Data Pipelines with Delta Live Tables

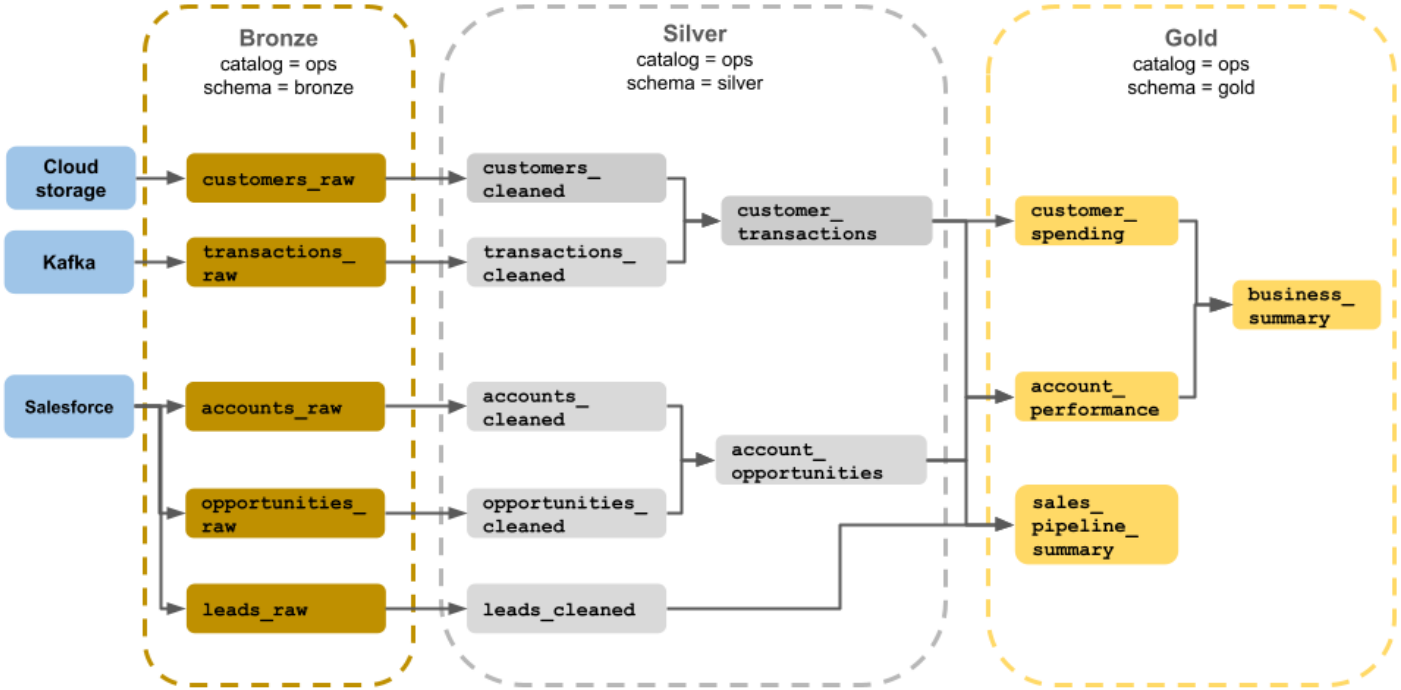
Managing and transforming data with innovative tools

MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Session Outline

- Session Overview
- Medallion Architecture
- Delta Live Tables
- Hands-on Labs

Introduction to Medallion Architecture & Delta Live Tables



Understanding Medallion Architecture

Medallion Architecture is designed to organize data into layers that enhance data quality and accessibility.

Delta Live Tables Overview

Delta Live Tables (DLT) simplify building and managing data pipelines while ensuring high data reliability.

Benefits of DLT

Using DLT leads to better performance, reliability, and automation in data processing workflows.

MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Medallion Architecture

Medallion Architecture Overview

An introduction to Medallion Architecture, discussing its significance and application in data engineering.

Bronze, Silver, Gold Layers

Explore the different layers of Medallion Architecture: Bronze, Silver, and Gold, each serving unique purposes.

