

Quiz -- Day 1

Modules: M01 (Platform & Workspace), M02 (ELT Ingestion), M03 (Delta Fundamentals)

Format: 20 questions, single correct answer (A-D)

Time: ~15-20 min

Write your answers in the table at the end, then check against the Answer Key at the bottom.

Q1

Which component of the Databricks Lakehouse Platform serves as the centralized governance layer for data, users, and permissions?

- A. Databricks Runtime
- B. Delta Lake
- C. Unity Catalog
- D. Photon Engine

Q2

In Unity Catalog, what is the correct hierarchy of securable objects from broadest to most granular?

- A. Schema > Catalog > Table
- B. Catalog > Schema > Table
- C. Metastore > Schema > Catalog
- D. Workspace > Catalog > Table

Q3

What is required before you can access a table inside a Unity Catalog schema?

- A. Only `SELECT` on the table
- B. `USE CATALOG` on the catalog and `USE SCHEMA` on the schema
- C. `ALL PRIVILEGES` on the catalog
- D. Workspace admin role only

Q4

Which SQL command reads data from a CSV file in a Unity Catalog Volume and creates a managed Delta table?

A.

`LOAD DATA INPATH '/path' INTO TABLE t`

B.

`COPY INTO t FROM '/path'`

C.

`CREATE TABLE t AS SELECT * FROM read_files('/path', format => 'csv')`

D.

`INSERT INTO t VALUES (SELECT * FROM '/path')`

Q5

Which statement about managed vs external tables in Unity Catalog is correct?

- A. Managed tables store data in a user-specified cloud location
- B. External tables have their data deleted when the table is dropped
- C. Managed tables store data in the catalog's managed storage and data is deleted on DROP
- D. External tables cannot be registered in Unity Catalog

Q6

A data engineer wants to apply SQL transformations on a DataFrame `df`. Which is the correct sequence?

A.

`df.createOrReplaceTempView("v")
spark.sql("SELECT * FROM v")`

B.

`df.registerTable("v")
spark.sql("SELECT * FROM v")`

C.

`df.saveAsTable("v")
spark.sql("SELECT * FROM v")`

D.

`spark.sql("CREATE VIEW v AS df")`

Q7

What does Delta Lake's transaction log guarantee?

- A. Exactly-once semantics for batch reads only
- B. ACID transactions, time travel, and schema enforcement
- C. Automatic data compression only
- D. Distributed caching of query results

Q8

Which file format does Delta Lake use to store data?

- A. JSON
- B. ORC
- C. Parquet
- D. Avro

Q9

A `DESCRIBE HISTORY` command on a Delta table shows 5 versions. What does each version represent?

- A. A schema change only
- B. A transactional operation (write, update, delete, merge, optimize, etc.)
- C. A user session
- D. A scheduled job execution

Q10

A data engineer runs the following query. What does it return?

`SELECT * FROM my_table VERSION AS OF 3`

- A. The current table filtered to 3 rows
- B. The table as it existed at transaction version 3
- C. The 3rd partition of the table
- D. An error because time travel is not supported

Q11

What is the default compute type for new Databricks workspaces (as of late 2025)?

- A. Classic single-node cluster
- B. Serverless compute

- C. GPU cluster
- D. High-concurrency cluster

Q12

Which of the following is the recommended way to store non-tabular files (CSV, JSON, images) in Unity Catalog?

- A. DBFS root
- B. DBFS mounts
- C. Unity Catalog Volumes
- D. Cluster local storage

Q13

What is the difference between ELT and ETL in the context of Databricks Lakehouse?

- A. ELT loads raw data first then transforms in-place; ETL transforms data before loading
- B. ELT is for batch only; ETL is for streaming
- C. ELT requires external tools; ETL is native
- D. They are the same process

Q14

A data engineer has a DataFrame with duplicate rows. Which PySpark method removes exact duplicates based on a subset of columns?

A.

```
df.distinct()
```

B.

```
df.dropDuplicates(["customer_id", "email"])
```

C.

```
df.removeDuplicates(["customer_id"])
```

D.

```
df.unique(["customer_id"])
```

Q15

What is the difference between a Temporary View and a Permanent View in Databricks?

- A. Temporary views are stored in Unity Catalog; permanent views are not

- B. Temporary views exist only within the SparkSession; permanent views are persisted in the metastore
- C. Temporary views support SQL only; permanent views support PySpark only
- D. There is no difference

Q16

What does the `%run` magic command do in a Databricks notebook?

