PROMOTING TO SILVER LAYER

- Silver layer may contain single or multiple tables.

- Contains validated & enriched data, that can be trusted for downstream analytics.

- Parquet format enforces Schema-On-Read
- Delta Lake enforces Schema-On-Write.

- Enforcement prevents bad records.

- Schema Evolution allows new fields to be added, but cannot remove fields.
This is because, files for prev. written records are left unmodified.

For prev. records, newly added fields are dynamically read as Null.

Schema is recorded in the Transaction Logs.

- Delta Lake allows adding of constraints to fields, such as CHECK, NOT NULL, BOOLEAN condition

Ex:-

```
ALTER TABLE <table_name>
ADD CONSTRAINT <constraint_name>
CHECK age > 0;
```

Constraint, if not matched, then throws exception, and causes Job failure.
Thus, we should apply alternative Quality check Approach :-

- Add a "validation" field that captures validation errors. Null means validation passed.
- Non-compliant data can be quarantined to alternate location.
- Instead of job failing warning should be given and failed validation msg should be recorded.

.withWatermark()

- Used to handle late arriving data. Data arriving before than interval set in withWatermark() gets discarded, and not considered.

Used as an attribute of Spark.readStream

Use case :-

≥ When dropping duplicates from Structured Streaming query, this can be used to enhance efficiency.

```
dedupedDF = spark.readStream
    .table("bronze")
    .select("time", "device_id", "timestamp")
    .withWatermark("time", "30 seconds")
    .dropDuplicates(["device_id", "time"])
```

used here
1st param is the column containing time or timestamp, based on which, delay of records arrival can be tracked.

if two or more records have this combination as matching one, only one will be kept, & rest will be dropped. It takes a list of string of column names.

For example, if the value in time column is, say 10:15:25 (hr ← 10, min ↑ 15, sec → 25), and if it arrives till 10:15:55 it will be considered but not later, as, there will be difference of more than 30 seconds here

In Spark, there are 2 timings, related to records of Structured Streaming :-

Event Time

This time is the one when the event occurs, that generates the records.
Ex:- If a Kafka event tracks user click, then the time at which user performs the click, will be the event time.

This time will be recorded by Kafka, and will physically exist as timestamp, in the generated record.
This record is sent to spark, for processing.

Processed Time

Processed time is the current time of the system/machine, on which the record is processed.
It is the time, at which spark is processing that record.

Processed time does not exist in the record physically, and is inferred by Spark, from the Operating System, dynamically.

⇒ Now, we understood the diff. of both the above times.
Using the .withWatermark() we actually define the max time interval b/w Event Time & Processed time.
If for any record arrived, the difference is more

than that specified time, that record is discarded..

Syntax:-

[.withWatermark("time", "5 minutes")]

"time" → column name of that column in the dataframe, that contains the event time

"5 minutes" → Max permissible time interval b/w event time & processed time, for which record will be considered as valid.

≥ Deduplication can also be done using an Insert-only Merge.
