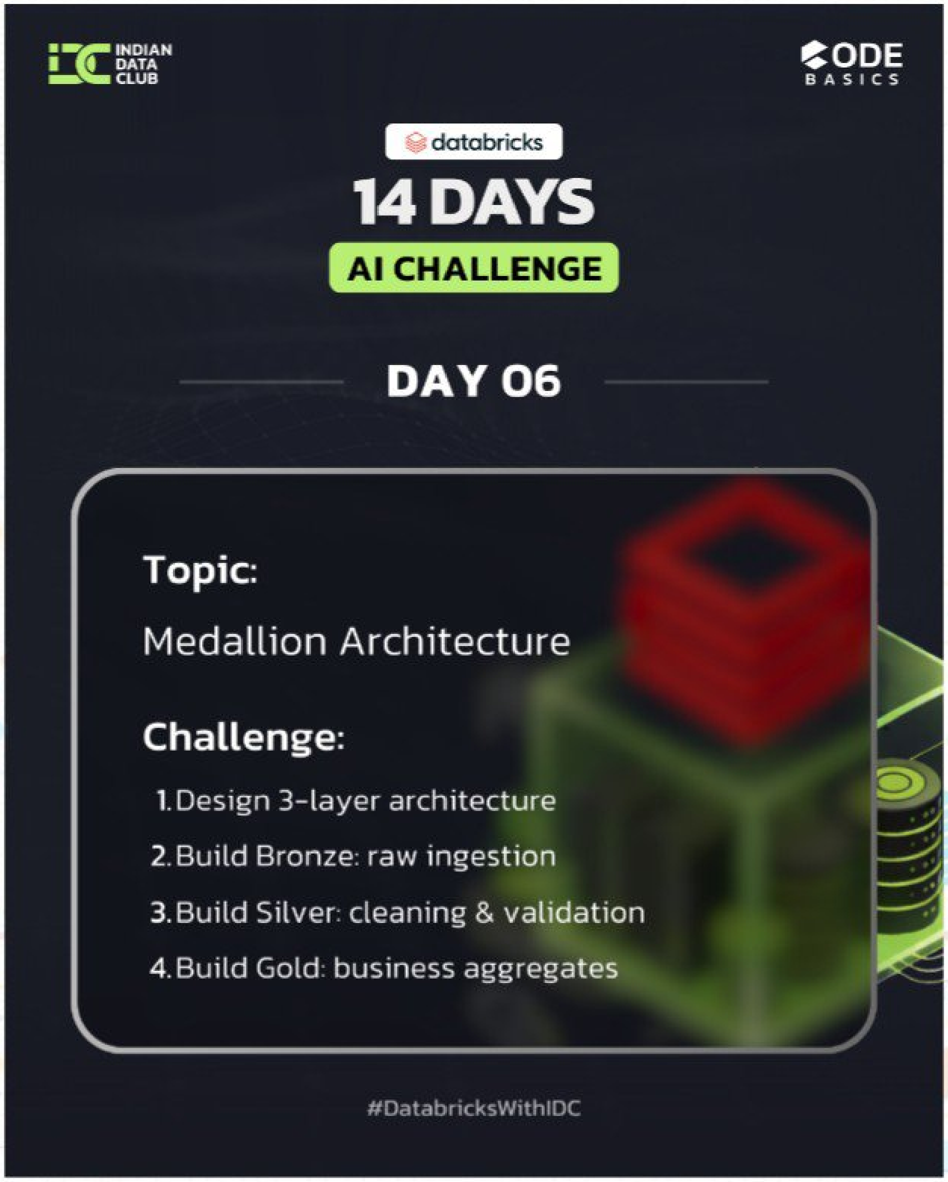
Day 6 - Medallion Architecture



INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 06

Topic:

Medallion Architecture

Challenge:

1. Design 3-layer architecture
2. Build Bronze: raw ingestion
3. Build Silver: cleaning & validation
4. Build Gold: business aggregates

#DatabricksWithIDC

What is Medallion Architecture?

Medallion Architecture is a data design pattern used in Databricks and modern data platforms to organize data into layers for better quality, performance, and scalability.

It has 3 layers:
1.Bronze – Raw data
2.Silver – Cleaned & validated data
3.Gold – Business-ready data

Bronze Layer (Raw Data)

✅ *What is Bronze?*
- *Stores raw data exactly as it comes from the source*
- *No transformations (or very minimal)*
- *Acts as a data backup*

📌 *Characteristics*
- *Data may have duplicates*
- *Data may have nulls or errors*
- *Schema can evolve*
- *Append-only pattern*

🛠️ *Example*
- *Raw CSV, JSON, API data*
- *Log files, transaction dumps*

Silver Layer (Cleaned Data)

✅ *What is Silver?*
- *Cleaned and standardized data*
- *Business rules applied*
- *Data is reliable and usable*

📌 *What we do here*
- *Remove duplicates*
- *Handle null values*
- *Fix data types*
- *Apply validations*
- *Filter bad records*

Gold Layer (Business Aggregates)

✅ *What is Gold?*

- *Final layer for analytics & reporting*
- *Aggregated and optimized for queries*
- *Used by dashboards, BI tools, stakeholders*

📌 *Examples*

- *Daily sales summary*
- *Monthly revenue by region*
- *Customer lifetime value*

Best Practices for Each Layer

Bronze :

✔ *Keep data immutable*
✔ *Add ingestion timestamp*
✔ *Don’t delete records*

Silver :

✔ *Apply data quality checks*
✔ *Use meaningful column names*
✔ *Track rejected records*

Gold :

✔ *Optimize for performance*
✔ *Use aggregations wisely*
✔ *Create tables per business use case*

Incremental Processing

❓ *What is Incremental Processing?*
Instead of reprocessing all data, process only new or changed data.

📌 *Why?*
·Faster
·Cost-efficient
·Scalable for big data

🛠️ *Example*
·Use ingestion_date
·Use MERGE INTO
·Use watermarking