

Delta Live Tables

Value Proposition and Benefits



Executive Summary




Modern Data Analytics Platforms (DAP) have gone through rapid architectural pattern changes over the past few years, independent of the use cases, to provide maximum benefits to the consumers.

This point of view is aimed to provide insights into the benefits of the “Delta Live Tables” to enhance modern DAP architecture, based on our experience with transformational journeys.

Delta Lives Tables is not just a technological advancement but also focuses on achieving business objectives.

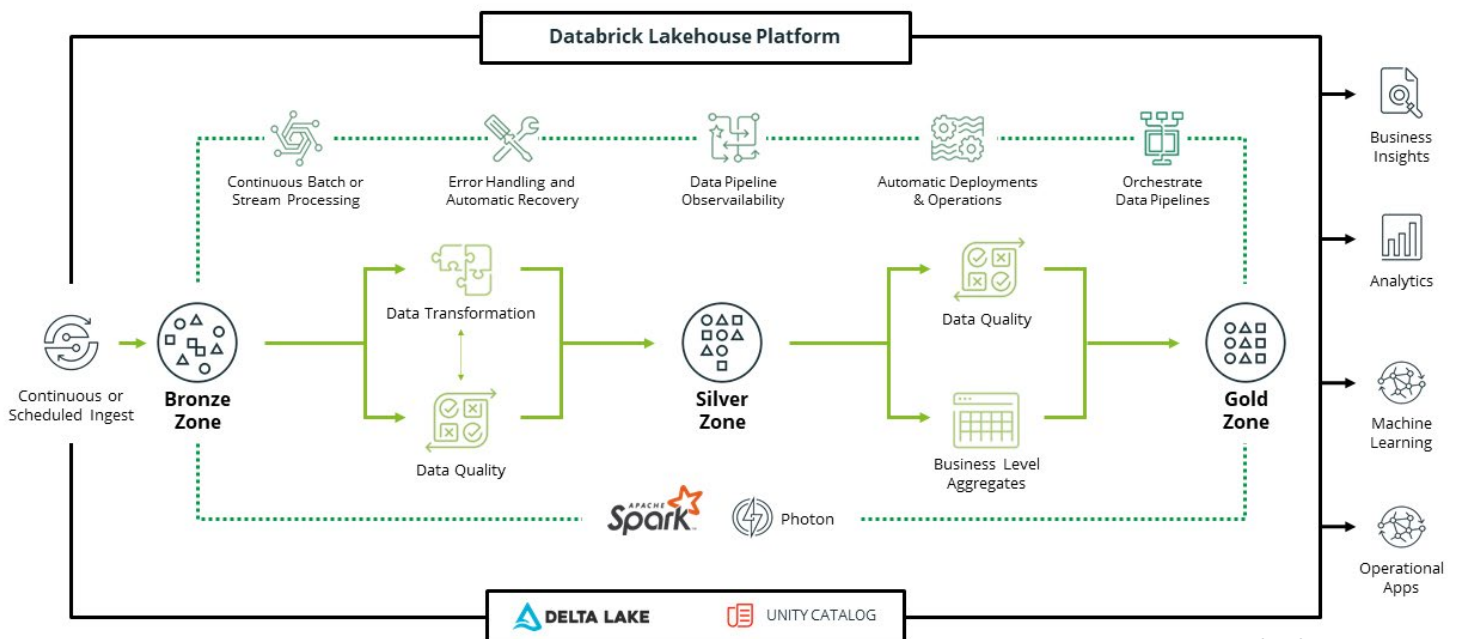
Operational efficiency translated into higher customer service which in turn translates into increased **Return On Investment (ROI)**.

Golden data is delivered to the business user in near real-time to:

-  Enable self-service analytics capability for faster time to market
-  Make faster and more reliable data driven decisions
-  Allow data science team to develop predictive analytics model and integrate with the data pipeline to predict the outcome before it becomes an issue

Delta Live Tables

Delta Live Tables (DLT) is an ETL framework that uses a simple declarative approach to build reliable data pipelines and automatically manage the infrastructure at scale, so data analysts and engineers can spend less time on tooling and focus on getting value from data.