Introduction to Delta Live Tables

Learn about Delta Live Tables and how they facilitate data pipeline management and processing.

Hands-on Lab

Participate in a hands-on lab session to build Medallion Architecture and gain practical experience.

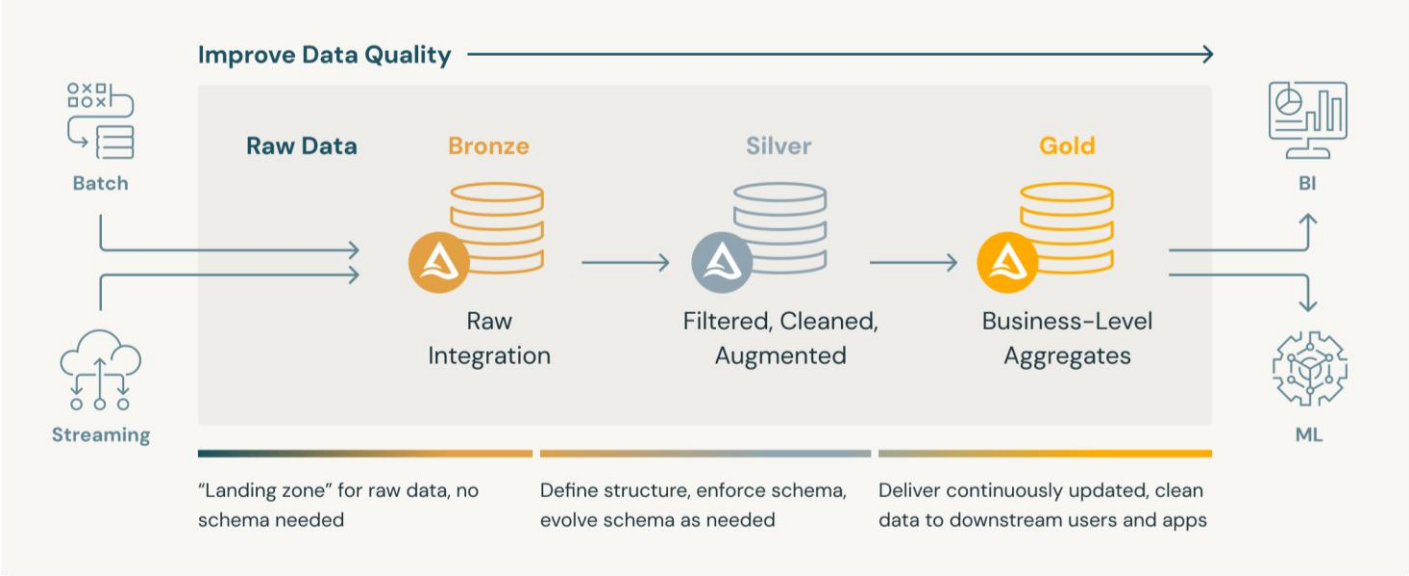
Navigating Delta Live Tables UI

Overview of the user interface for Delta Live Tables, emphasizing navigation and functionality.

MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Medallion Architecture

Overview and Benefits



Medallion Architecture

Medallion architecture describes a data refinement process structured in three layers: Bronze, Silver, and Gold.

Data Quality Improvement

Implementing this architecture enhances overall data quality, ensuring reliable and accurate information.

Governance and Performance

This structure leads to better data governance and improves system performance for users.

Bronze Layer (Raw Data)

```
import dlt
from pyspark.sql.functions import *

@dlt.table(
 comment="Raw sales data ingested to Bronze layer",
 table_properties={
  "quality": "bronze"
 }
)
def bronze_sales():
   return (
      spark.read.json("/databricks-
datasets/samples/sales/data.json")
   )
```

Definition of Raw Data

Raw data is data that has not been processed or analyzed. It represents the original state of information collected.

Minimal Transformations

The bronze layer involves minimal transformations to maintain data integrity. This ensures the original data remains intact for future analysis.

Purpose of Bronze Layer

The purpose of the bronze layer is to serve as initial storage for incoming data, allowing for easy access and retrieval.

