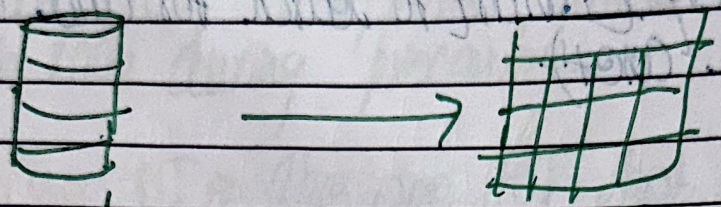
Date: / /

Databricks Professional DataEngineer Certificate

Bronze layer Ingestion patterns

- Singleplex → One to One mapping.
- Multiplex → Many to One mapping.

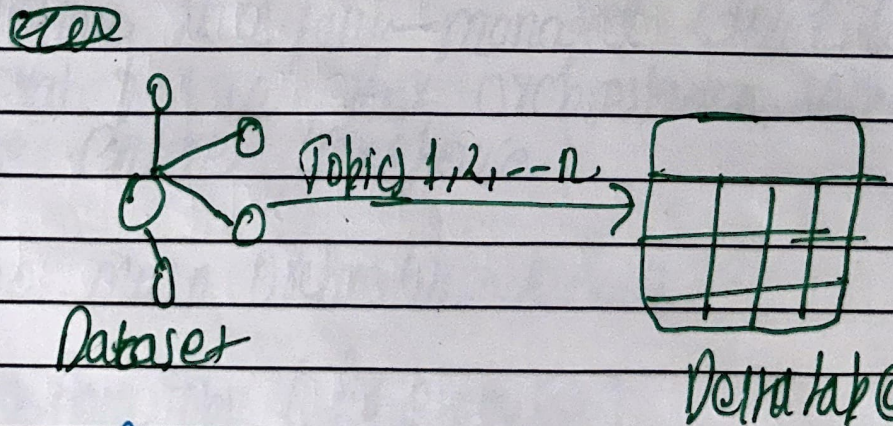
Single plex mapping.



Each pe dataset/topic is ingested to a bronze tab. called as Singleplex mapping

Note → If we use it practically there can be limit of maximum Concurrent jobs in workspace.

Multiplex Ingestion Model.



And after bronze layer there are segregated in the silver layer.

Date: / /

Slowly Changing dimension (SCD)

Slowly Changing Dimensions (SCD) is a data management concept that determines how tables handles data which change over time.

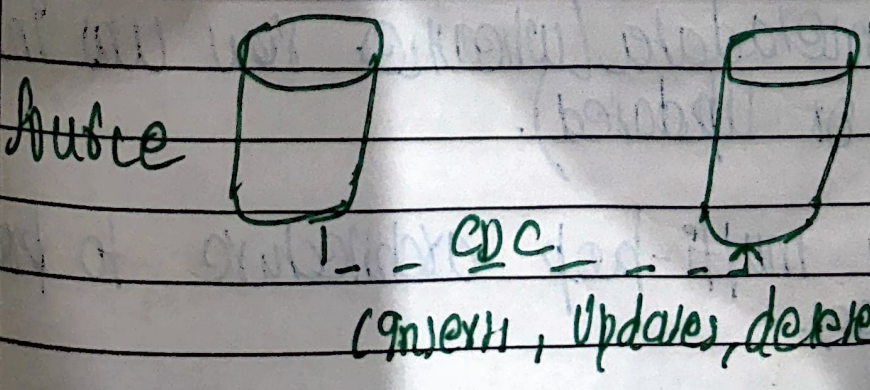
Whether you want to overwrite the values or retain their history.

SCD types:

- Type 0 → No changes allowed, these kind of tables are either static tables or Append Only tables.
- Type 1 → Overwrite. No history is retained. (use delta time travel if required).
- Type 2 → Add new row of each change and mark the old as obsolete, retain the full history of values.

Change data Capture (CDC)

It is a process of identifying changes made to data in the source and delivering those changes to the targ



There can be row level changes →

① New records insertion

② Updating existing records

③ Deleting existing records.

Merge operation limitation -

① Merge can not be performed if multiple source rows matched and attempted to modify the same target row in the Delta table

② CDC feed with multiple updates for the same key will generate an exception

Delta lake CDF (Change Data Feed)

↳ It automatically generated CDC feeds about delta lake tables.

↳ Records row-level changes for all data written into delta table.

Row data + metadata (whether row was inserted deleted or updated).

↳ It is used in multi-hop architecture to propagate the changes.

Example of CDF Enabled table

Delta table (V1) →

ID	Country	Vaccination Rate
FR	France	0.7
CA	Canada	0.65

After update ↓

Delta table (V2) →

CountryID	Country	Rate
FR	France	0.65
CA	Cada	0.75

Table changes ↓

ID	Country	Rate	Changetype	Time	Version
FR	France	0.7	Update_prei	07:00	2
FR	"	0.75	Update_post	07:00	2
CA	Canada	0.6	Update_pre	09:00	2
CA	"	0.65	Update_post	09:00	2

Preimage → Before update was done.
Post image → After Update was done.

