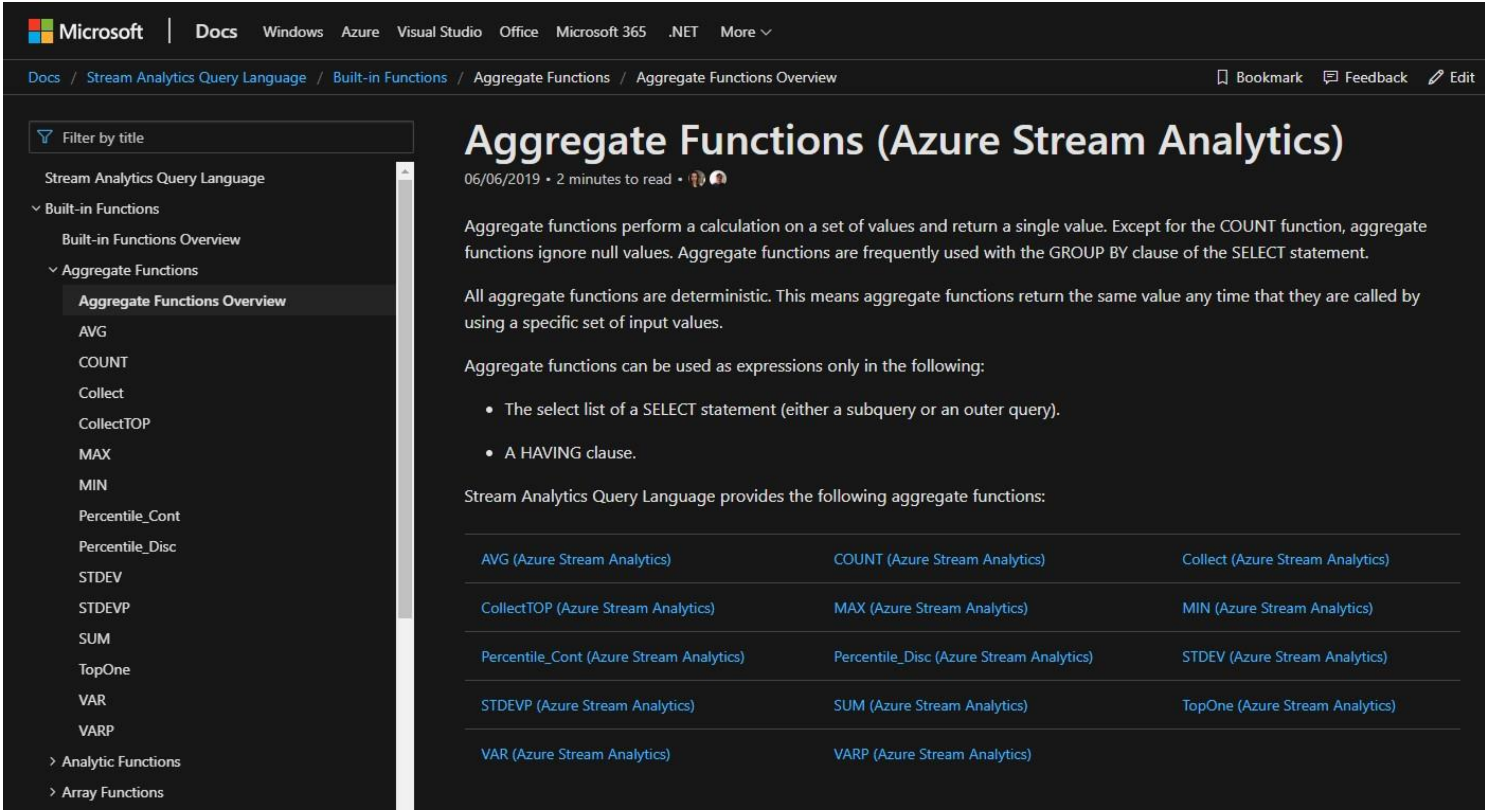
# Built-in Functions (Azure Stream Analytics)

06/06/2019 • 2 minutes to read •

Azure Stream Analytics provides some built-in functions. The categories of built-in functions are:

## Types of Functions

| Function Category | Description |
|---|---|
| Aggregate Functions | Operate on a collection of values but return a single, summarizing value. |
| Analytic Functions | Return a value based on defined constraints. |
| Array Functions | Returns information from an array. |
| GeoSpatial Functions | Perform specialized GeoSpatial functions. |
| Input Metadata Functions | Query the metadata of property in the data input. |
| Record Functions | Returns record properties or values. |
| Windowing Functions | Perform operations on events within a time window. |
| Scalar Functions | Operate on a single value and then return a single value. Scalar functions can be used wherever an expression is valid. |



# Aggregate Functions (Azure Stream Analytics)

06/06/2019 • 2 minutes to read •

Aggregate functions perform a calculation on a set of values and return a single value. Except for the COUNT function, aggregate functions ignore null values. Aggregate functions are frequently used with the GROUP BY clause of the SELECT statement.