Examples of Raw Data

Examples include raw sales figures, customer demographics, and product data, showcasing various types of incoming data.

Silver Layer (Refined Data)

Data Cleaning and Transformation

The Silver Layer involves cleaned, transformed, and enriched data, ensuring it is ready for analysis.

Data Validation and Quality Checks

Important steps in this layer include data validation, quality checks, and schema enforcement to maintain data integrity.

Joined Data Examples

Examples include sales data joined with customer and product details, providing a comprehensive view of the data.

```
@dlt.table(
  comment="Cleaned and deduplicated Silver sales data"
)
@dlt.expect("valid_quantity", "quantity > 0")
@dlt.expect_or_drop("no_nulls", "customerId IS NOT NULL")
def silver_sales():
    df = dlt.read("bronze_sales")
    return df.filter("quantity > 0").dropDuplicates(["orderId"])
```

Introduction to Gold Layer

Aggregated Data

The Gold Layer contains aggregated or highly curated data essential for effective business intelligence.

Business KPIs

This layer includes key performance indicators (KPIs) and metrics to monitor business performance.

Optimized for Reporting

The Gold Layer is optimized for business intelligence (BI) and reporting purposes, ensuring clarity and insight.

```
CREATE LIVE TABLE gold_sales_summary
COMMENT "Aggregated daily sales totals"
AS
SELECT
  orderDate,
  SUM(quantity * unitPrice) AS total_sales,
  COUNT(DISTINCT customerId) AS unique_customers
FROM
  LIVE.silver_sales
GROUP BY
  orderDate;
```

Benefits of Medallion Architecture

Advantages of Medallion Architecture

Incremental Data Quality

Medallion Architecture enhances data quality over time, ensuring more and reliable data for decision-making.

Easier Troubleshooting

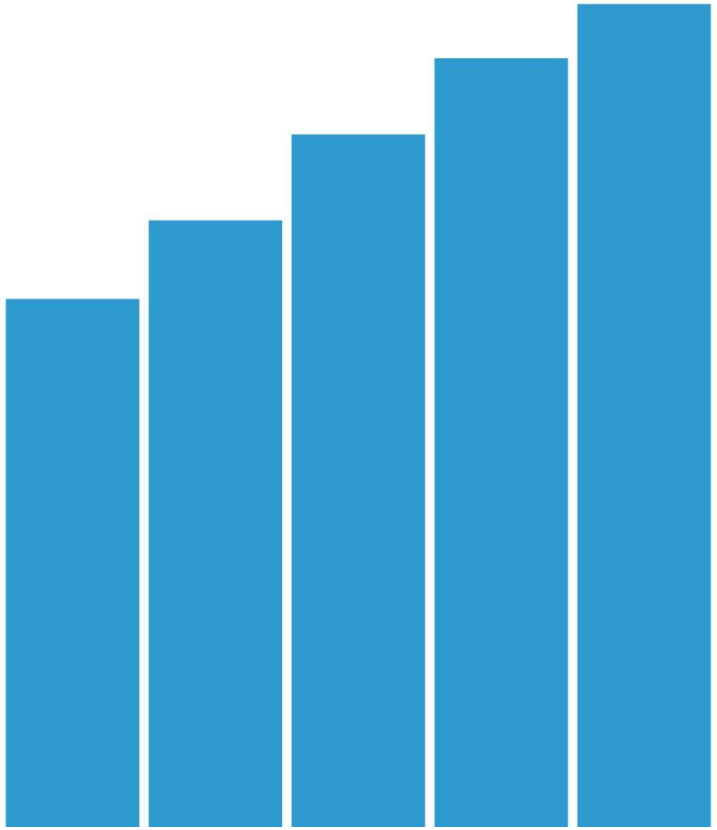
The architecture simplifies troubleshooting processes and improves da tracking, making it easier to trace issues.

Scalability

Medallion Architecture supports scalability for large datasets, allowing organizations to grow their data capabilities efficiently.

