Name : Parth Nandedkar
Date : 13 Feb 2024
Topics : Azure Databricks SQL and Delta Lake
Batch : Data Engineering Batch-1

Azure Databricks SQL :

Unified Analytics Platform: Azure Databricks provides a unified analytics platform that supports both SQL queries and code in languages like Python, Scala, and R. This allows users to seamlessly switch between SQL and other programming languages based on their requirements.

Interactive SQL Queries: Users can write and execute SQL queries directly in Databricks notebooks or through the Databricks SQL endpoint. SQL queries can be used to interactively explore and analyze data stored in various formats and sources.

Support for Multiple Data Sources: Azure Databricks supports querying data from a wide range of data sources including Delta Lake, Apache Spark tables, relational databases (such as Azure SQL Database, Azure Synapse Analytics, or on-premises databases), and data lakes (such as Azure Data Lake Storage or AWS S3).

Performance Optimization: Databricks optimizes SQL queries for performance by leveraging the distributed computing capabilities of Apache Spark. Users can take advantage of features like query optimization, data caching, and partition pruning to improve query execution times.

Delta Lake: Delta Lake is an open-source storage layer that brings ACID transactions, schema enforcement, and data versioning to Apache Spark. Users can leverage Delta Lake to optimize SQL queries, ensure data consistency, and simplify data management tasks.

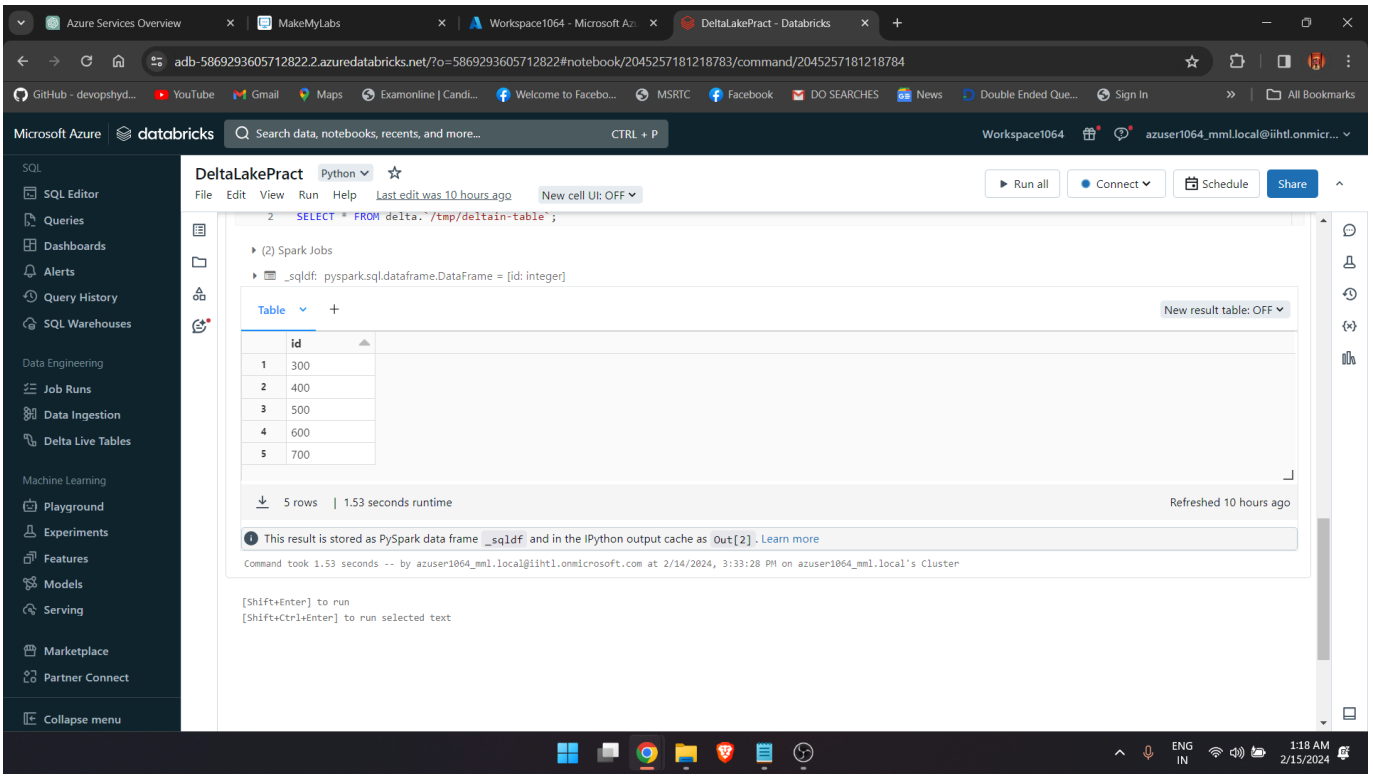
Advanced Analytics: SQL queries in Azure Databricks can be combined with advanced analytics techniques such as machine learning, graph processing, and streaming analytics. This enables users to perform complex analyses and derive insights from their data using SQL as well as other programming languages.