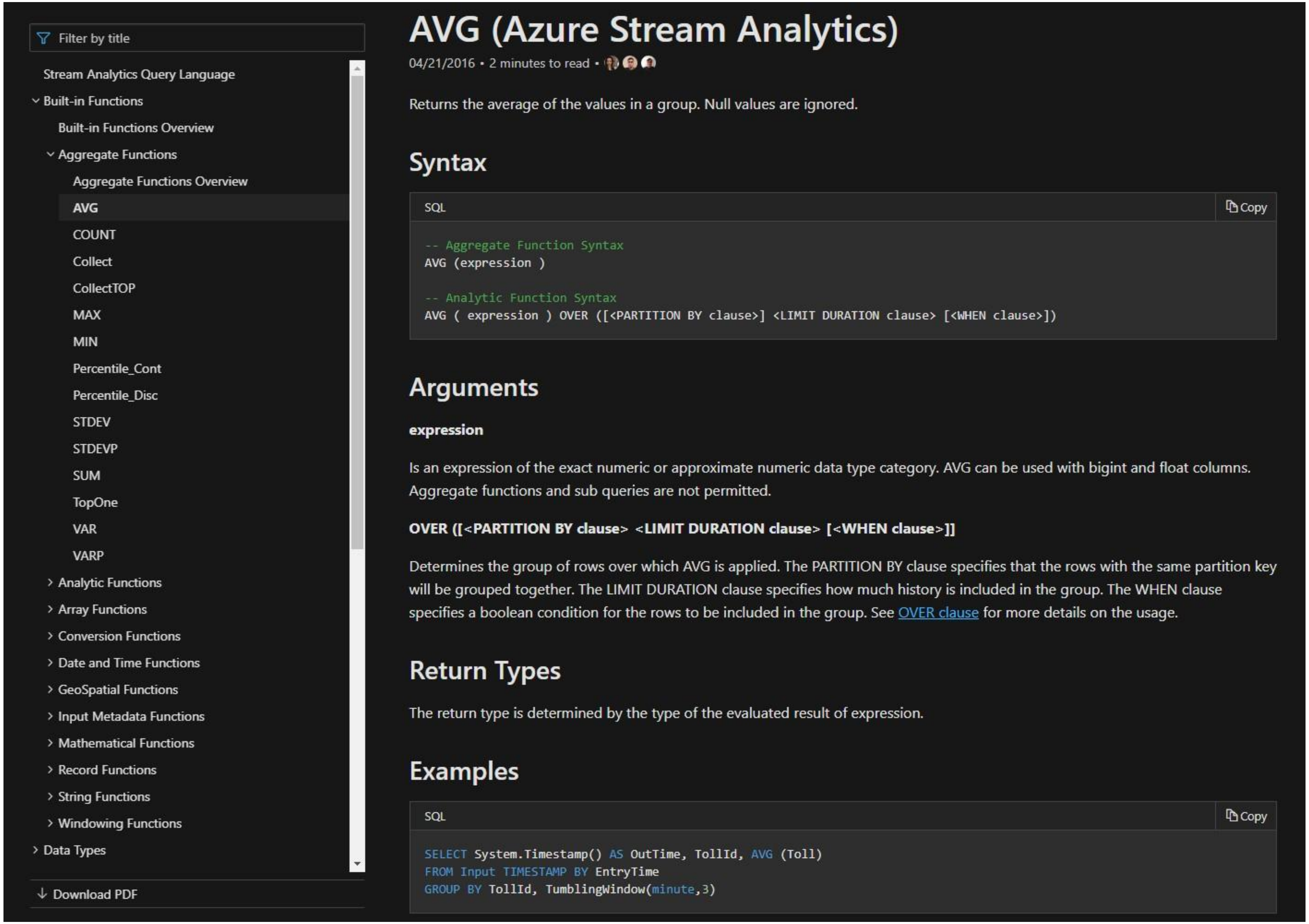
All aggregate functions are deterministic. This means aggregate functions return the same value any time that they are called by using a specific set of input values.

Aggregate functions can be used as expressions only in the following:

- The select list of a SELECT statement (either a subquery or an outer query).
- A HAVING clause.