Multiple Downstream Consumers

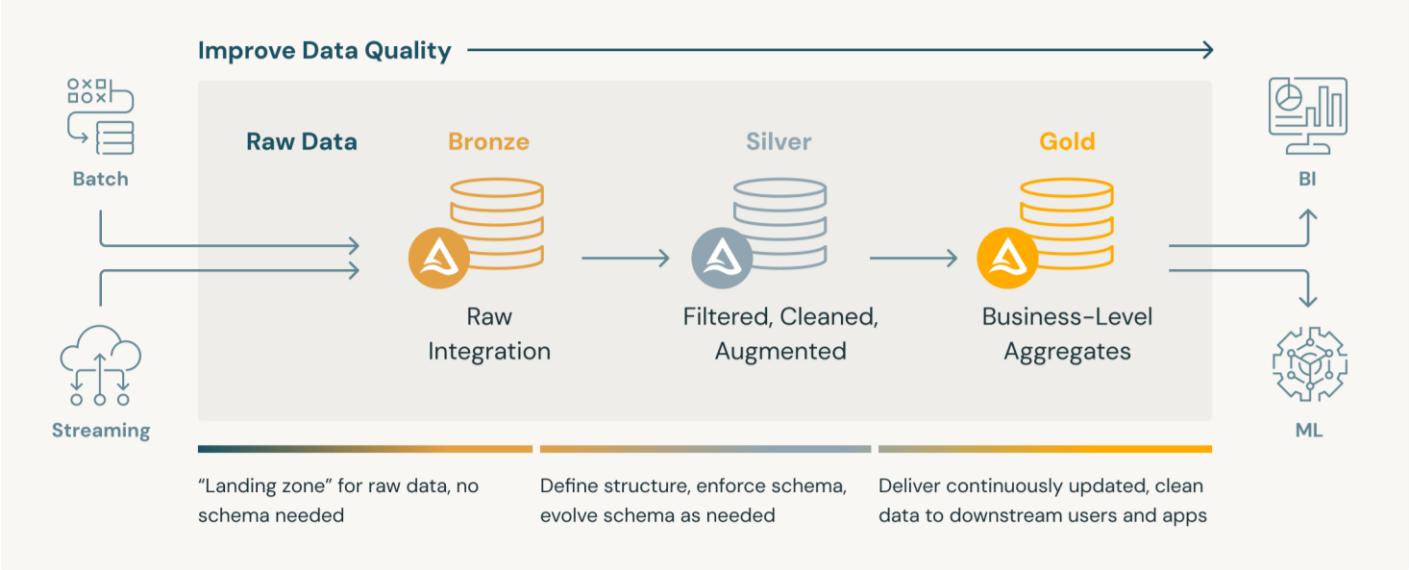
The architecture accommodates multiple downstream consumers, enhancing collaboration and data sharing across teams.



Delta Live Tables

MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Introduction to Delta Live Tables



