Contents

[Question 1 4](#_Toc155540727)

[Question 2 5](#_Toc155540728)

[Question 3 7](#_Toc155540729)

[Question 4 9](#_Toc155540730)

[Question 5 11](#_Toc155540731)

[Question 6 13](#_Toc155540732)

[Question 7 14](#_Toc155540733)

[Question 8 16](#_Toc155540734)

[Question 9 17](#_Toc155540735)

[Question 10 18](#_Toc155540736)

[Question 11 19](#_Toc155540737)

[Question 12 20](#_Toc155540738)

[Question 13 20](#_Toc155540739)

[Question 14 22](#_Toc155540740)

[Question 15 23](#_Toc155540741)

[Question 16 24](#_Toc155540742)

[Question 17 25](#_Toc155540743)

[Question 18 27](#_Toc155540744)

[Question 19 28](#_Toc155540745)

[Question 20 29](#_Toc155540746)

[Question 21 30](#_Toc155540747)

[Question 22 32](#_Toc155540748)

[Question 23 33](#_Toc155540749)

[Question 24: 34](#_Toc155540750)

[Question 25 37](#_Toc155540751)

[Question 26 39](#_Toc155540752)

[Question 27 40](#_Toc155540753)

[Question 28 41](#_Toc155540754)

[Question 29 42](#_Toc155540755)

[Question 30 43](#_Toc155540756)

[Question 31 44](#_Toc155540757)

[Question 32 46](#_Toc155540758)

[Question 33 48](#_Toc155540759)

[Question 34 50](#_Toc155540760)

[Question 35 52](#_Toc155540761)

[Question 36 53](#_Toc155540762)

[Question 37 54](#_Toc155540763)

[Question 38 55](#_Toc155540764)

[Question 39 60](#_Toc155540765)

[Question 40: 61](#_Toc155540766)

[Question 41 66](#_Toc155540767)

[Question 42 69](#_Toc155540768)

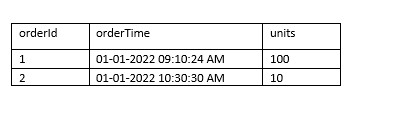
[Question 43 70](#_Toc155540769)

[Question 44 71](#_Toc155540770)

[Question 45 72](#_Toc155540771)

## Question 1

You were asked to create a table that can store the below data, orderTime is a timestamp but the finance team when they query this data normally prefer the orderTime in date format, you would like to create a calculated column that can convert the orderTime column timestamp datatype to date and store it, fill in the blank to complete the DDL.

****

1. CREATE TABLE orders (
2. orderId int,
3. orderTime timestamp,
4. orderdate date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ,
5. units int)

* **AS DEFAULT (CAST(orderTime as DATE))**
* **GENERATED ALWAYS AS (CAST (orderTime as DATE))**
* **GENERATED DEFAULT AS (CAST(orderTime as DATE))**
* **AS (CAST(orderTime as DATE))**
* **Delta lake does not support calculated columns, value should be inserted into the table as part of the ingestion process**

**Explanation**

The answer is, GENERATED ALWAYS AS (CAST(orderTime as DATE))

<https://docs.microsoft.com/en-us/azure/databricks/delta/delta-batch#--use-generated-columns>

Delta Lake supports generated columns which are a special type of columns whose values are automatically generated based on a user-specified function over other columns in the Delta table. When you write to a table with generated columns and you do not explicitly provide values for them, Delta Lake automatically computes the values.