Stream Analytics Query Language provides the following aggregate functions:

| | | |
|---|---|---|
| AVG (Azure Stream Analytics) | COUNT (Azure Stream Analytics) | Collect (Azure Stream Analytics) |
| CollectTOP (Azure Stream Analytics) | MAX (Azure Stream Analytics) | MIN (Azure Stream Analytics) |
| Percentile_Cont (Azure Stream Analytics) | Percentile_Disc (Azure Stream Analytics) | STDEV (Azure Stream Analytics) |
| STDEVP (Azure Stream Analytics) | SUM (Azure Stream Analytics) | TopOne (Azure Stream Analytics) |
| VAR (Azure Stream Analytics) | VARP (Azure Stream Analytics) | |



# AVG (Azure Stream Analytics)

04/21/2016 • 2 minutes to read •

Returns the average of the values in a group. Null values are ignored.

## Syntax

SQL

Copy

```sql
-- Aggregate Function Syntax
AVG (expression )

-- Analytic Function Syntax
AVG ( expression ) OVER ([<PARTITION BY clause>] <LIMIT DURATION clause> [<WHEN clause>])
```

## Arguments

**expression**

Is an expression of the exact numeric or approximate numeric data type category. AVG can be used with bigint and float columns. Aggregate functions and sub queries are not permitted.

**OVER ([<PARTITION BY clause> <LIMIT DURATION clause> [<WHEN clause>]]**

Determines the group of rows over which AVG is applied. The PARTITION BY clause specifies that the rows with the same partition key will be grouped together. The LIMIT DURATION clause specifies how much history is included in the group. The WHEN clause specifies a boolean condition for the rows to be included in the group. See OVER clause for more details on the usage.

## Return Types

The return type is determined by the type of the evaluated result of expression.

## Examples

SQL

Copy

```sql
SELECT System.Timestamp() AS OutTime, TollId, AVG (Toll)
FROM Input TIMESTAMP BY EntryTime
GROUP BY TollId, TumblingWindow(minute,3)
```