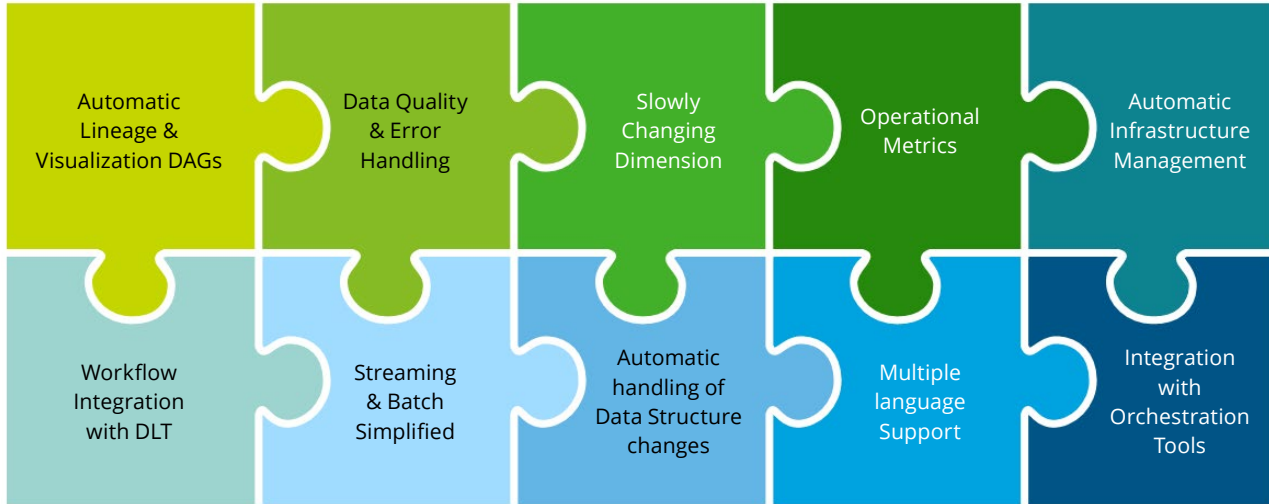
Source: Databricks

Without DLT, the development team would need to have deeper knowledge in ETL/ ELT, setting up streaming environment (queues, notifications, retries, avoid duplicate processing, storing operational metrics), schema management (enforcement, evolution, data quality, table management, data security, data modeling), and distributed computing techniques on how to scale processing with a managed infrastructure.

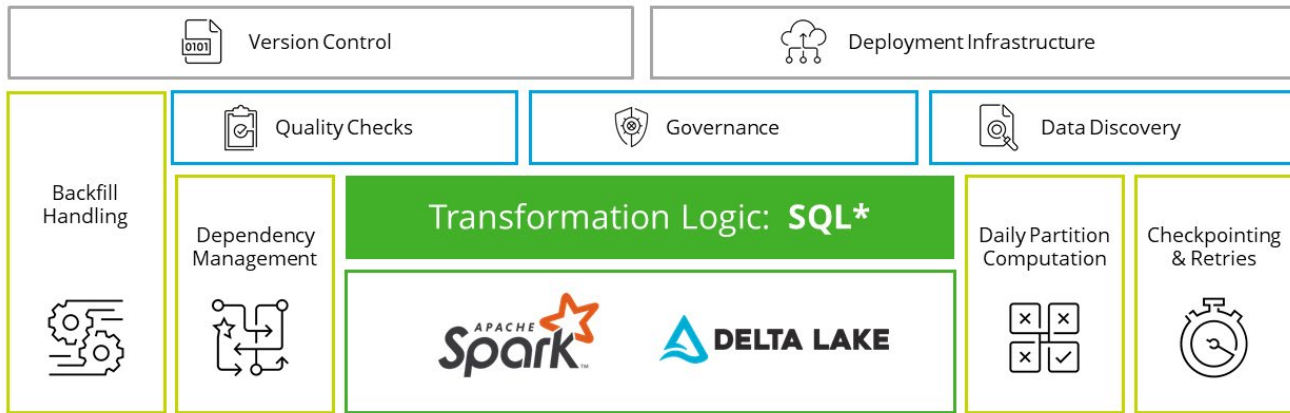
Value Proposition of Delta Live Tables

Delta Live Tables (DLT) offers many of the following benefits while continuously introducing new features in their roadmap. These benefits not only provide higher ROI but also accelerate ETL development, in addition to enabling easy maintenance and support.