- A. Runs an external Python script on the driver node
- B. Executes another notebook in the same execution context, sharing variables and functions
- C. Runs a SQL query and returns results
- D. Restarts the cluster and runs all cells

Q17

When writing a DataFrame to a Delta table, what happens if the incoming data has a column not present in the target schema?

Default behavior (schema enforcement):

```
df.write.mode("append").saveAsTable("my_table")
```

With schema evolution:

```
df.write.option("mergeSchema",  
"true").mode("append").saveAsTable("my_table")
```

- A. The new column is silently ignored
- B. The write fails with a schema mismatch error (schema enforcement)
- C. The new column is automatically added to the table
- D. The entire table is recreated with the new schema

Q18

Which `dbutils` submodule provides methods for listing, copying, and removing files?

A.

```
dbutils.fs.ls("/path")
```

B.

```
dbutils.data.list("/path")
```

C.

```
dbutils.files/browse("/path")
```

D.

```
dbutils.storage.ls("/path")
```

Q19

What does `RESTORE TABLE my_table TO VERSION AS OF 2` do?

- A. Creates a copy of version 2 as a new table
- B. Reverts the table to its state at version 2, creating a new version in the history
- C. Deletes all versions after version 2
- D. Displays the table at version 2 without modifying it

Q20

Which statement correctly reads a JSON file with an explicit schema in PySpark?

A.

```
df = spark.read.json("/path/data.json", schema="id INT, name STRING")
```

B.

```
from pyspark.sql.types import StructType, StructField, IntegerType,  
StringType  
schema = StructType([StructField("id", IntegerType()), StructField("name",  
StringType())])  
df = spark.read.schema(schema).json("/path/data.json")
```

C.

```
df = spark.read.format("json").option("schema", "id:int,  
name:string").load("/path")
```

D.

```
df = spark.read.json("/path").cast("id INT, name STRING")
```

Your Answers

Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10

Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20

Score: ___ / 20

Scroll down for Answer Key

Answer Key -- Day 1

#	Ans	Explanation
Q1	C	Unity Catalog provides centralized access control, auditing, lineage, and data discovery across workspaces.
Q2	B	The hierarchy is: Metastore > Catalog > Schema > Table/View/Function.
Q3	B	Unity Catalog requires explicit <code>USE CATALOG</code> + <code>USE SCHEMA</code> grants before accessing objects within.
Q4	C	<code>CREATE TABLE ... AS SELECT * FROM read_files()</code> (CTAS with <code>read_files</code>) is the standard pattern for loading files into managed Delta tables.
Q5	C	Managed tables use catalog-managed storage; dropping them deletes the underlying data. External tables reference external locations and survive DROP.
Q6	A	<code>createOrReplaceTempView()</code> registers a DataFrame as a session-scoped temporary view queryable via <code>spark.sql()</code> .
Q7	B	The Delta transaction log (<code>_delta_log</code>) provides ACID guarantees, enables time travel, and enforces schema on write.
Q8	C	Delta Lake stores data as Parquet files with an additional JSON-based transaction log.
Q9	B	Each version in Delta History corresponds to a single transactional operation on the table.
Q10	B	<code>VERSION AS OF</code> enables time travel, returning the table snapshot at the specified version number.
Q11	B	Since late 2025, Serverless compute is the default compute type for new Databricks workspaces.

#	Ans	Explanation
Q12	C	Unity Catalog Volumes are the recommended replacement for DBFS root/mounts for storing non-tabular files.
Q13	A	ELT (Extract-Load-Transform) loads raw data into the lakehouse first, then transforms in-place using Spark. ETL transforms before loading. Databricks favors ELT.
Q14	B	<code>dropDuplicates(["col1", "col2"])</code> removes duplicate rows based on a subset of columns. <code>distinct()</code> works on all columns.
Q15	B	Temporary views live only in the SparkSession (lost on cluster restart). Permanent views are persisted in Unity Catalog.
Q16	B	<code>%run</code> executes another notebook in the same context, making its variables and functions available to the calling notebook.
Q17	B	By default, Delta enforces schema -- the write fails. Use <code>mergeSchema=true</code> to enable schema evolution.
Q18	A	<code>dbutils.fs</code> provides file system utilities: <code>ls()</code> , <code>cp()</code> , <code>rm()</code> , <code>head()</code> , <code>mkdirs()</code> .
Q19	B	<code>RESTORE TABLE</code> reverts the table to a previous state and creates a new version entry in the history log.
Q20	B	Define a <code>StructType</code> schema and pass it via <code>.schema(schema)</code> before calling <code>.json()</code> .