**Time Attribute vs Timestamp in Flink**

In Flink, **Time Attributes** and **Timestamps** are two important concepts used for event time processing. They determine how Flink processes and assigns time to events.

**1. Time Attribute in Flink**

* A **Time Attribute** is a special field in a Flink table that represents the **time basis** for event processing.
* It can be either **Event Time** or **Processing Time**.

**Types of Time Attributes:**

1. **Event Time Attribute**
   * Extracted from the event itself.
   * Requires **watermarks** to handle late events.

**Example:**

CREATE TABLE orders (

order\_id STRING,

event\_time TIMESTAMP(3),

WATERMARK FOR event\_time AS event\_time - INTERVAL '5' SECONDS

) WITH (...);

* + Here, event\_time is the **Time Attribute** for Event Time processing.

1. **Processing Time Attribute**
   * Uses Flink’s **system clock** for real-time processing.

**Example:**

CREATE TABLE orders (

order\_id STRING,

event\_time TIMESTAMP(3),

processing\_time AS PROCTIME()

) WITH (...);

* + processing\_time is the **Time Attribute** for Processing Time.

**2. Timestamp in Flink**

* A **Timestamp** is a normal column that stores time-related data.
* It **does not automatically drive time-based operations** like windowing or watermarks.

**Example of a Regular Timestamp Column**

CREATE TABLE orders (

order\_id STRING,

order\_timestamp TIMESTAMP(3)

) WITH (...);

* order\_timestamp **does not act as a Time Attribute** unless explicitly defined as one.

**Key Differences Between Time Attribute and Timestamp**

| **Feature** | **Time Attribute** | **Timestamp Column** |
| --- | --- | --- |
| **Purpose** | Drives event time or processing time logic | Just stores timestamp values |
| **Used in Watermarks?** | Yes, for event time processing | No, unless converted to a time attribute |
| **Used in Windowing?** | Yes, required for defining windows | No, unless explicitly assigned |
| **Example Use Case** | Tumbling windows, Sliding windows | Storing order creation time |

**Which One to Use?**

* Use **Time Attributes** when defining **event time processing and windowing**.
* Use **Timestamps** when you just need to **store time-related data** without time-based processing.

In **TIMESTAMP(p)**, the **p** represents the **precision** of the timestamp, which defines the number of **fractional seconds (decimal places) stored in the timestamp**.

**Precision (p) in TIMESTAMP(p)**

* p can range from **0 to 9**.
* It controls how many digits are stored after the **decimal point** in the seconds field.
* The default precision is **6** if not specified.

**Examples of TIMESTAMP(p)**

| **Definition** | **Example Timestamp Value** | **Precision** |
| --- | --- | --- |
| TIMESTAMP(0) | 2024-02-21 10:05:30 | No milliseconds |
| TIMESTAMP(3) | 2024-02-21 10:05:30.123 | Milliseconds (3 decimal places) |
| TIMESTAMP(6) | 2024-02-21 10:05:30.123456 | Microseconds (6 decimal places) |
| TIMESTAMP(9) | 2024-02-21 10:05:30.123456789 | Nanoseconds (9 decimal places) |

**Use Cases**

* **Lower precision (p=0,1,2)** → For use cases where sub-second accuracy is **not needed** (e.g., logging events).
* **Higher precision (p=3,6,9)** → Needed for **real-time processing**, financial transactions, and scientific applications.