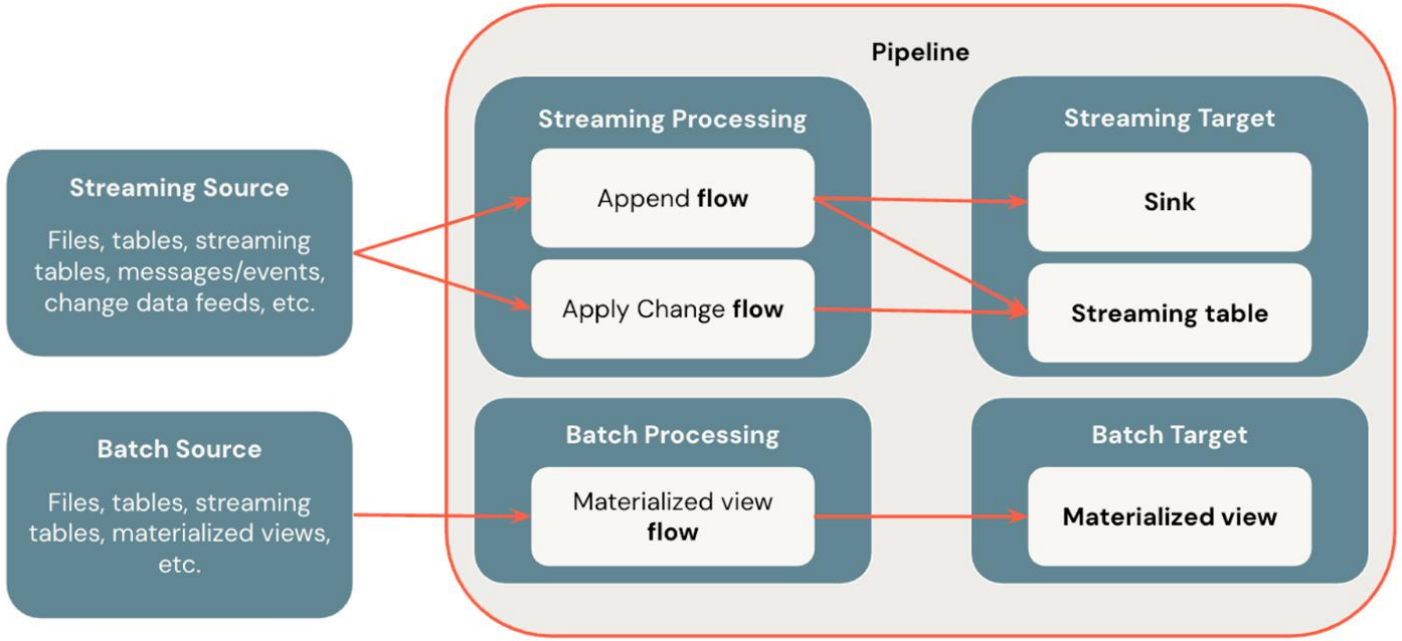
Managed ETL Framework
Delta Live Tables is a managed ETL framework that leverages Delta Lake and Apache Spark to facilitate efficient data processing.

Declarative Pipeline Creation
Users can build data pipelines in a declarative manner using SQL or Python, which streamlines development and boosts productivity.

Automated Infrastructure Management
Delta Live Tables automates infrastructure management and orchestration, enabling users to concentrate on data transformation and analysis.

Data Quality Enforcement
The framework includes features for enforcing data quality and monitoring, ensuring that high-quality data is maintained throughout the pipelines.

Key Concepts of DLT



Flows: Core processing units for streaming & batch (Append exposed). Includes *Apply Changes* (handles CDC) & *Materialized View* (batch updates).

Streaming Tables: Managed tables with streaming flows (Append, Apply Changes). Flows can be explicit or implicit.

Materialized Views: Batch targets with implicit flow definitions.

Sinks: Streaming targets (Delta, Kafka, EventHubs) supporting Append flows.

Pipelines: Containers for flows, tables, views, and sinks with automatic orchestration.

Databricks SQL: Use SQL to create/refresh streaming tables and materialized views with implicit flows.

Benefits of Delta Live Tables

Simplified ETL Pipelines

Delta Live Tables streamline ETL processes, allowing developers to implement pipelines with significantly less code.

Automatic Error Handling

Automatic error handling and recovery mechanisms ensure robust data processing and reduce manual intervention.

Declarative Data Quality Checks

Declarative checks for data quality provide built-in expectations that enhance the reliability of data pipelines.

Scalable and Maintainable

Delta Live Tables support scalability and maintainability, making it easier to adapt pipelines to growing data needs.

Real-Time Monitoring Dashboard

A real-time monitoring dashboard allows users to track data pipelines, ensuring optimal performance and quick insights.

