Bronze Ingestion Patterns :-

Source schema is retained

- This step includes how raw data should be mapped to initial bronze tables.

This mapping should not include operations, such as joins, aggregations, etc.

This mapping can be :-

- 1:1 :- 1 Bronze table for each dataset.
  Then, the bronze table is called SinglePlex.
- n:1 :- 1 Bronze table for multiple datasets.
  Here, the bronze table is called Multiplex Bronze Table

1. SinglePlex Ingestion :-

- Each dataset ingestion runs in isolation, given the resources appropriately scale to the data.

⇒ Good for batch processing.
Complex for Stream processing.

Hard limits for no. of jobs that can be run on a workspace :-

i) 1000 concurrent running jobs
ii) 5000 runs per hour.

↓

disadvantages of running multiple streams on a single cluster

2) Multiplex Ingestion :-

⇒ Assuming initial ingestion stage processing is minimal, cloud object storage, ∠ files can also be used with autoloader.

⇒ Databricks use Structured Streaming to reduce ingestion complexity to Delta Lake.

⇒ Combining many datasets into a single stream allows smaller data to be updated with same freq. as of high velocity data.

Kafka should not be used as Bronze data source

- as data retention is limited.
- Lost/deleted data can't be recovered.
- Separate process will be needed to write to each silver table.
- Stream data may get piled up and lost.

Bronze layer eliminates all the above problems.

- For large streaming queries, Bronze layer is must.
- Bronze table should only be used to store raw data & metadata, so that, throughput to downstream tables is optimum.
  It is a low-cost buffer.

Using AUTO LOADER with Structured Streaming :-

⇒ Using autoloader, to read incoming incremental load of json file in cloud object storage :-

```
query = spark.readStream
        .format("cloudFiles")
        .option("cloudFiles.format", "json")
        .option("cloudFiles.schemaLocation", "<path>")
        .load("<data_path>")
```

setting schema

path where the json files exist

⇒ For using auto-loader in Structured streaming, the format should be :- `.format("cloudFiles")` (camel case)

In this case the file format is passed as an option :- `.option("cloudFiles.format", "json")`

file format here, can be json, csv, avro, etc

```
query = spark.
```

1) readStream details
2) . join / select
3) . writeStream details