**Note: Databricks also supports partitioning using generated column**

## Question 2

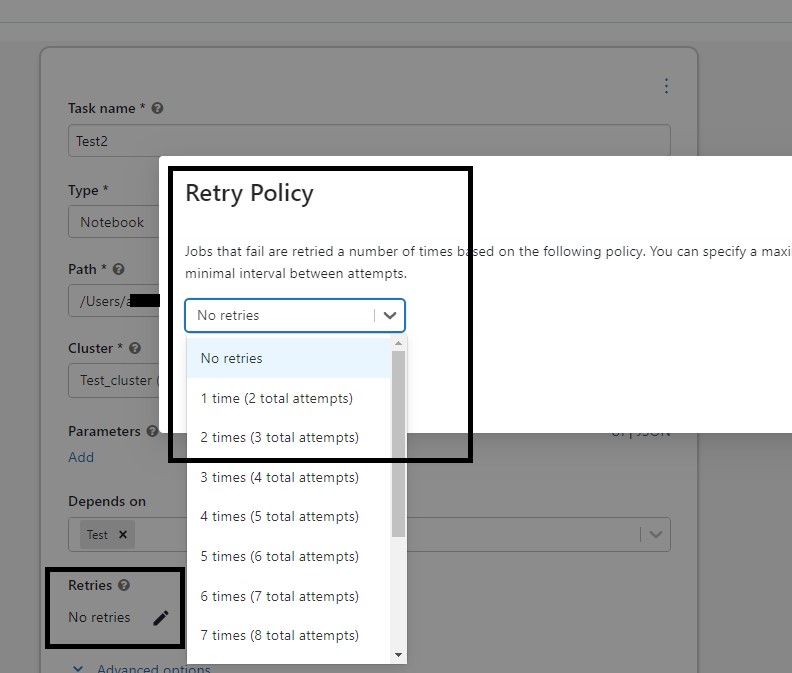
The data engineering team noticed that one of the job fails randomly as a result of using spot instances, what feature in Jobs/Tasks can be used to address this issue so the job is more stable when using spot instances?

* **Use Databrick REST API to monitor and restart the job**
* **Use Jobs runs, active runs UI section to monitor and restart the job**
* **Add second task and add a check condition to rerun the first task if it fails**
* **Restart the job cluster, job automatically restarts**
* **Add a retry policy to the task**

**Explanation**

The answer is,  Add a retry policy to the task

Tasks in Jobs support Retry Policy, which can be used to retry a failed tasks,  especially when using spot instance it is common to have failed executors or driver.



Bottom of Form

## Question 3

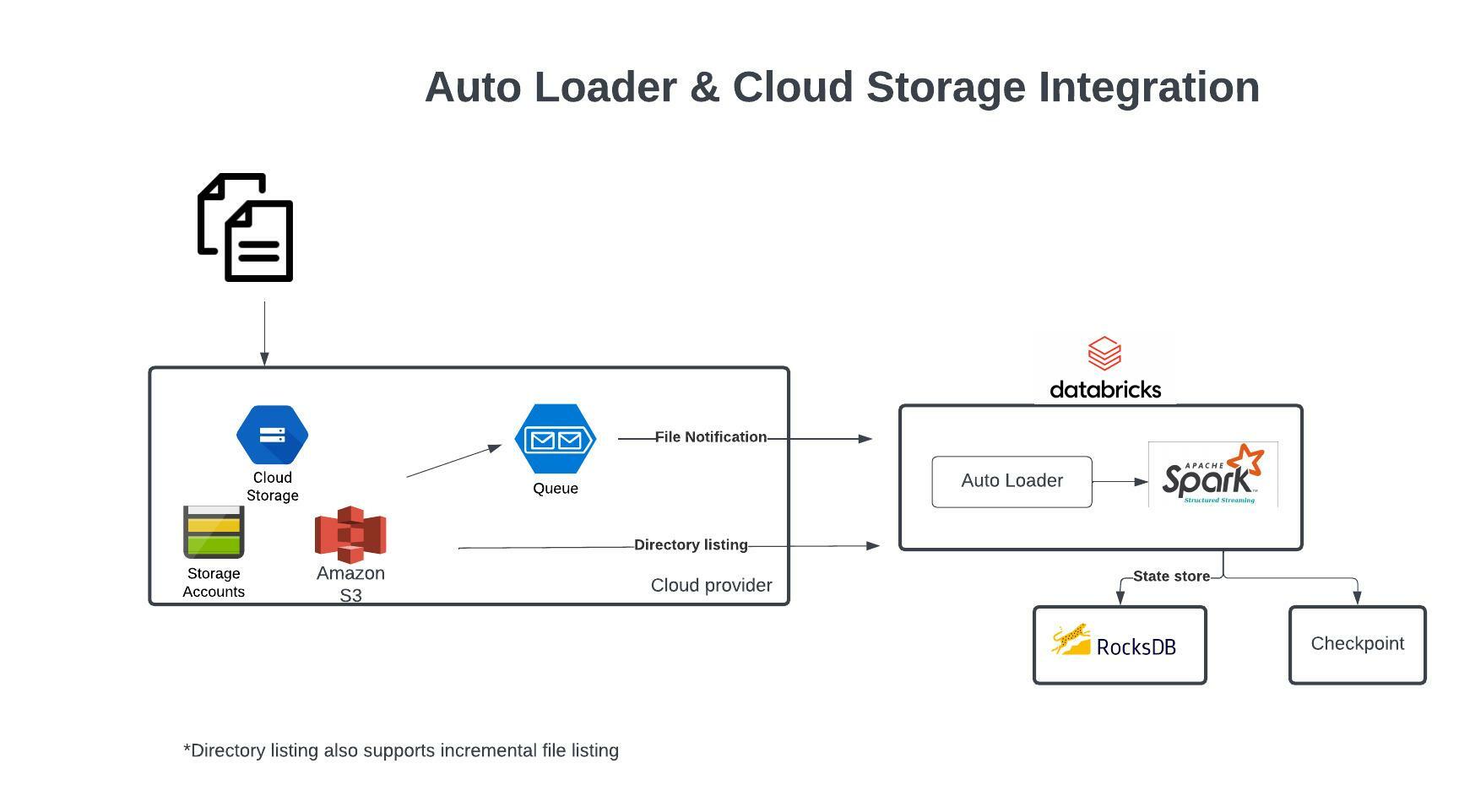
What is the main difference between AUTO LOADER and COPY INTO?

