Cloud-Scale Analytics Platform on Azure

Audience

Intermediate to Advanced **Data Engineering learners**

Role Simulation

Learners act as **Enterprise Data Engineers** working for a mid-to-large organization undergoing a cloud data modernization initiative.

1. Business Context

A multinational retail enterprise operates across multiple regions and channels (online & physical stores).
The company wants to modernize its legacy data platform and build a **cloud-native analytics solution** that supports:

- Near-real-time analytics
- Historical trend analysis
- Business dashboards for leadership
- High data quality and reliability

The platform must be built on **Azure-based big data technologies** and follow **enterprise-grade data engineering best practices**.

2. High-Level Objective

Design and implement a **scalable, incremental, and reliable data engineering pipeline** that:

1. Ingests raw data into a Data Lake
2. Processes and transforms data using Spark on Azure Databricks

3. Stores curated data in Delta Lake format
4. Supports **incremental data uploads**
5. Applies **data calibration and validation rules**
6. Serves clean, analytics-ready data to Power BI

3. Mandatory Technology Stack

Learners **must use** the following tools and concepts:

Core Platform

- Azure Databricks
- Apache Spark (PySpark or Spark SQL)

Storage

- Azure Data Lake (Bronze / Silver / Gold architecture)
- Delta Lake (ACID tables, versioning)

Analytics & Visualization

- Power BI (connected to curated Delta tables)

Data Engineering Concepts

- Incremental data ingestion
- Data calibration & data quality enforcement
- Schema evolution handling
- Enterprise-grade logging & error handling

4. Data Domains & Source Systems

4.1 Source Data (Simulated)

Learners should assume the following **source systems**:

1. Sales Transactions (Primary Fact Table)

- Format: CSV or JSON
- Arrival: Daily (incremental)
- Volume: High (millions of rows over time)

Example fields:

- transaction_id
- transaction_timestamp
- store_id
- product_id
- quantity
- unit_price
- discount
- total_amount
- currency

2. Product Master Data

- Format: CSV
- Arrival: Periodic updates (slowly changing)

Fields:

- product_id
- product_name
- category
- brand
- standard_price

3. Store / Region Reference Data

- Format: CSV
- Arrival: Rare changes

Fields:

- store_id
- store_name
- region
- country

5. Target Architecture (Required)

Learners must design and implement the following **multi-layer architecture**:

5.1 Bronze Layer (Raw Data Lake)

Purpose

- Store raw, unmodified data exactly as received

Requirements

- Data stored in Data Lake in original format
- No business transformations
- Partitioned by ingestion date
- Ingestion metadata added:
  - ingestion_timestamp
  - source_system

5.2 Silver Layer (Cleaned & Standardized Data)

Purpose

- Data cleansing, standardization, and calibration

Mandatory Processing

- Convert all timestamps to UTC
- Normalize currencies (if applicable)
- Handle nulls and invalid values
- Remove duplicates
- Enforce schema using Delta Lake

Incremental Processing

- Only new or changed records must be processed
- Use watermarking or transaction timestamp logic

5.3 Gold Layer (Analytics-Ready Data)

Purpose

- Optimized tables for reporting and analytics

Examples

- Daily sales summary
- Monthly revenue by region
- Product performance metrics

Requirements

- Aggregated, business-friendly schemas
- Optimized Delta tables
- Ready for Power BI consumption

6. Incremental Data Upload Requirements

Learners **must implement incremental ingestion**, not full reloads.

Accepted Strategies

- Transaction timestamp comparison
- Delta Lake MERGE (UPSERT)
- Watermark-based ingestion

Mandatory Features

- Track last processed timestamp
- Ensure idempotent runs (no duplicates)
- Support late-arriving data

7. Data Calibration & Data Quality Rules

Data calibration is **mandatory** and must be explicitly documented and implemented.

7.1 Calibration Examples

- Ensure total_amount = quantity × unit_price – discount
- Recalculate totals if mismatched
- Standardize currency values
- Enforce numeric precision

7.2 Data Quality Rules

Learners must implement and log:

- Null checks on key fields
- Range validation (e.g., quantity > 0)
- Duplicate transaction detection

- Invalid foreign key detection (store_id / product_id)

Output

- Invalid records must be:
    - Logged
    - Stored separately (quarantine table)

8. Delta Lake Requirements

Learners must demonstrate:

- Delta table creation
- ACID compliance
- MERGE operations
- Time Travel queries
- Schema evolution handling

9. Power BI Integration Requirements

Dashboards (Minimum)

Learners must create **at least 2 dashboards**:

1. **Executive Sales Overview**
    a. Total revenue
    b. Revenue trend
    c. Top regions
2. **Product Performance**
    a. Top products
    b. Category-wise sales
    c. Monthly trends

Data Source

- Power BI must connect directly to Gold Delta tables

10. Logging, Monitoring & Error Handling

Mandatory Logging

- Ingestion start & end times
- Record counts per layer
- Error messages & failed records

Failure Handling

- Pipeline must not corrupt existing data
- Failed batches must be re-runnable

11. Security & Enterprise Practices (Conceptual)

Learners must **document** (even if not fully implemented):

- Data access control strategy
- Separation of environments (Dev / Test / Prod)
- Data lineage explanation

12. Deliverables (Strict)

Each learner / team must submit:

1. Architecture Document

- Diagram (logical)

- Explanation of each layer
- Technology choices

2. Data Model Documentation

- Bronze / Silver / Gold schemas
- Business logic explanations

3. Databricks Code

- Notebooks or scripts
- Well-commented
- Parameterized where possible

4. Power BI Report

- At least 2 dashboards
- Clear visuals and KPIs

5. README / Final Report

- End-to-end pipeline explanation
- Incremental logic
- Calibration logic
- Challenges & assumptions

13. Evaluation Criteria

Learners will be evaluated on:

- Correct use of Azure Databricks & Spark
- Proper Data Lake & Delta Lake design
- Incremental ingestion correctness
- Data calibration accuracy
- Code quality & documentation
- Enterprise-grade thinking

14. Stretch Goals (Optional but Encouraged)

- Slowly Changing Dimensions (SCD Type 2)
- Data quality metrics dashboard
- Performance optimization (partitioning, caching)