# Stream Analytics Timing

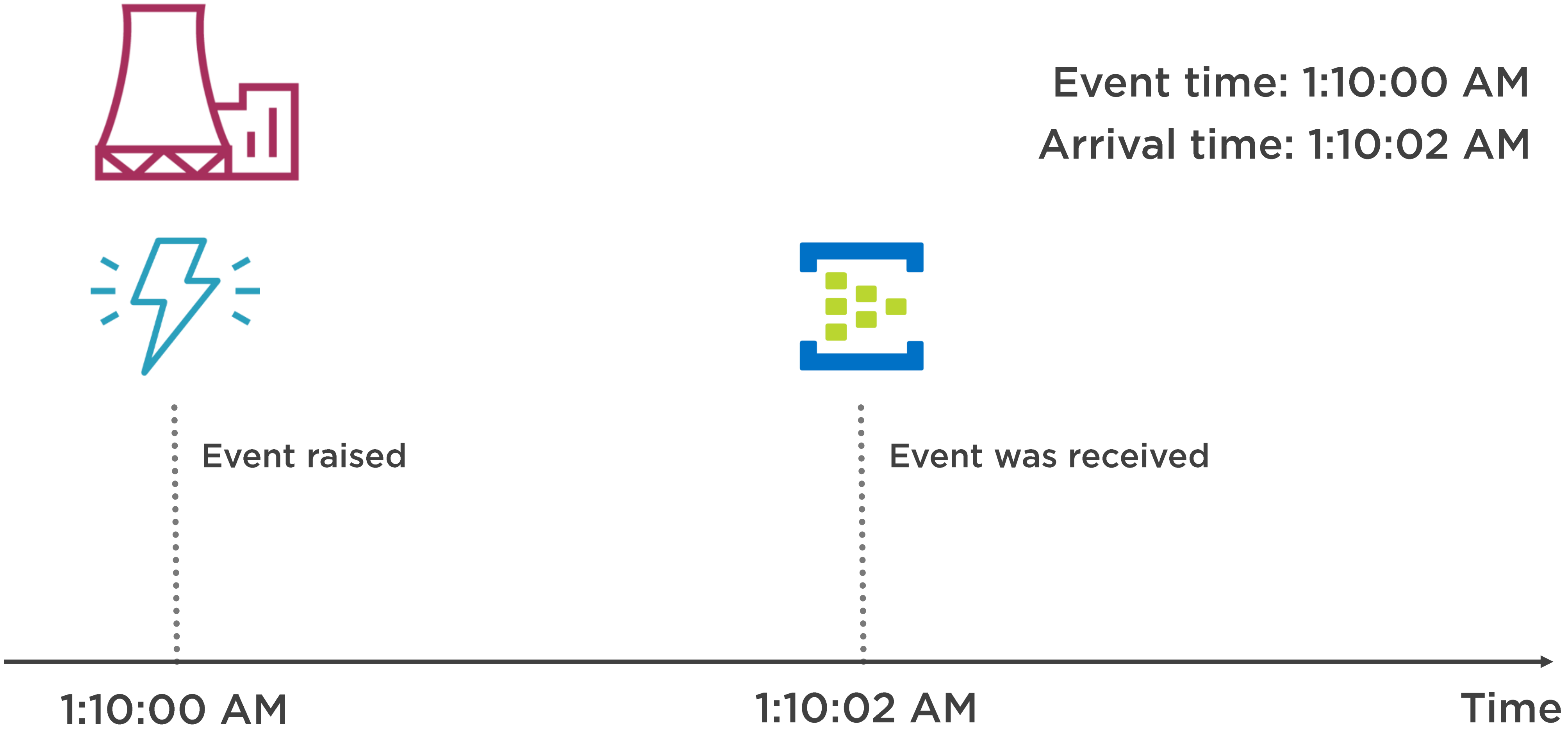
# Stream Analytics Timing

Arrival time and event time

Event ordering policies

All data stream events have a timestamp.

# Arrival Time and Event Time



Event time: 1:10:00 AM

Arrival time: 1:10:02 AM

Event raised

Event was received

1:10:00 AM

1:10:02 AM

Time

# Arrival Time and Event Time

- Arrival time for Event Hubs (and IoT Hub) events is when the event was received

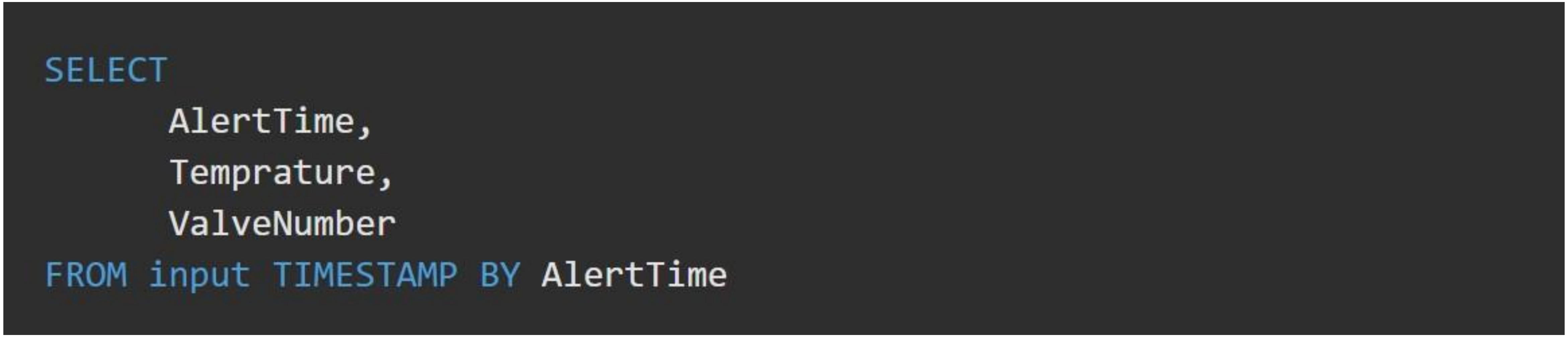
- Arrival time for Blob storage events is the last modified time

- By default, Azure Stream Analytics uses the arrival time as timestamp

- You might need to use event time instead of arrival time

Using the TIMESTAMP BY clause, you can specify custom timestamp values.

# Using Event Time as Timestamp



```
SELECT
      AlertTime,
      Temprature,
      ValveNumber
FROM input TIMESTAMP BY AlertTime
```

