AUTO CDC APIs

Learning Objectives

- Definition of Expectations
- SQL and Python Syntax

AUTO CDC APIs

- Simplify change data capture (CDC) processing
- Replace MERGE INTO statement
- Support SCD type 1 and type 2

SQL Syntax

```
CREATE FLOW flowname AS
AUTO CDC INTO target_table
FROM stream(cdc_source_table)
KEYS (key_field)
APPLY AS DELETE WHEN operation_type = "DELETE"
SEQUENCE BY sequence_field
COLUMNS * EXCEPT (operation_type , sequence_field)
STORED AS  SCD TYPE 1;
```

SEQUENCE BY

- Specify the logical order of CDC events in the source data
  - Used to automatically handling out-of-sequence records

- Two data structures are created
  - **taget view**
  - **__apply_changes_storage_**target backing table
    - extra information (e.g., tombstones) to handle out-of-order data

- Sequencing on multiple columns
  - `SEQUENCE BY STRUCT(operation_timestamp, operation_number)`

Python Syntax

```
create_auto_cdc_flow(
  target = "target_table",
  source = "cdc_source_table",
  keys = ["key_field"],
  sequence_by = col("operation_date"),
  apply_as_deletes = expr("operation_type = 'DELETE'"),
  except_column_list = ["operation_type", "operation_date"],
  stored_as_scd_type = 1
)
```

Old Syntax

- **SQL**

```
APPLY CHANGES INTO target_table
FROM stream(cdc_source_table)
KEYS (key_field)
APPLY AS DELETE WHEN operation_type = "DELETE"
SEQUENCE BY operation_date
COLUMNS * EXCEPT (operation_type, operation_date)
STORED AS  SCD TYPE 1;
```

Old Syntax

- **Python**
  - apply_changes(

```
    target = "target_table",
    source = "cdc_source_table",
    keys = ["key_field"],
    sequence_by = col("operation_date"),
    apply_as_deletes = expr("operation_type = 'DELETE'"),
    except_column_list = [" operation_type", "operation_date"]
)
```