**Flink Retract and Update**

In **Apache Flink**, a **dynamic table** represents a continuously evolving dataset, often derived from a streaming source. These tables are used in **Flink SQL** and **Table API** to process real-time data efficiently.

The **"Retract" mode** in Flink is a **changelog mode** used to handle **updates and deletions** in **dynamic tables**. It helps in **processing modifications to data** in streaming queries where **previous results need to be corrected**.

**How Does Retract Mode Work?**

Flink handles **dynamic table updates** using **changelog messages**, which can be of three types:

1. **INSERT** – A new row is added to the table.
2. **RETRACT (Delete)** – A previously inserted row is removed.
3. **UPDATE (Replace)** – A row is retracted first and then reinserted with updated values.

The retract mechanism works as follows:

* When a row **changes**, Flink first **retracts** (removes) the old row.
* Then, it **inserts** the new row with updated values.
* If a row is completely removed, only a **retract message** is sent.

**Example of Retract Mode in Flink SQL**

**Scenario: Aggregation in a Streaming Query**

Imagine a Flink SQL query that **counts the number of orders per product in real-time**:

SELECT product\_id, COUNT(\*) AS order\_count

FROM orders

GROUP BY product\_id;

**Retract Mode Behavior**

* When a new **order** arrives, Flink **inserts** the updated count.
* When an **order is removed** or modified, Flink **retracts** the old count and **inserts** the new count.

**Example Input Stream (orders Table)**

| **product\_id** | **order\_id** |
| --- | --- |
| A | 1 |
| A | 2 |
| B | 3 |
| A | 4 |

**Resulting Changelog Output**

| **product\_id** | **order\_count** | **Type** |
| --- | --- | --- |
| A | 1 | INSERT |
| A | 2 | UPDATE (RETRACT 1, INSERT 2) |
| B | 1 | INSERT |
| A | 3 | UPDATE (RETRACT 2, INSERT 3) |

**When is Retract Mode Used?**

Retract mode is used in queries where updates are required, such as:  
✅ **Aggregations (SUM, COUNT, AVG)** over streaming data.  
✅ **Joins on streaming tables** where rows change over time.  
✅ **Handling late events** that require correcting previous computations.

**Key Takeaways**

🔹 **Retract messages ensure correctness** when updating results in dynamic tables.  
🔹 **Flink first removes outdated rows** before inserting new ones.  
🔹 **Used in aggregation, joins, and updatable queries** on streaming data.