* **COPY INTO supports schema evolution.**
* **AUTO LOADER supports schema evolution.**
* **COPY INTO supports file notification when performing incremental loads.**
* **AUTO LOADER supports reading data from Apache Kafka**
* **AUTO LOADER Supports file notification when performing incremental loads.**

**Explanation**

Auto loader supports both directory listing and file notification but COPY INTO only supports directory listing.

Auto loader file notification will automatically set up a notification service and queue service that subscribe to file events from the input directory in cloud object storage like Azure blob storage or S3. File notification mode is more performant and scalable for large input directories or a high volume of files.



Auto Loader and Cloud Storage Integration

Auto Loader supports a couple of ways to ingest data incrementally

1. Directory listing - List Directory and maintain the state in RocksDB, supports incremental file listing
2. File notification - Uses a trigger+queue to store the file notification which can be later used to retrieve the file, unlike Directory listing File notification can scale up to millions of files per day.

**[OPTIONAL]**

**Auto Loader vs COPY INTO?**

**Auto Loader**

Auto Loader incrementally and efficiently processes new data files as they arrive in cloud storage without any additional setup. Auto Loader provides a new Structured Streaming source called cloudFiles. Given an input directory path on the cloud file storage, the cloudFiles source automatically processes new files as they arrive, with the option of also processing existing files in that directory.

When to use Auto Loader instead of the COPY INTO?

* You want to load data from a file location that contains files in the order of millions or higher. Auto Loader can discover files more efficiently than the COPY INTO SQL command and can split file processing into multiple batches.
* You do not plan to load subsets of previously uploaded files. With Auto Loader, it can be more difficult to reprocess subsets of files. However, you can use the COPY INTO SQL command to reload subsets of files while an Auto Loader stream is simultaneously running.

Auto loader file notification will automatically set up a notification service and queue service that subscribe to file events from the input directory in cloud object storage like Azure blob storage or S3. File notification mode is more performant and scalable for large input directories or a high volume of files.

Here are some additional notes on when to use COPY INTO vs Auto Loader

When to use COPY INTO

<https://docs.databricks.com/delta/delta-ingest.html#copy-into-sql-command>

When to use Auto Loader

<https://docs.databricks.com/delta/delta-ingest.html#auto-loader>

## Question 4

Why does AUTO LOADER require schema location?

* **Schema location is used to store user provided schema**
* **Schema location is used to identify the schema of target table**
* **AUTO LOADER does not require schema location, because its supports Schema evolution**
* **Schema location is used to store schema inferred by AUTO LOADER**
* **Schema location is used to identify the schema of target table and source table**

**Explanation**

The answer is, Schema location is used to store schema inferred by AUTO LOADER, so the next time AUTO LOADER runs faster as does not need to infer the schema every single time by trying to use the last known schema.