Business and Technical Value Proposition



DLT allows the developer to focus on the core transformation logic by abstracting away all the surrounding operational complexity.



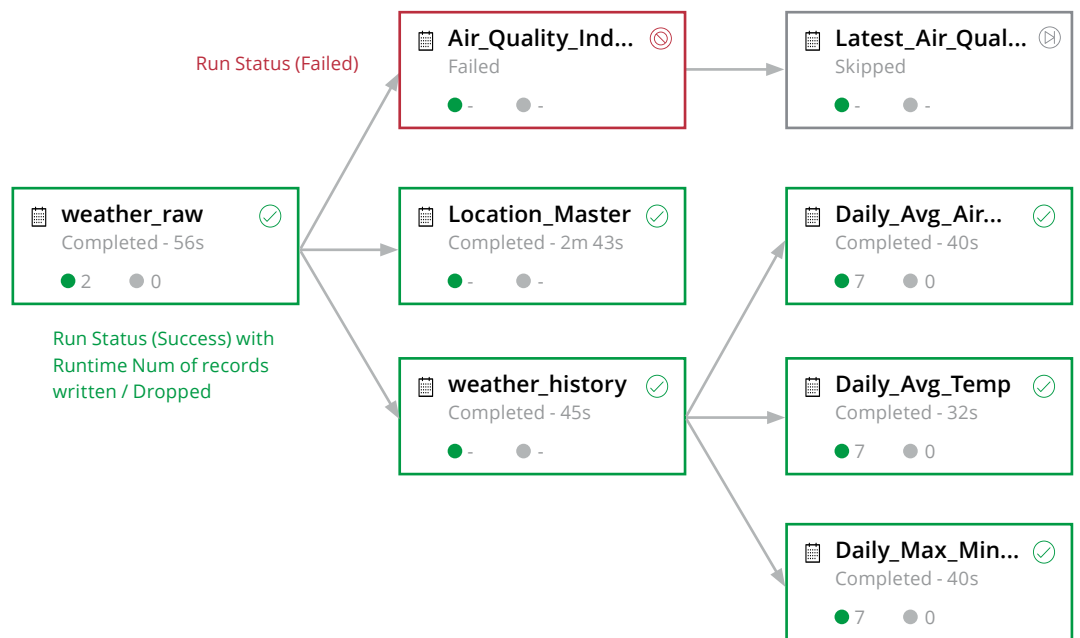
*SQL at the Core. Python is also supported

Automatic Lineage & Visualization DAGs

Developers can develop the code in any order within a single notebook or across notebooks. DLT automatically determines dependencies across the data pipeline and creates a visualization DAG while automatically checking table dependencies and syntax errors.

USE CASE(S)

- Identify syntax errors in the pipeline
- Identify table dependencies across notebooks
- Provide visual representation of lineage (observability)
- Simplify impact analysis on downstream systems and processes



DLT automatically creates the above DAG visualization, provisions infrastructure, executes the pipelines and makes all operational statistics available for the user.



Data Quality and Error Handling

DLT provides built-in quality controls, testing, monitoring and enforcement to help ensure that accurate Data Quality checks can be performed while data is streaming to fix, allow or fail, based on the severity of the data quality rules.

USE CASE(S)

- Flag bad records without stopping the process
- Drop invalid records if certain conditions are not met
- Fail the job if there are data quality issues
- Store rules and reuse them across multiple objects
- Capture and store runtime statistics of rows processed, failed and data quality expectations metrics including history (observability)

DLT simplifies implementation of multiple data quality checks without needing to develop complex code on either stream or batch data. It can also store DQ rules and apply them against multiple objects (Reusable/Portable). It helps prevent bad data from flowing into tables, measures data quality and provides tools to troubleshoot bad data.



Flag and Continue: Keep records that violate the expectation. Records that violate the expectation are added to the target dataset along with valid records



Drop Rows and Continue: Drop the records that violate the expectation



Fail and Rollback: Halt execution immediately when a record fails validation. If the operation is a table update, the system atomically rolls back the transaction

You can also define multiple expectations aka quality checks within a single statement.