DLT Architecture

Data Ingestion

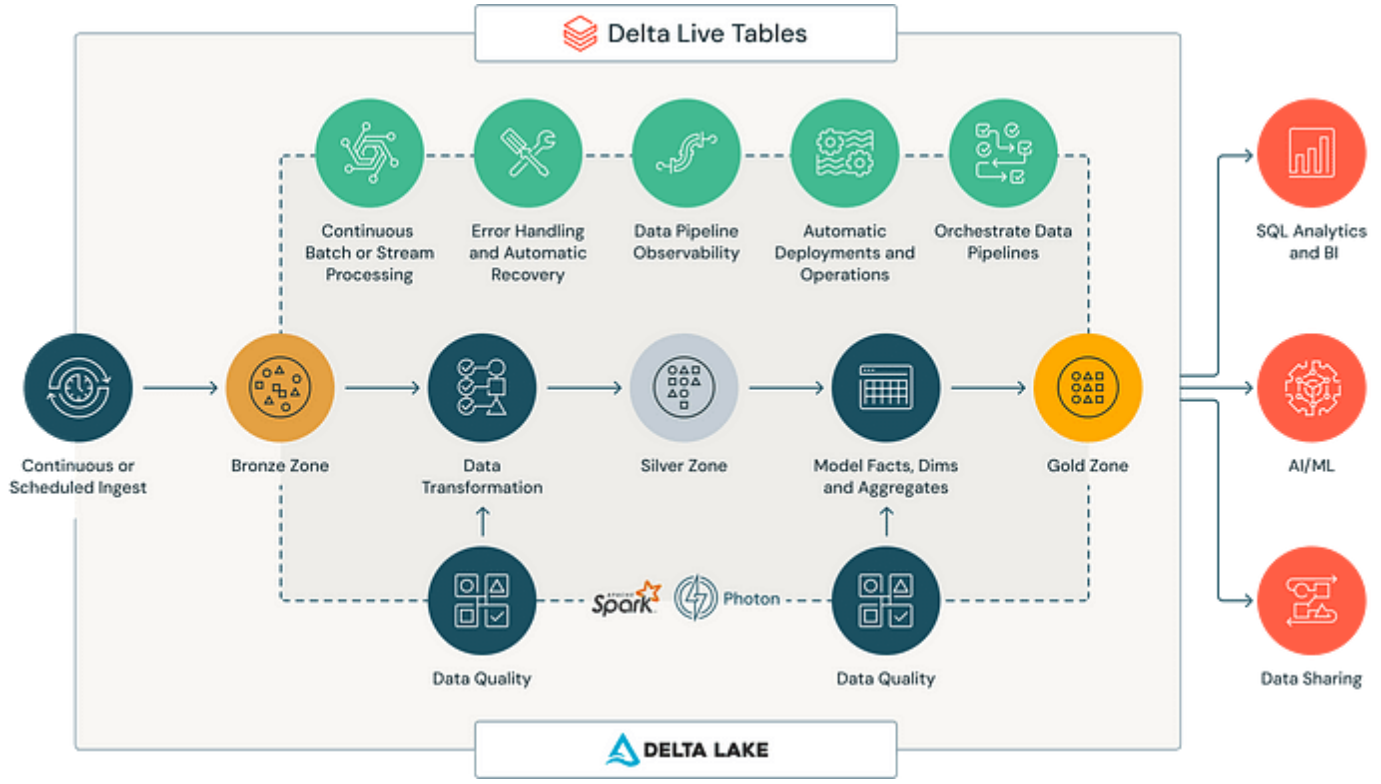
The DLT architecture begins with data ingestion from various sources, seamlessly bringing data into the processing pipeline.

Transformations

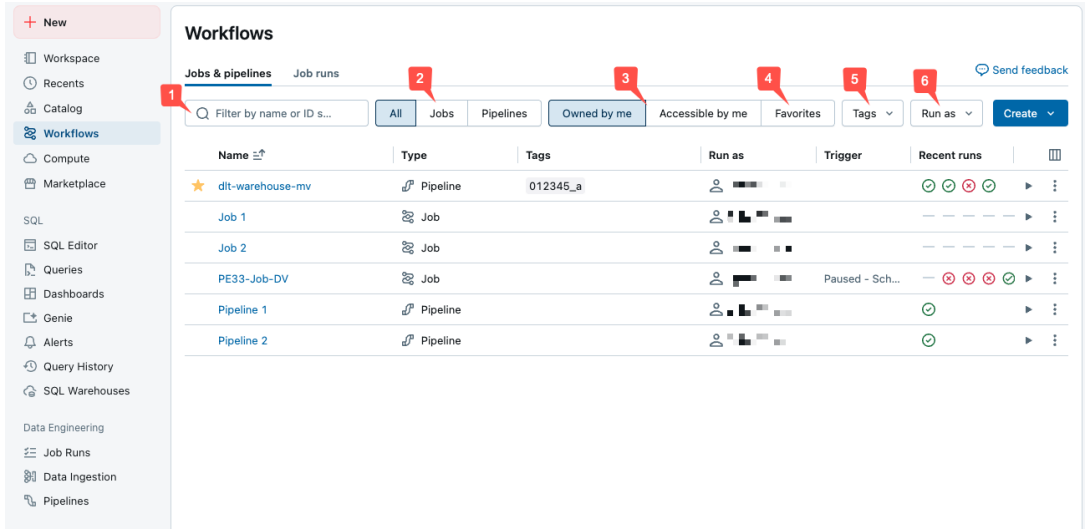
Data undergoes transformations within the DLT pipeline, allowing for data cleansing, enrichment, and preparation for analysis.

