**MODULE – 5**

Eventstreams -> To capture and process real time events without needing to write any code.

Setup event sources, destinations, transformations in event stream

Components of eventstreams:

Work by creating pipeline of events from multiple internal and external sources to different destinations. You can also add some transformations (filtering/ aggregating/ enriching)

Nodes 🡪 **Sources**, **Destination**, **Transformations**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sources | |  |  | | --- | --- | | Event Hub/ IoT Hub | SQL DB CDC | | PostgreSQL DB CDC | MySQL DB CDC | | Cosmos DB CDC | GCP pub/sub | | AWS Kinesis data streams | Cloud kafka | | Fabric workspace events | Blob storage | | Custom endpoint | Sample data | |
| Destinations | |  |  | | --- | --- | | Eventhouse | Lakehouse | | Custom endpoint | Derived stream | | Fabric activator |  | |
| Transformations | * Filter * Manage fields * Aggregate * Group by * Union * Expand * Join |

* Window duration: Length of each window interval, can be sec, min, hr, days

Eg: duration=10min 🡪 each window covers 10mins of event data.

* Window offset: Optional. Shifts start and end of each window interval by a specified amount of time.

Eg: window offset=2mins 🡪 each window starts and ends 2mins later than usual

* Grouping key: one or more cols in your event data that you wish to group by

Eg: by sensor ID, or item category

* Aggregation function: one or more of funcs you want to apply to each group of events in each window. Where the counts, sums, avgs, min, max and even custom functions become useful.

Windowing Functions in Eventstream:

To perform operations on the data in temporal windows 🡪 aggregating/ filtering/ transforming streaming events that occur within a specified time period.

Useful for analyzing streaming data that changes over time 🡪 sensor readings, web clicks, on-line transactions etc.

**Window types:**

### Tumbling window:

Divides incoming events into fixed and nonoverlapping intervals based on arrival time. This window shows a count of tweets per time zone every 10 seconds apart.

A screenshot of a computer

AI-generated content may be incorrect.

### Sliding window

Takes the events into fixed and overlapping intervals based on time and divides them.

Window is a 10-sec sliding window that alerts the user whenever an article is mentioned more than three times in under 10sec

A screenshot of a computer screen

AI-generated content may be incorrect.

### Snapshot window

Group event stream events that have the same timestamp and are unlike the other windowing functions, which require the function to be named. In snapsot windows you add the **System.timestamp()** to the **GROUP BY** clause.

The window provides a count of tweets with the same article type that occur at exactly the same time.

A screenshot of a computer

AI-generated content may be incorrect.

### Hopping window

Different from tumbling window as they model scheduled overlapping window.

A 10 sec hopping window, which refreshes every 5 sec and provides the total count of tweets over the past 10 seconds.

A diagram of a project

AI-generated content may be incorrect.

### Session window

Simply divides events into variable and non-overlapping intervals that are based on a gap of lack of activity.

A window, which shows the total count of tweets that occur within 5mins of one another

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.