Slowly Changing Dimension

No need to write complex SCD logic. Simplified functions take care of the complexity

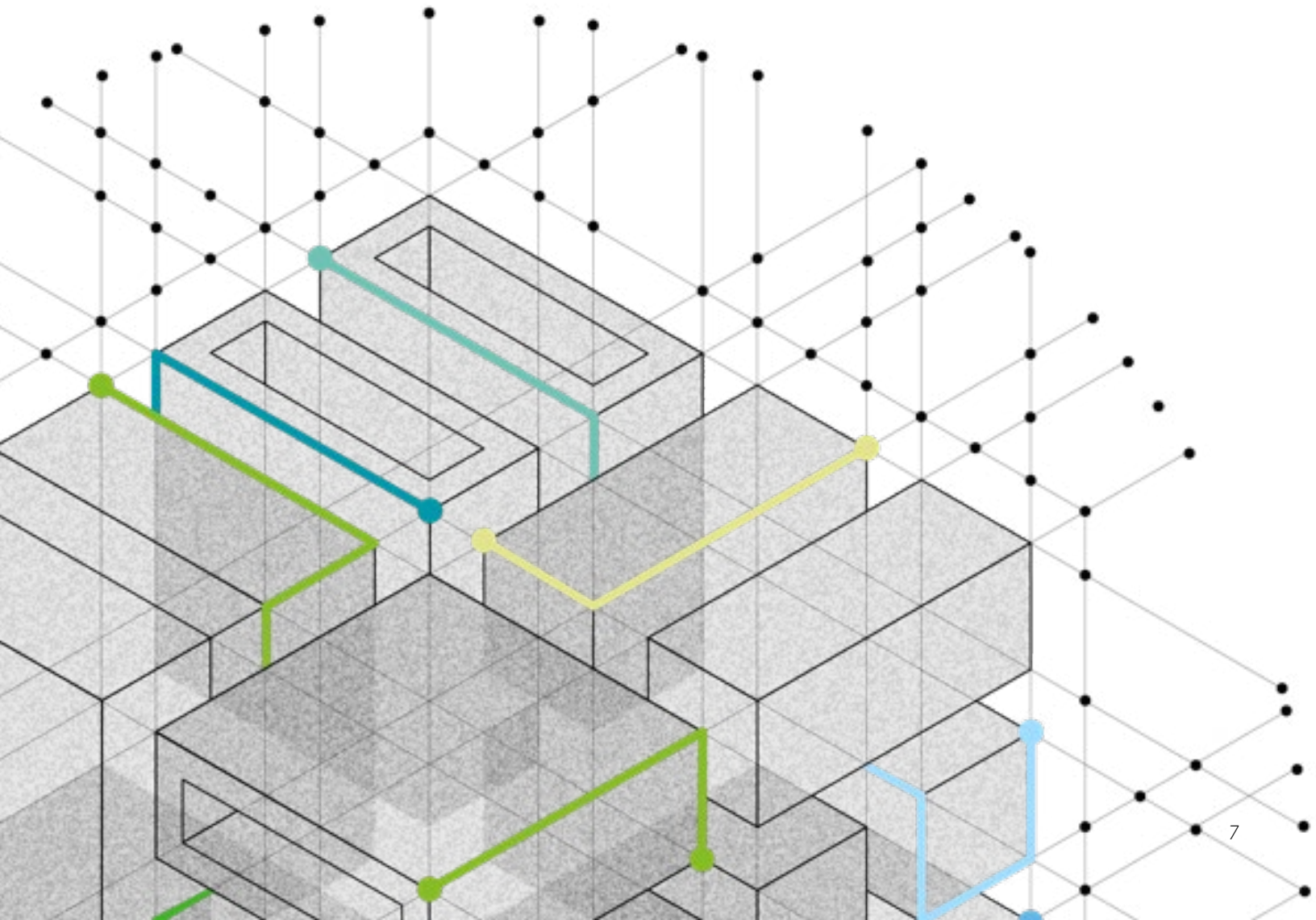
USE CASE(S)

Implement Type-1 SCD (Update records if exists; Insert otherwise) and SCD-2 (Expire old record and create new record for updates to maintain auditability / version; insert new records)

Handle out of order data arrival as part of SCD

Fill missing columns data from previously available data. Source systems may send partial update (few columns)

Implementing SCD using traditional ETL/ELT tools or SQL technology was complicated. Databricks simplifies this by hiding complex implementations behind simple specifications including handling of out of order events using a user specified sequence column.