Auto Loader samples the first 50 GB or 1000 files that it discovers, whichever limit is crossed first. To avoid incurring this inference cost at every stream start up, and to be able to provide a stable schema across stream restarts, you must set the option cloudFiles.schemaLocation. Auto Loader creates a hidden directory \_schemas at this location to track schema changes to the input data over time.

The below link contains detailed documentation on different options

[Auto Loader options | Databricks on AWS](https://docs.databricks.com/ingestion/auto-loader/options.html)

## Question 5

Which of the following statements are incorrect about the lakehouse

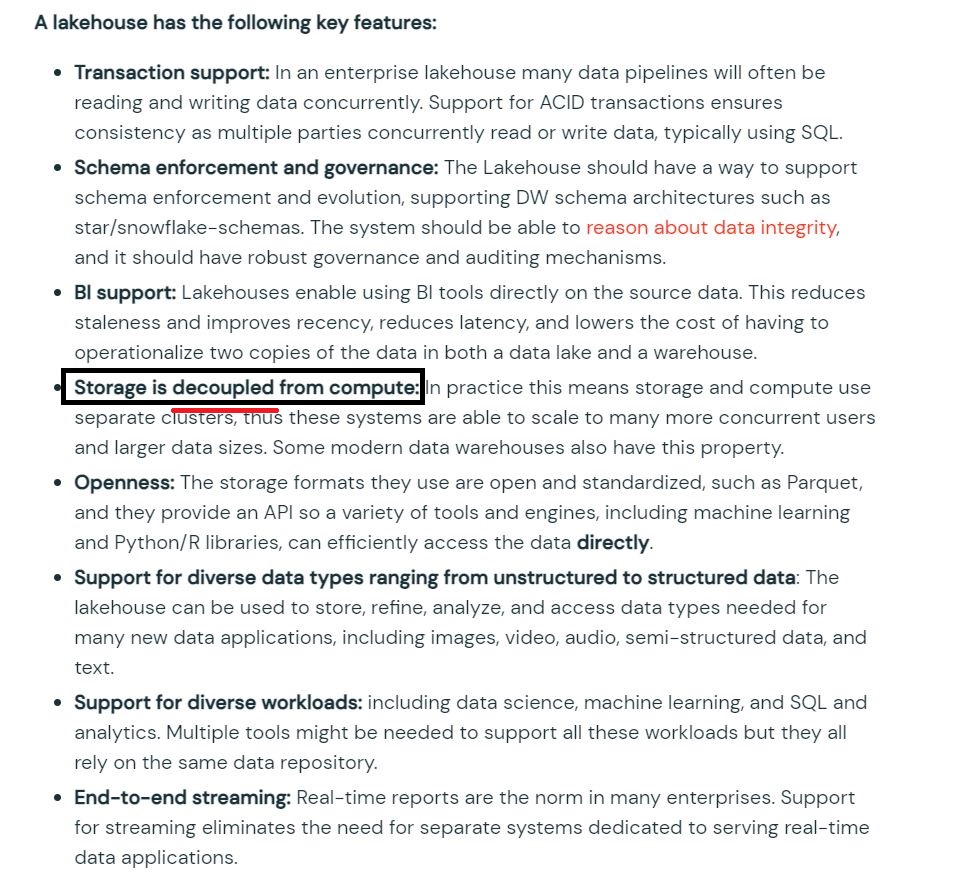
* **Support end-to-end streaming and batch workloads**
* **Supports ACID**
* **Support for diverse data types that can store both structured and unstructured**
* **Supports BI and Machine learning**
* **Storage is coupled with Compute**

**Explanation**

The answer is,  Storage is coupled with Compute.

The question was asking what is the incorrect option, in Lakehouse Storage is decoupled with compute so both can scale independently.