# Event Ordering Policies

Azure Stream Analytics might receive late arrival or out of order events

Can be caused by using TIMESTAMP BY

Or in the case of multiple producers, they have clock skews

Or network latency

# Event Ordering Policies

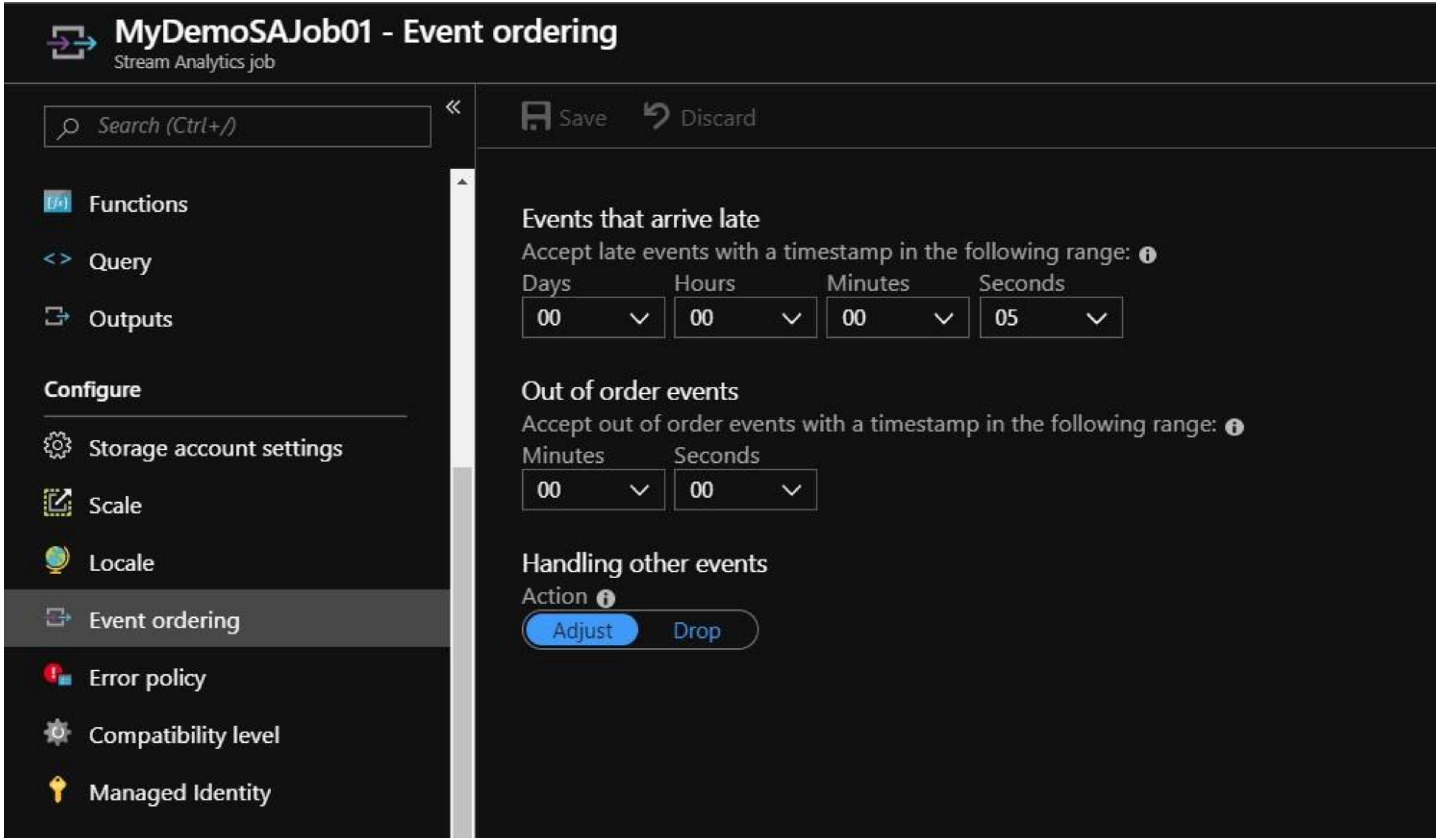
## Late arrival policy

Accept events within the policy window, adjust or drop the rest

## Out of order policy

Reorder events that arrive out of order but within the policy window, adjust or drop rest

# Event Ordering Policies



MyDemoSAJob01 - Event ordering

Stream Analytics job

Events that arrive late

Accept late events with a timestamp in the following range:

| Days | Hours | Minutes | Seconds |
|---|---|---|---|
| 00 | 00 | 00 | 05 |

Out of order events

Accept out of order events with a timestamp in the following range:

| Minutes | Seconds |
|---|---|
| 00 | 00 |

Handling other events

Action

[✓ Adjust] Drop

Event ordering policies are applied only if TIMESTAMP BY is used.

Demo

## Stream Analytics Query Language

- Data types
- Joins
- Functions

## Configuring event ordering policies

# Summary

**Supported input formats for Azure Stream Analytics**

**The input is mapped to the output with Stream Analytics Query Language**

- Data types, language elements, and built-in functions

**Data stream event timestamps**

- Event time vs. arrival time
- Event ordering policies

**Demo:**

- Stream Analytics Query Language
- Event ordering policies