Project : Retail 360 Analytics :
Mentor : Anandh Kumar

Objective:
This capstone project will involve applying data engineering techniques to a retail sales dataset using PySpark in Azure Databricks. Students will perform data profiling, validation, consistency checks, integrity checks, and quality assessments on various dimension and fact tables. Following this, they will apply schema enforcement, change data capture (CDC) with upserts, and streaming of data. The processed data will be stored in Azure Data Lake Storage (ADLS) using Delta Lake format, optimized for storage, partitioning, parallelism, query optimization, and caching.

Project Breakdown:

1. Initial Data Loading and Pre-Processing (Raw/Bronze Layer)
Objective: Load raw data into Azure Data Lake Storage (ADLS) and perform initial data screening using Python and Pandas.

Mount Azure Data Lake Storage (ADLS) in Azure Databricks using a SAS token stored securely in Azure Key Vault.

Step 1.1: Load Raw Data:

Import retail sales CSV files into Pandas DataFrames.
Review the dataset structure, including columns, data types, and initial observations of data quality.

Step 1.2: Initial Data Screening:

Clean the data by removing duplicates, handling missing values, and performing basic data type conversions.
Save the cleaned data back to ADLS in the "raw" or "bronze" folder.

Deliverables:

Cleaned CSV files stored in the ADLS "raw" or "bronze" folder.

2. Data Profiling (Silver/Cleansed Layer)

Objective: Perform data profiling to understand the data's structure, distribution, and quality for both dimension and fact tables.

Step 2.1: Load Data into PySpark:

Load the pre-processed data from the "raw" or "bronze" layer into PySpark DataFrames.

Step 2.2: Profiling for Dimension Tables (nation, region, orders, customer, part, partsupp, supplier):

Column Statistics: Calculate summary statistics such as count, mean, standard deviation, min, max, and percentiles for numerical columns.
Categorical Analysis: Evaluate unique values, frequency distribution, and cardinality for categorical columns.
Null Value Analysis: Identify the number of null values per column and calculate the percentage of missing data.

Step 2.3: Profiling for Fact Table (line items):

Perform similar profiling steps as above, focusing on transactional data metrics such as order counts, average prices, and quantities.

Deliverables:

A comprehensive profiling report detailing the structure, distribution, and quality of each table.

3. Data Validation and Consistency Checks
Objective: Validate the data against predefined rules and check for consistency across dimension and fact tables.

Step 3.1: Data Validation:

Schema Validation: Ensure that all columns adhere to the predefined schema types.
Range Checks: Validate that numerical columns fall within expected ranges (e.g., quantity and price should be positive).

Step 3.2: Consistency Checks Across Dimension and Fact Tables:

Referential Integrity: Ensure that foreign key relationships between fact and dimension tables are maintained. For example, every customer_id in the fact table should exist in the customer dimension table.

Data Synchronization: Verify that dimension data (like product prices and descriptions) matches with what's referenced in the fact table.

Step 3.3: Cross-Table Validation:

Perform checks to ensure consistency between related datasets, such as ensuring all regions in the orders table are present in the region dimension table.

Deliverables:

Validation reports that outline any discrepancies or issues identified during the validation and consistency checks.

4. Data Quality Assessment
Objective: Assess and improve data quality across all tables to ensure high standards for downstream analytics.

Step 4.1: Data Quality Checks:

Accuracy: Ensure the data accurately reflects the real-world entities it is supposed to represent (e.g., correct customer names and addresses).
Completeness: Confirm that all required fields are populated, especially critical columns (e.g., order_id, product_id).
Uniqueness: Check for duplicate records, particularly in unique key fields (e.g., customer_id in the customer dimension table).
Timeliness: Verify that date fields are accurate and consistent, such as ensuring order_date precedes delivery_date.

Step 4.2: Quality Improvement:

Use PySpark to perform data cleaning actions based on the quality checks, such as correcting errors, filling missing values, or removing duplicates.

Deliverables:

Data quality reports and updated DataFrames reflecting the cleaned and validated data.

Day 3:

After initial data validation in PySpark, participants will perform various transformations on the data in the Bronze layer. These tasks will help in refining the data for further processing in the Silver layer and ensure participants gain hands-on experience with different PySpark functions and transformations.