[What Is a Lakehouse? - The Databricks Blog](https://databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html)



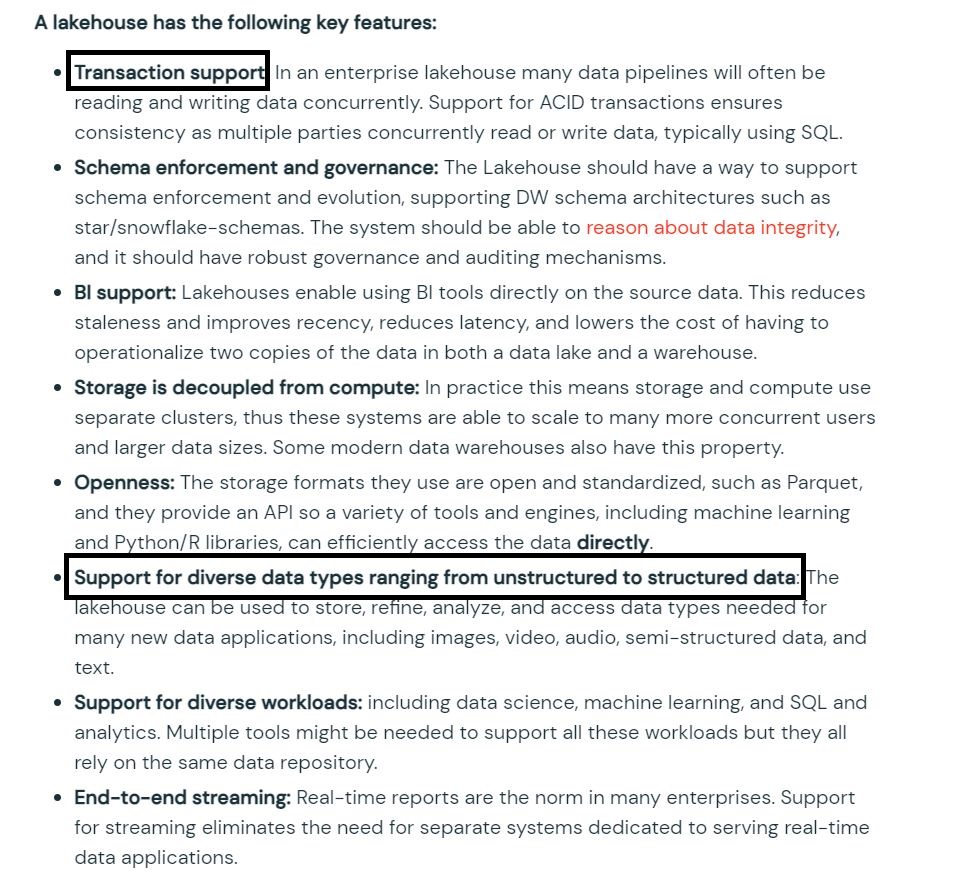
## Question 6

You are designing a data model that works for both machine learning using images and Batch ETL/ELT workloads. Which of the following features of data lakehouse can help you meet the needs of both workloads?

* **Data lakehouse requires very little data modeling.**
* **Data lakehouse combines compute and storage for simple governance.**
* **Data lakehouse provides autoscaling for compute clusters.**
* **Data lakehouse can store unstructured data and support ACID transactions.**
* **Data lakehouse fully exists in the cloud.**

**Explanation**

The answer is A data lakehouse stores unstructured data and is ACID-compliant,



## Question 7

Which of the following locations in Databricks product architecture hosts jobs/pipelines and queries?

* **Data plane**
* **Control plane**
* **Databricks Filesystem**
* **JDBC data source**
* **Databricks web application**

**Explanation**

The answer is Control Plane,

Databricks operates most of its services out of a control plane and a data plane, *please note serverless features like SQL Endpoint and DLT compute use shared compute in Control pane.*

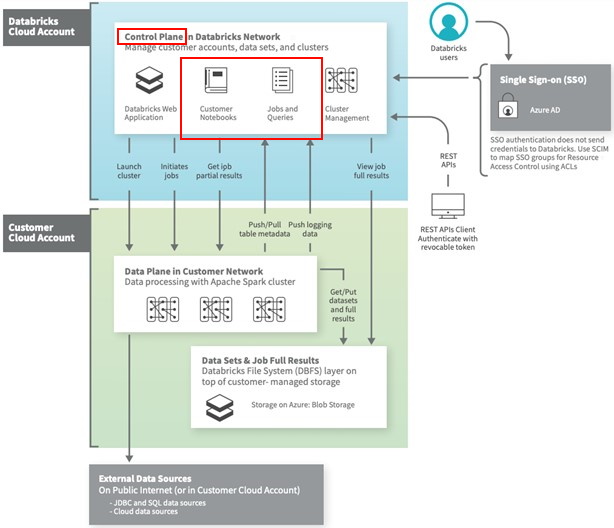
**Control Plane:**Stored in Databricks Cloud Account

* The control plane includes the backend services that Databricks manages in its own Azure account. Notebook commands and many other workspace configurations are stored in the control plane and encrypted at rest.

