**Assignment4**

* Delta Lake is the optimized storage layer that provides the foundation for storing data and tables in the Databricks lakehouse.
* Delta Lake is open source Service that extends Parquet data files with a file-based transaction log for ACID properties and scalable metadata handling.
* Delta Lake is fully compatible with Apache Spark APIs, and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale.
* Delta Lake is the default storage format for all operations on Databricks. Unless otherwise specified, all tables on Databricks are Delta tables.
* Databricks originally developed the Delta Lake protocol and continues to actively contribute to the open source project.
* Many of the optimizations and products in the Databricks platform build upon the guarantees provided by Apache Spark and Delta Lake.
* For information on optimizations on Databricks.

**Updating and modifying Delta Lake tables:**

Atomic transactions with Delta Lake provide many options for updating data and metadata. Databricks recommends you avoid interacting directly with data and transaction log files in Delta Lake file directories to avoid corrupting your tables.

* Delta Lake supports upserts using the merge operation.
* Delta Lake provides numerous options for services overrites based on filters and partitions.
* You can manually or automatically update your schema without rewriting data.
* Column Mapping enables columns to be renamed or deleted without rewriting data.

**Incremental and streaming workloads on Delta Lake:**

Delta Lake is optimized for Structured Streaming on Databricks. Delta Live Tables extends native capabilities with simplified infrastructure deployment, enhanced scaling, and managed data dependencies.

**Querying previous versions of a table:**

Each write to a Delta table creates a new table version. You can use the transaction log to review modifications to your table and query previous table versions.

**Managing files and indexing data with Delta Lake:**

* Databricks sets many default parameters for Delta Lake that impact the size of data files and number of table versions that are retained in history.
* Delta Lake uses a combination of metadata parsing and physical data layout to reduce the number of files scanned to fulfill any query.

**Data pipelines using Delta Lake and Delta Live Tables:**

Databricks encourages users to leverage a medallion architecture to process data through a series of tables as data is cleaned and enriched. Delta Live Tables simplifies ETL workloads through optimized execution and automated infrastructure deployment and scaling.