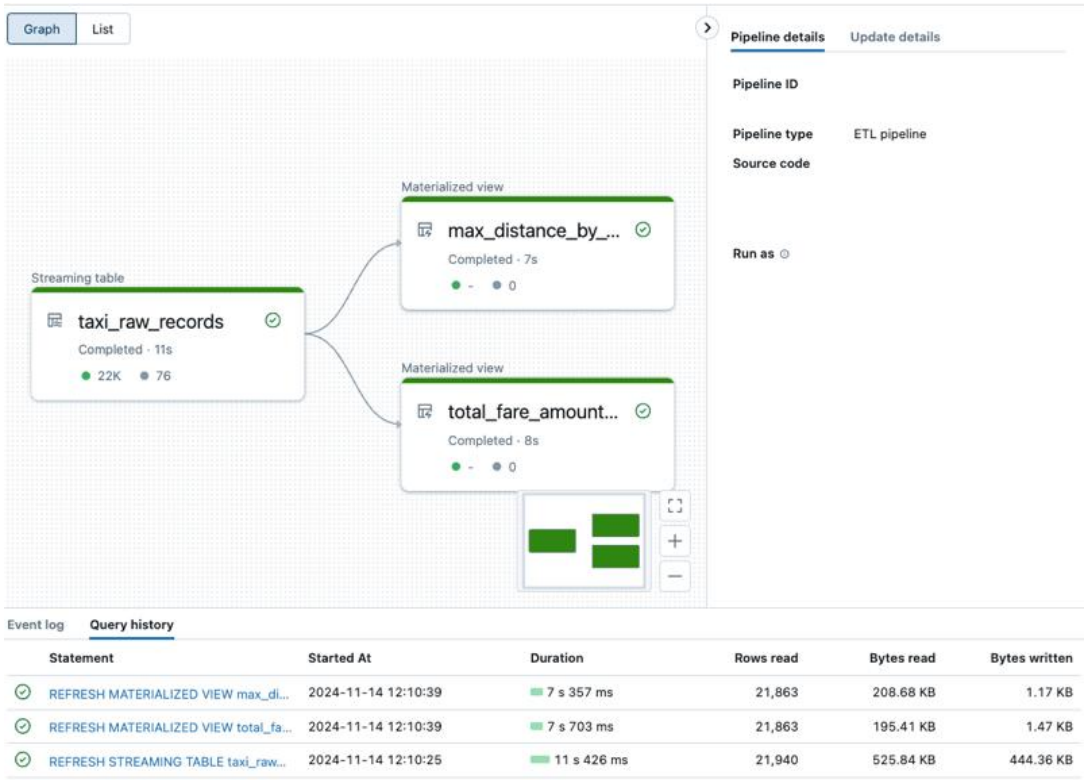
Publishing Data

After processing, the transformed data is published to Delta tables, making it available for analysis and reporting.