**Data Plane:**  Stored in Customer Cloud Account

* The data plane is managed by your Azure account and is where your data resides. This is also where data is processed. You can use Azure Databricks connectors so that your clusters can connect to external data sources outside of your Azure account to ingest data or for storage.

Here is the product architecture diagram highlighted where



## Question 8

You are currently working on a notebook that will populate a reporting table for downstream process consumption, this process needs to run on a schedule every hour. what type of cluster are you going to use to set up this job?

* **Since it’s just a single job and we need to run every hour, we can use an all-purpose cluster**
* **The job cluster is best suited for this purpose.**
* **Use Azure VM to read and write delta tables in Python**
* **Use delta live table pipeline to run in continuous mode**

**Explanation**

The answer is, The Job cluster is best suited for this purpose.

Since you don't need to interact with the notebook during the execution especially when it's a scheduled job, job cluster makes sense. Using an all-purpose cluster can be twice as expensive as a job cluster.

FYI,

When you run a job scheduler with option of creating a new cluster when the job is complete it terminates the cluster. You cannot restart a job cluster.

## Question 9

Which of the following developer operations in CI/CD flow can be implemented in Databricks Repos?

**Merge when code is committed**

**Pull request and review process**

**Trigger Databricks Repos API to pull the latest version of code into production folder**

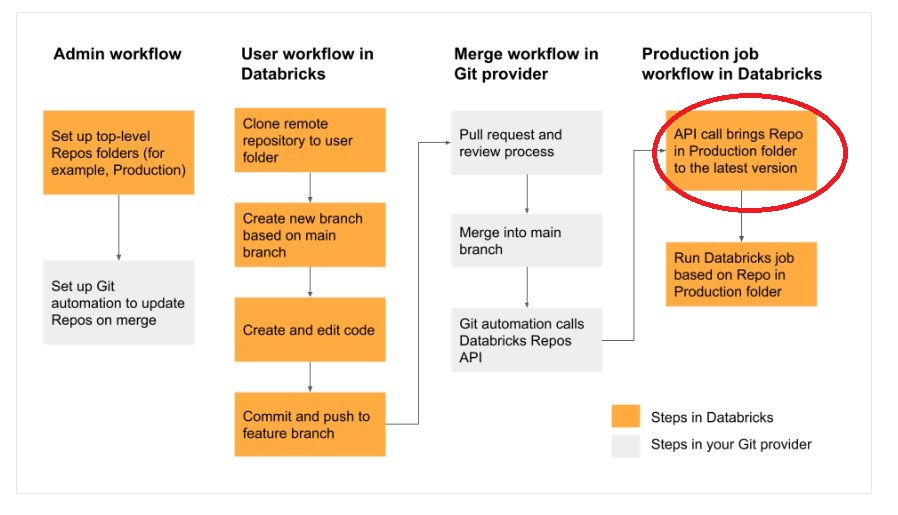
**Resolve merge conflicts**

**Delete a branch**

**Explanation**

See the below diagram to understand the role Databricks Repos and Git provider plays when building a CI/CD workflow.

All the steps highlighted in yellow can be done Databricks Repo, all the steps highlighted in Gray are done in a git provider like Github or Azure DevOps



## Question 10

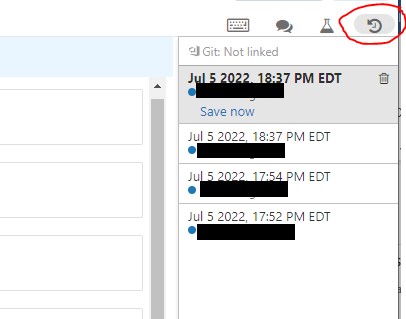
You are currently working with the second team and both teams are looking to modify the same notebook, you noticed that the second member is copying the notebooks to the personal folder to edit and replace the collaboration notebook, which notebook feature do you recommend to make the process easier to collaborate.