Operational Metrics / Observability

Provides details of past run statistics at each table level, historical runs of pipelines over its life cycle, job success/failure status, individual and total run times, and an easy link to Spark UI/Ganglia charts, logs and operational metrics for performance tuning.

USE CASE(S)

- Monitor the state of the historical runs of data pipelines, runtime statistics such as records processed, rejected, table dependencies, run time, runtime being used etc.

- Observe trends in data quality and identify instances of data drift

- Provide easy to access links for Spark UI / Ganglia charts for better understanding of CPU and Memory usages and other operational metrics

Databricks stores all the event logs for your pipeline, monitors the state of your data pipelines and makes it available within its user interface, or even better through API calls.

DLT tracks table dependencies and provides a lineage diagram while storing all the data quality metrics in a log to understand data quality issue across your pipelines. DLT logs are exposed as a Delta table with all data expectation metrics to generate reports to monitor the data quality using Databricks SQL or BI tools of our choice.

DLT provides monitoring UIs that provide runtime statistics like number of rows processed, failed, as well as metrics on a per-expectation basis. Users can look at historical performance (previous runs), allowing them to track pipeline performance and data quality over time.



Automatic Infrastructure Management

Remove overhead by automating complex and time-consuming activities like task orchestration, error handling and recovery, auto-scaling, and performance optimization.

USE CASE(S)

Infrastructure to scale up/down, out/in automatically based on data volume without over or under provisioning, while getting optimal performance

Intelligent scaling of cluster in deciding when to scale up or down based on the amount of data being pulled out of event-based systems such as Kafka, EventHub, Kinesis streams etc.

Automatically retry the pipeline to better handle unexpected infrastructure failures

DLT automates operations and eliminates the challenges of server sizing. It handles failures with retries of the pipeline during infrastructure operational issues, and uses data volume and event queue details to optimize cluster scaling.

Workflow Integration with DLT

Complex logic can be written in a separate notebook but can be integrated with DLT through Job flow (Workflow)

USE CASE(S)

Integrate existing notebooks with DLT pipeline

Provide visibility into the data flow, table dependencies and aggregated data quality metrics across all data pipelines

Provide row level logging for operational, quality and status of the data pipeline

With Job flow, it is easy to define workflow with dependency to integrate the logic in traditional notebooks with DLT without needing to use a separate orchestration/scheduling tool.



Streaming and Batch Simplified

Architecture that allows developers to combine streaming and batch capability without needing to develop separate pipelines.

USE CASE(S)

Process the data as soon as it becomes available for improved speed to market and decision making; invoke machine learning models to predict the outcome while streaming

No need to setup, manage and support streaming technologies

Enable the business to develop streaming applications! (Streaming at the user's fingertips)

DLT makes it easier in integrating with Autoloader, efficiently streaming the data files as they arrive in real-time without setting up complex cloud services, and will not reprocess duplicate files. DLT automatically handles data that arrives out of order, implements all the data quality, business logic, calls predictive models and stores the data across all layers and makes the data available as soon as the data arrives. No more batch processing, scheduling or waiting.

Multiple language support

Capability to choose commonly available language for the ease of development.

USE CASE(S)

Analyst: Desire to use a common language such as SQL

Data Scientists and Data Engineers: Have flexibility to use Python to pull in more complex libraries and take advantage of the complexity that Python offers in addition to SQL

Users/Developers can leverage SQL or Python to build declarative pipelines – easily defining 'what' to do, not 'how' to do it.



Automatic handling of Data Structure Changes

When data sources produce change records that contain only a subset of the fields in the target table, and unmodified columns are represented as NULLs, Delta Live Tables can combine (coalesce) these partial updates into a single complete/up-to-date row.

USE CASE(S)

Source system table can change the structure, but application must not only be protected from those changes but also accommodate the changes

Provide flexibility in how to handle the changes, by adding columns, storing rescued data in its own column, failing the pipeline or ignoring schema changes

DLT handles these changes aka Schema Evolution, so there is no need to develop complicated DDL to evolve the schema of the target table. Instead, columns are automatically added to the target table, including complex/nested schemas.