Joins:

- **Perform inner, left, right, and full joins between datasets to enrich data.**

Adding New Columns:

- **Add new columns based on static values or calculated logic.**

Deriving New Columns:

- **Create new columns using calculations or transformations from existing columns.**

Applying String, Date, and Math Functions:

- **Manipulate data using string functions (e.g., length, substring).**

- **Extract date components (e.g., year, month).**
- **Apply math functions (e.g., sqrt, round).**

Applying Aggregate Functions:

- **Perform aggregations like sum, average, count, min, and max.**

Column Renaming:

- **Rename columns for better readability or alignment with business terminology.**

Filtering Data:

- **Filter records based on conditions or remove null values.**

Sorting Data:

- **Sort datasets based on specific columns.**

Other Transformations:

- **Perform other necessary transformations based on business requirements or data enhancements.**

Delta table Activities

5. Schema Enforcement (Silver/Cleansed Layer)
Objective: Enforce schema consistency to ensure all datasets conform to the required structure.

Step 5.1: Define and Apply Schema:

Define strict schemas using PySpark StructType and StructField for each table.
Apply these schemas to the DataFrames loaded from the "raw" or "bronze" layer.

Step 5.2: Handle Schema Mismatches:

Implement error handling for records that do not conform to the schema (e.g., log errors or separate them into an exception table).

Deliverables:

PySpark DataFrames with enforced schemas and logs of any schema mismatches.

6. Change Data Capture (CDC) with Upserts (Silver/Cleansed Layer)
Objective: Implement Change Data Capture (CDC) to handle data updates and deletions using the upsert method.

Step 6.1: Identify Data Changes:

Compare incoming data batches with existing data to detect new, updated, or deleted records.

Step 6.2: Perform Upserts Using Delta Lake:

Use Delta Lake's merge functionality to perform upserts (insert new records and update existing ones) based on a unique identifier (e.g., order_id).

Step 6.3: Store Updated Data:

Save the upserted DataFrames as Delta Tables in the "silver" or "cleansed" layer.
Ensure Delta Tables are partitioned based on specific columns (e.g., order_date or region).

Deliverables:

Delta Tables with upserted data in the "silver" or "cleansed" folder.

7. Streaming Data Processing (Silver/Cleansed Layer)

Objective: Set up streaming data processing for continuous data ingestion and transformation.

Step 7.1: Simulate a Streaming Data Source:(Optional)

Use the existing datasets to simulate a real-time streaming source.
Use PySpark Structured Streaming to read from this simulated source.

Step 7.2: Stream Processing:

Apply transformations, filtering, or enrichment on the streaming data to match the schema and quality of batch-processed data.

Step 7.3: Output Streamed Data:

Write the streaming output to a Delta Table in the "silver" or "cleansed" layer.
Optimize the Delta Table for query performance using partitioning and Z-Ordering.

Deliverables:

Continuously updated Delta Tables in the "silver" or "cleansed" folder.

8. Optimization Techniques

Objective: Optimize storage and query performance using Delta Lake features in Azure Databricks.

Step 8.1: Storage Optimization:

Partition Delta Tables on relevant columns (e.g., order_date, region).
Use Delta Lake's OPTIMIZE command to compact small files and improve query performance.

Step 8.2: Increase Parallelism:

Adjust the number of partitions using PySpark functions (repartition() or coalesce()) to enhance parallel processing.

Step 8.3: Query Optimization and Caching:

Cache frequently accessed DataFrames to reduce query response time.
Use Delta Lake's Z-Ordering feature to optimize queries on commonly filtered columns.

Deliverables:

Optimized Delta Tables with appropriate partitioning and caching strategies.

9. Business Curation and Sales Analysis in Azure Synapse (Gold Layer)

Objective: Perform business curation and sales analysis using Transact-SQL (T-SQL) in Azure Synapse, leveraging the cleansed data stored in Delta Tables from the Silver layer.

Step 9.1: Set Up Azure Synapse Analytics:

Create Synapse Workspace: If not already done, set up an Azure Synapse workspace linked to your Azure Data Lake Storage (ADLS) where the Delta Tables are stored.
Create Linked Services: Configure linked services in Synapse to connect to the ADLS containing your Silver layer data.

Step 9.2: Access Delta Tables from Azure Synapse:

Create External Tables: In Synapse, create external tables using T-SQL that reference the Delta Tables in the Silver layer. Ensure that Synapse can read the Delta format correctly.

Step 9.3: Perform Business Curation and Sales Analysis:

Curate Business Data: Use T-SQL to filter, aggregate, and transform the data for sales analysis. This may include:

Sales Aggregation: Calculate total sales, average order value, and other key metrics.
Customer Segmentation: Segment customers based on purchase behavior, region, or demographics.
Time-Based Analysis: Perform year-over-year, month-over-month, or seasonal trend analysis.