* **Databricks notebooks should be copied to a local machine and setup source control locally to version the notebooks**
* **Databricks notebooks support automatic change tracking and versioning**
* **Databricks Notebooks support real-time coauthoring on a single notebook**
* **Databricks notebooks can be exported into dbc archive files and stored in data lake**
* **Databricks notebook can be exported as HTML and imported at a later time**

**Explanation**

Answer is Databricks Notebooks support real-time coauthoring on a single notebook

Every change is saved, and a notebook can be changed my multiple users.



## Question 11

You are currently working on a project that requires the use of SQL and Python in a given notebook, what would be your approach

**Create two separate notebooks, one for SQL and the second for Python**

**A single notebook can support multiple languages, use the magic command to switch between the two.**

**Use an All-purpose cluster for python, SQL endpoint for SQL**

**Use job cluster to run python and SQL Endpoint for SQL**

**Explanation**

The answer is, A single notebook can support multiple languages, use the magic command to switch between the two.

Use %sql and %python magic commands within the same notebook.

## Question 12

Which of the following statements are correct on how Delta Lake implements a lake house?

* **Delta lake uses a proprietary format to write data, optimized for cloud storage**
* **Using Apache Hadoop on cloud object storage**
* **Delta lake always stores meta data in memory vs storage**
* **Delta lake uses open source, open format, optimized cloud storage and scalable meta data**
* **Delta lake stores data and meta data in computes memory**

**Explanation**

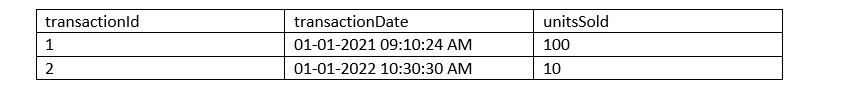
Delta lake is

* Open source
* Builds up on standard data format
* Optimized for cloud object storage
* Built for scalable metadata handling
* Delta lake is not
* Proprietary technology
* Storage format
* Storage medium

· Database service or data warehouse

## Question 13

You were asked to create or overwrite an existing delta table to store the below transaction data.

****

CREATE OR REPLACE DELTA TABLE transactions (

transactionId int,

transactionDate timestamp,

unitsSold int)

CREATE OR REPLACE TABLE IF EXISTS transactions (

transactionId int,

transactionDate timestamp,

unitsSold int)

FORMAT DELTA

CREATE IF EXSITS REPLACE TABLE transactions (

transactionId int,

transactionDate timestamp,

unitsSold int)

CREATE OR REPLACE TABLE transactions (

transactionId int,

transactionDate timestamp,

unitsSold int)

**(Correct)**

**Explanation**

The answer is

CREATE OR REPLACE TABLE transactions (

transactionId int,

transactionDate timestamp,

unitsSold int)

When creating a table in Databricks by default the table is stored in DELTA format.

## Question 14

if you run the command VACUUM transactions retain 0 hours? What is the outcome of this command?

* **Command will be successful, but no data is removed**
* **Command will fail if you have an active transaction running**
* **Command will fail, you cannot run the command with retentionDurationcheck enabled**
* **Command will be successful, but historical data will be removed**
* **Command runs successful and compacts all of the data in the table**

**Explanation**

The answer is,

Command will fail, you cannot run the command with retentionDurationcheck enabled.

1. VACUUM [ [db\_name.]table\_name | path] [RETAIN num HOURS] [DRY RUN]

* Recursively vacuum directories associated with the Delta table and remove data files that are no longer in the latest state of the transaction log for the table and are older than a retention threshold. Default is 7 Days.
* The reason this check is enabled is because, DELTA is trying to prevent unintentional deletion of history, and also one important thing to point out is with 0 hours of retention there is a possibility of data loss(see below kb)

Documentation in VACUUM https://docs.delta.io/latest/delta-utility.html

https://kb.databricks.com/delta/data-missing-vacuum-parallel-write.html

## Question 15

You noticed a colleague is manually copying the data to the backup folder prior to running an update command, incase if the update command did not provide the expected outcome so he can use the backup copy to replace table, which Delta Lake feature would you recommend simplifying the process?