Monitoring and Trobleshooting





Pipeline Options Available in Databricks

Option	Description	When to Use	Key Pros	Key Cons
Delta Live Tables (DLT)	Managed framework for building reliable, tested ETL pipelines declaratively.	Preferred for complex pipelines with streaming, quality, monitoring needs.	Simplifies pipeline management, built-in quality, monitoring.	Requires learning new framework, some flexibility trade-offs.
Databricks Jobs (notebooks)	Schedule notebooks as jobs for running custom ETL or ML workflows.	Simple batch workflows, scheduled jobs, or ad-hoc runs.	Full control of code, flexible.	No built-in quality or lineage tracking. Must build manually.
Databricks Workflows	Orchestrate multiple jobs with dependencies, including multi-task workflows.	Complex orchestration of multiple jobs/tasks.	Supports task dependencies, retries, alerts.	Still requires manual pipeline code management.
Apache Spark Streaming / Structured Streaming	Use Spark APIs directly for streaming data processing inside notebooks or jobs.	Custom streaming pipelines with fine-grained control.	Very flexible, powerful streaming.	Requires building your own monitoring, checkpointing, and quality checks.
Third-party Orchestration (e.g., Apache Airflow, Azure Data Factory)	External orchestrators triggering Databricks jobs or notebooks.	Complex enterprise pipelines with cross-system dependencies.	Rich orchestration features outside Databricks.	Adds external system complexity and cost.

Hands-on Labs

Lab Overview

Medallion Architecture

Implementing a simple Medallion Architecture involves understanding the Bronze, Silver, and Gold layers of data processing.

Task Objective

The objective is to create a clear understanding of each layer within the Medallion Architecture.

Tools for Implementation

Databricks notebooks and Delta Live Tables are essential tools for implementing the Medallion Architecture.

Provided Data

The lab will utilize provided CSV files containing sales, customer demographics, and products data.

Delta Live Tables UI Navigation

Accessing Delta Live Tables

Learn how to access Delta Live Tables from the Databricks workspace for efficient data management.

Pipeline Creation and Configuration

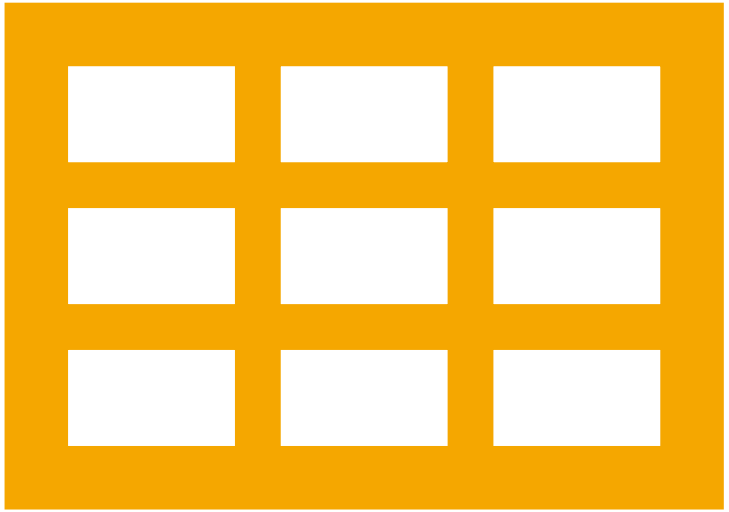
Walkthrough of creating and configuring pipelines in Delta Live Tables for data processing.

Monitoring and Management

Overview of how to monitor and manage pipeline performance and job runs within the UI.

Understanding UI Elements

Explore key UI elements including job runs, data freshness, and error logs for effective management.



MC ID: 7124506 | Parveen KR | Parveen.R@hotmail.com

Creating Basic DLT Pipelines

Creating a New DLT Pipeline

Begin by creating a new DLT pipeline, which is the foundation for managing data workflows.

Defining Tables

Define the Bronze, Silver, and Gold tables to categorize data at different processing levels.

Running the Pipeline

Once set up, run the pipeline and monitor its performance in real-time for effective data processing.

Validating Results

Finally, validate the results to ensure the data has been processed correctly throughout the pipeline.

Challenge Labs

Data Quality in Silver Layer

In this lab, participants will implement data quality expectations to ensure reliable data in the Silver layer.

Monthly Customer Segment Analysis

This lab focuses on creating a new Gold table for analyzing customer segments on a monthly basis.

Schema Evolution in Bronze Tables

Participants will learn how to handle schema evolution in Bronze tables effectively during this lab.