Store Curated Data in Gold Layer:

Create Gold Layer Tables: Create tables in the Synapse database to store curated data that is business-ready for analytics.

Load Data into Gold Tables: Use T-SQL INSERT INTO statements or CREATE TABLE AS SELECT (CTAS) to load the curated data into the Gold layer tables.

Deliverables:

Synapse workspace with external tables linked to Delta Tables.
Gold layer tables containing curated sales analysis data.

10. Integration with Power BI for Reporting
Objective: Use the curated data in the Gold layer for creating insightful Power BI reports.

Step 10.1: Connect Power BI to Azure Synapse:

Open Power BI Desktop and create a connection to Azure Synapse Analytics using the Synapse database credentials.

Step 10.2: Load Gold Layer Data into Power BI:

Import the curated sales analysis data from the Gold layer tables into Power BI.

Step 10.3: Create Reports and Dashboards:

Visualize Key Metrics: Create visualizations like bar charts, line graphs, and tables to represent sales trends, customer segments, and regional performance.
Interactive Dashboards: Build interactive dashboards that allow users to filter data by time period, region, product category, or customer segments.

Step 10.4: Publish Reports to Power BI Service:

Publish your Power BI reports to the Power BI Service to make them accessible to stakeholders.
Set up data refresh schedules to ensure that reports reflect the most up-to-date data from the Gold layer.

Deliverables:

Power BI reports and dashboards based on curated data from the Gold layer.
Published reports in Power BI Service with scheduled data refreshes.

Expected Outcomes:
Upon completing this capstone project, students will be able to:

- ☐ Perform data profiling, validation, consistency checks, integrity checks, and quality assessments using PySpark.
- ☐ Enforce schema consistency and manage data quality for various dimension and fact tables.
- ☐ Implement Change Data Capture (CDC) with upserts using Delta Lake.
- ☐ Set up and manage streaming data processing with PySpark Structured Streaming.
- ☐ Apply optimization techniques for storage, partitioning, parallelism, and query performance in Delta Lake.

Draw the architecture and workflow diagram for the project

Capstone Project Status Call Guidelines

Objective:
To provide daily updates on the progress of the capstone project, discuss challenges faced, share observations, and collaboratively find solutions.

Daily Status Call Details:

- **Time:** Every day in the evening (exact time to be agreed upon)
- **Participants:** All team members involved in the capstone project
- **Duration:** Approximately 30-45 minutes

Agenda for the Status Call:

1. **Welcome and Introduction (5 minutes)**
    - Brief welcome by the facilitator or team lead.
    - Quick roll call to ensure all participants are present.
2. **Project Progress Updates (10-15 minutes)**
    - **Individual Updates:** Each participant provides a brief summary of their progress on the capstone project since the last call.
        - What tasks have been completed?
        - What milestones have been achieved?
        - Any pending tasks?
3. **Challenges and Issues Faced (10-15 minutes)**
    - **Open Floor for Challenges:** Each participant shares any challenges or obstacles they have encountered.
        - Technical difficulties or bugs
        - Data inconsistencies or quality issues
        - Any roadblocks or dependencies affecting progress

    - **Collaborative Problem Solving:** The group discusses possible solutions or workarounds for the challenges raised.
4. **Observations and Learnings (5-10 minutes)**
    - **Share Observations:** Participants share key observations, insights, or learnings from their work.
        - Interesting findings or patterns in the data
        - Successful strategies or approaches that worked well
        - Any feedback or suggestions for improving the project workflow
5. **Next Steps and Action Items (5 minutes)**
    - **Outline Next Steps:** Discuss the next steps and action items for the upcoming day.
    - **Assign Tasks:** Assign any new tasks or responsibilities based on the discussion.
    - **Set Expectations:** Clarify what is expected from each participant before the next status call.
6. **Q&A and Wrap-Up (5 minutes)**
    - **Q&A:** Open the floor for any questions or final comments.
    - **Wrap-Up:** Summarize key takeaways from the call and confirm the time for the next meeting.

Guidelines for Participants:

- **Be Prepared:** Come to the call prepared with an update on your work and any challenges you need help with.
- **Be Concise:** Keep your updates and discussions concise to ensure everyone has a chance to speak.
- **Be Open:** Share any issues or difficulties openly so that the team can support each other.
- **Be Collaborative:** Engage in collaborative problem-solving and offer suggestions or solutions where possible.

Expected Outcomes:

- Regular updates on project progress.
- Timely identification and resolution of challenges.
- Shared learning and best practices.
- Clear action items and next steps to ensure steady progress.