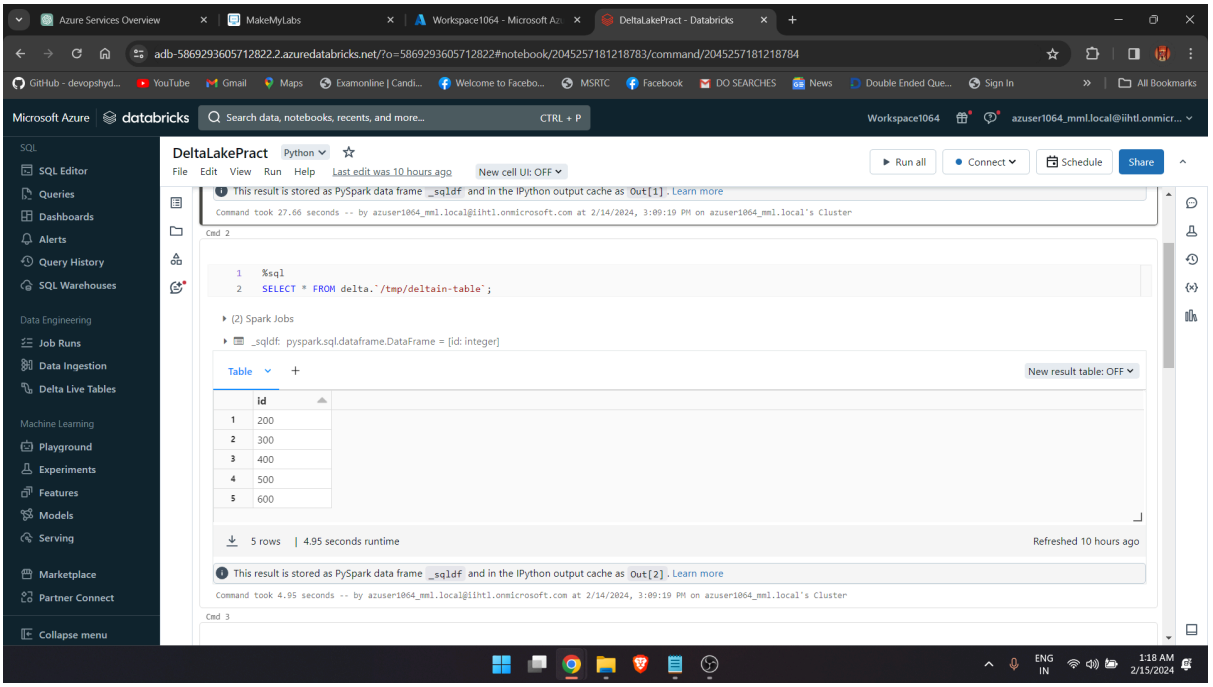
Integration with BI Tools: Azure Databricks can be integrated with popular business intelligence (BI) tools such as Power BI, Tableau, and Looker. Users can connect these tools to Databricks clusters and run SQL queries to visualize and analyze data directly within their BI dashboards and reports.

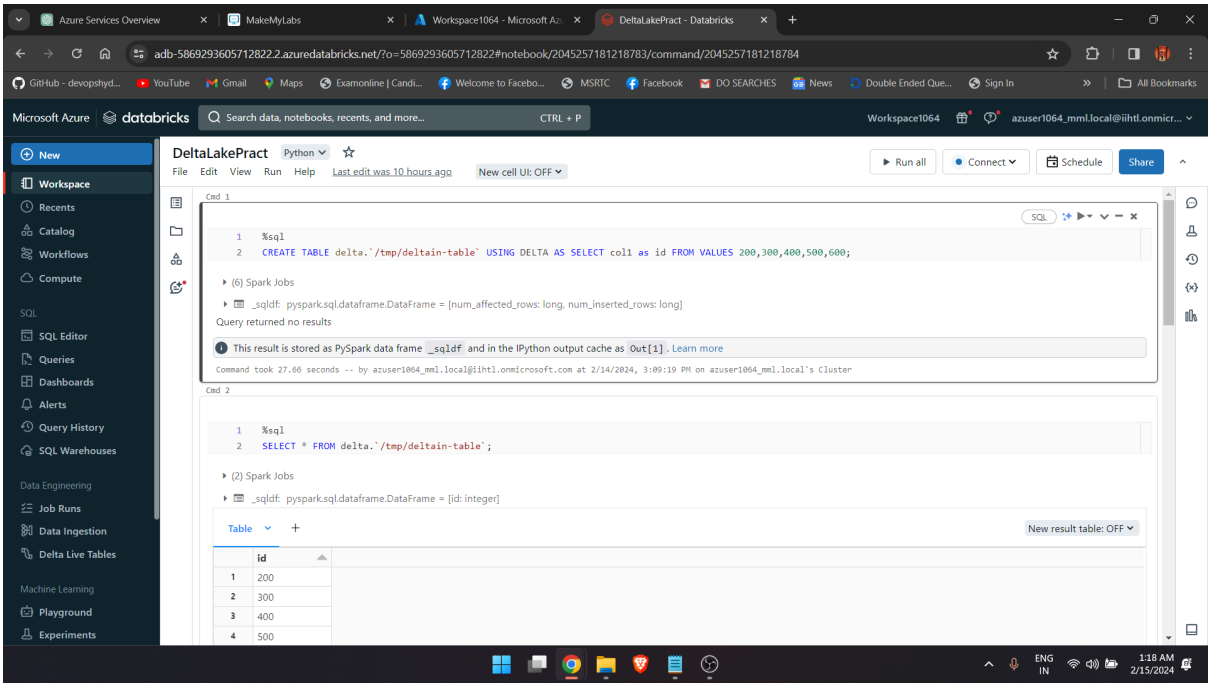
Security and Compliance: Azure Databricks provides robust security features to protect sensitive data and comply with regulatory requirements. This includes support for encryption, authentication, authorization, and audit logging for SQL queries and data access.