And if there is insert or delte it will record the delet and insert in same format

Enabling CDF

1 New table → Create table mytable (id int, name str
TableProperties (delta.enbleChangeDate

Date: / /

Existing table

Alter table mytable
Set TblProperties (delta.enableChangeData.feed = tr

↳ And to enable CDF to all new table

↳ Spark.databricks.delta.properties.defaults.enable
ChangeData.feed

Use CDF when

↳ Table changes include updates and/or deletes
↳ Small fraction of records updated in each batch (from CDC feed)

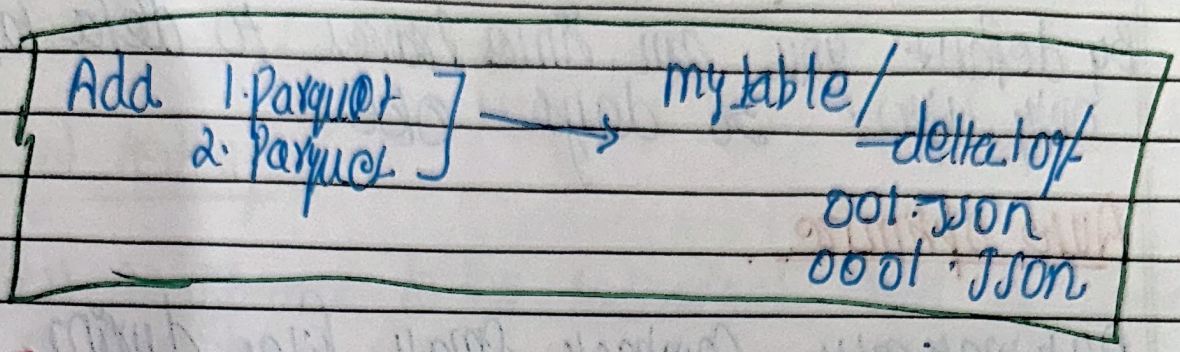
Stream-static join:

↳ Streaming tables are ever-appending data sources
↳ Static tables contain data that may be changed or overwritten - breaks the requirements of an ever-appending source for structured streaming

Partitioning: A partition is a subset of rows that share the same value for predefined partitioning column
It is like subfolders being created; it optimizes the query.

Transaction log

Each commit to the table is written out as a JSON file



Transaction log Checkpoints.

- Databricks automatically creates parquet checkpoint files every 10 commits to accelerate the resolution of current table state.
- Then, Spark has to perform incremental processing of newly added JSON files.
- Delta lake captures statistics in the transaction log for each added data file.

Delta lake file statistics.

- These statistics indicate per file:-

(1) Total number of records.

- Statistics on the first 32 columns of the table

(2) min value in each col
(3) max " " " "
(4) Null value count for each column.

Note:- Running VACUUM doesn't delete the delta log files. And these log files are automatically cleaned up by databricks

By default you can time travel to delta table only upto 30 days old

Auto optimize.

Automatically compacts small files during individual writes to a table

4 2 complementary features.

(i) optimized writes: attempts to write 128 MB files

(ii) Auto compaction.:

- After the write completes, it checks if files can further be compacted.
- If yes, it runs an OPTIMIZE job toward a file size of 128 MB (instead of 1GB).

REST API.

Databricks offers many api to create, run, ---- etc jobs

Data Pipeline testing → Data quality tests: Test the quality of the data. It apply check constraint to delta tables

There are standard term: test the code logic

- Unit testing
- Integration testing
- End-to-End testing

Unit testing

- It is an approach to test individual units of code, such as functions.
- If you make any change to them in future, you can determine whether they still work as you expect them to.
- This helps you find problems with your code faster and easlier in the development life cycle.
- An assertion is a statement that enables you to test the assumptions you have made in your code.

```
assert func() == expected_value
```

Syntax of assert

Integration testing

- Approach to test the interaction b/w subsystems of an application.
- Software modules are integrated logically and tested as a group.

End to End testing

- Approach to Ensure that your application can run properly under real-world scenarios
- Simulate a user experience from start to finish

Certification Overview

- 120 mins for exam
- 60 questions
- Pass score 70% (42/60)
- Exam fee = $200

Exam questions

- Data processing (18/60)
- Data modeling (12/60)
- Databricks tooling (12/60)
- Security and governance (6/60)
- Testing and deployment (6/60)
- Monitoring and logging (6/60)

Out of scope topics

- DLT (Delta Live Tables)
- Scala
- Orchestration tools (Airflow, ADF etc)
- Pub/sub system config (kafka ...)

- Automation Servers (Jenkin, Azure Devops, etc).
- Manged CI/CD.
- Gitflow.
- Cloud-Specific Security and integration
- Infrastructure as Code (Terraform --)

Code Examples

- Code will be in mainly python
- Delta Lake functionality in SQL.

Question type

- MCQ only Questions
- Question types:

1. Conceptual Questions
2. Code-Based Question

Date: / /

Short :- Crash Notes for Databrick Professional De.

1. The minimum permission to view metrics and Spark UI of cluster is "Can Attach To" privilege on the cluster.
2. For production Databricks jobs "Job Clusters" are recommended to use.
3. Job clusters provide isolated environment for each job runs.
4. "Can Restart" permission is minimal permission to start and terminate the job.
5. Whenever External table is dropped only metadata of table is deleted but the data files is kept on same location.
6. "Can manage" is minimum permission to edit cluster.
7. If you want to print secret keys "REDACTED" is printed.
8. %sh magic command executes shell code on the local driver machine which leads to significant performance overhead.
9. File statistics in the delta transaction log is used to leverage the query optimizer.

(10) Shallow cloned table doesn't copy the data but refers the Delta transaction logs. If you run VACUUM command on source table you will get error of data not present in the shallow cloned table.

(11) Syntax for DEEP Clone → create or replace orders_archive deep clone orders.

(12) Select * from table@v36 → Table version 36.

(13) If data is continously appended from bronze to silver the we can use structured Streaming in batch mode using trigger availableNow option.

(14) Optimized writes and auto compaction automatically generate smaller data file to reduce the duration of future MERGE operation.

(15) DBX maintains history of your jobrun upto 60 days. If you need to preserve job runs.