Integration with Orchestration tools

DLT pipelines can be integrated with your enterprise orchestration / scheduling tool of choice.

USE CASE(S)

Invoke DLT pipelines from enterprise orchestration tools

Integrate custom developed Python UDF within DLT

Schedule pipeline data, either to run on a scheduled interval or continuously

DLT integrates very well with Apache Airflow or ADF (Azure Data Factory) while providing capabilities to run at a scheduled interval or continuously, so the data gets processed from Bronze->Silver->Gold layer as soon as the data arrives.



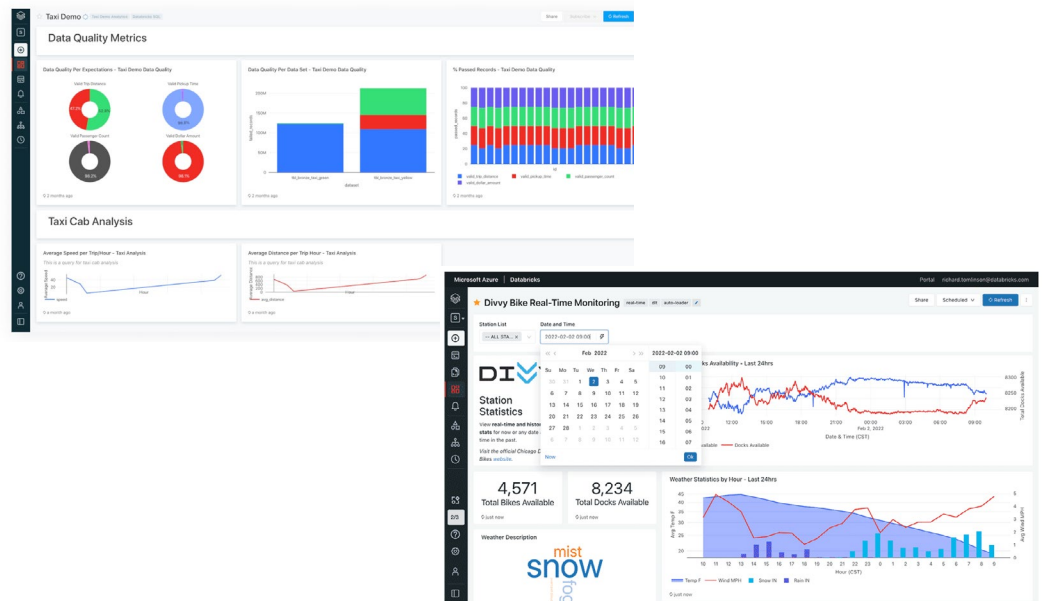
DLT Exposed for Reporting and Dashboards

DLT pipelines integrate with business intelligence tools for reporting and dashboards.

USE CASE(S)

Create near real-time / batch dashboards and reports against DLT tables with continuous data refresh

Spot anomalous behavior and display it visually as soon as it happens to avoid opportunity loss, equipment failures, security breaches and fraudulent transaction etc.



DLT exposed through SQL endpoints (Databricks Runtime / Servers) can be configured either as ad-hoc or scheduled running queries. Results can be exposed as Reports/Dashboards.

Recap

The power of DLT is not just meeting one or few use cases but all of the above use cases under one umbrella. Databricks' investment in making user friendly delta live table notebook development and new features that is expected to become GA will continue to make the development and business community excited to reap more benefits from the modern Lakehouse platform.

AUTHORS



Mani Kandasamy

Technology Fellow
Deloitte Consulting LLP
mkandasamy@deloitte.com

Mani is a technology executive with Deloitte Consulting LLP AI & Data Engineering Offering and leads @Scale cloud data modernization and analytics solutions for a global portfolio of Deloitte's clients.



Vijay Balasubramaniam

Partner Solutions Architect
Databricks
vijay.balasubramaniam@databricks.com

Vijay Balasubramaniam is a Partner Solutions Architect at Databricks. He leverages his expertise in data management to help partners and customers be successful in large-scale analytics initiatives.

Deloitte.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor. Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

About Deloitte

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. In the United States, Deloitte refers to one or more of the US member firms of DTTL, their related entities that operate using the "Deloitte" name in the United States and their respective affiliates. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see www.deloitte.com/about to learn more about our global network of member firms.