Scalability and Elasticity: Databricks offers scalable and elastic SQL query processing capabilities, allowing users to handle large volumes of data and dynamically scale compute resources based on workload demands.

Collaboration and Version Control: Users can collaborate on SQL queries and share their analyses with team members using Databricks notebooks. Notebooks support version control, enabling users to track changes, revert to previous versions, and collaborate effectively on SQL-based projects.




Azure Services Overview
MakeMyLabs
Workspace1064 - Microsoft Az
DeltaLakePract - Databricks
adb-5869293605712822.2.azuredatabricks.net/?o=5869293605712822#notebook/2045257181218783/command/2045257181218784
Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
Workspace1064
azuser1064_mml.local@iihtl.onmicr...
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Features
Models
Serving
Marketplace
Partner Connect
Collapse menu
DeltaLakePract
Python
File Edit View Run Help
Last edit was 10 hours ago
New cell UI: OFF
Run all
Connect
Schedule
Share
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[1]. Learn more
Command took 27.66 seconds -- by azuser1064_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:09:19 PM on azuser1064_mml.local's Cluster
Cmd 2
%sql
SELECT * FROM delta.`/tmp/deltain-table`;
(2) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]
Table
New result table: OFF
id
1 200
2 300
3 400
4 500
5 600
5 rows | 4.95 seconds runtime
Refreshed 10 hours ago
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[2]. Learn more
Command took 4.95 seconds -- by azuser1064_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:09:19 PM on azuser1064_mml.local's Cluster
Cmd 3
ENG
IN
1:18 AM
2/15/2024


Azure Services Overview
MakeMyLabs
Workspace1064 - Microsoft Az
DeltaLakePract - Databricks
adb-5869293605712822.2.azuredatabricks.net/?o=5869293605712822#notebook/2045257181218783/command/2045257181218784
Microsoft Azure
databricks
Search data, notebooks, recents, and more... CTRL + P
Workspace1064
azuser1064_mml.local@iihtl.onmicr...
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
DeltaLakePract
Python
File Edit View Run Help
Last edit was 10 hours ago
New cell UI: OFF
Run all
Connect
Schedule
Share
Cmd 1
SQL
%sql
CREATE TABLE delta.`/tmp/deltain-table` USING DELTA AS SELECT col1 as id FROM VALUES 200,300,400,500,600;
(6) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]
Query returned no results
This result is stored as PySpark data frame _sqldf and in the IPython output cache as Out[1]. Learn more
Command took 27.66 seconds -- by azuser1064_mml.local@iihtl.onmicrosoft.com at 2/14/2024, 3:09:19 PM on azuser1064_mml.local's Cluster
Cmd 2
%sql
SELECT * FROM delta.`/tmp/deltain-table`;
(2) Spark Jobs
_sqldf: pyspark.sql.dataframe.DataFrame = [id: integer]
Table
New result table: OFF
id
1 200
2 300
3 400
4 500
ENG
IN
1:18 AM
2/15/2024

Delta Lake :

Definition: Delta Lake is an open-source storage layer that brings ACID transactions, schema enforcement, and data versioning to Apache Spark and other big data processing engines. It's designed to address challenges associated with data lakes, such as data reliability, data consistency, and data quality.

ACID Transactions: Delta Lake provides support for ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring that operations like inserts, updates, and deletes are atomic and consistent.

Schema Enforcement: Delta Lake allows users to define and enforce schema on their data, ensuring that it adheres to predefined structures and data types. This helps maintain data quality and consistency over time.

Data Versioning: Delta Lake automatically versions the data as changes are made, allowing users to time-travel to previous versions of the data or rollback to specific points in time. This feature is valuable for auditing, compliance, and reproducibility.

Use Cases: Delta Lake is commonly used for building reliable data pipelines, implementing data lakes with transactional capabilities, enabling stream processing with exactly-once semantics, and improving data quality in big data environments.