* **Use time travel feature to refer old data instead of manually copying**
* **Use DEEP CLONE to clone the table prior to update to make a backup copy**
* **Use SHADOW copy of the table as preferred backup choice**
* **Cloud object storage retains previous version of the file**
* **Cloud object storage automatically backups the data**

**Explanation**

The answer is, Use time travel feature to refer old data instead of manually copying.

<https://databricks.com/blog/2019/02/04/introducing-delta-time-travel-for-large-scale-data-lakes.html>

1. SELECT count(\*) FROM my\_table TIMESTAMP AS OF "2019-01-01"
2. SELECT count(\*) FROM my\_table TIMESTAMP AS OF date\_sub(current\_date(), 1)
3. SELECT count(\*) FROM my\_table TIMESTAMP AS OF "2019-01-01 01:30:00.000"

## Question 16

Which one of the following is not a Databricks lakehouse object?

* **Tables**
* **Views**
* **Database/Schemas**
* **Catalog**
* **Functions**
* **Stored Procedures**

**Explanation**

The answer is, Stored Procedures.

Databricks lakehouse does not support stored procedures.

## Question 17

What type of table is created when you create delta table with below command?

CREATE TABLE transactions USING DELTA LOCATION "DBFS:/mnt/bronze/transactions"

* **Managed delta table**
* **External table**
* **Managed table**
* **Temp table**
* **Delta Lake table**

**Explanation**

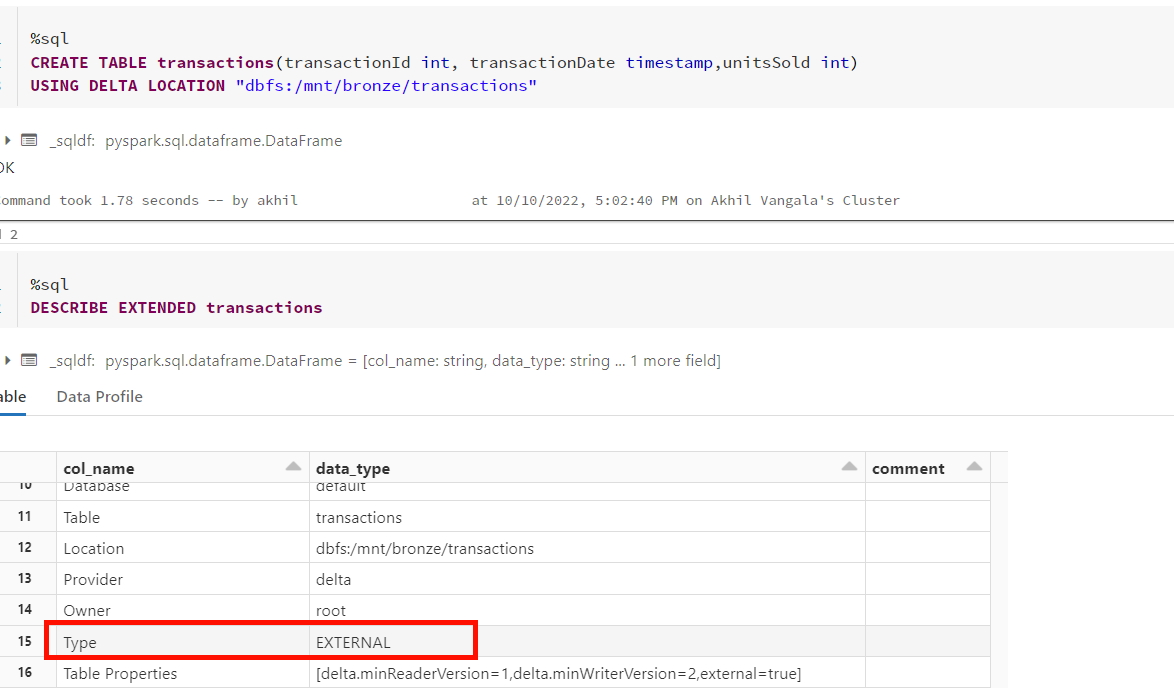
Anytime a table is created using the LOCATION keyword it is considered an external table, below is the current syntax.

Syntax

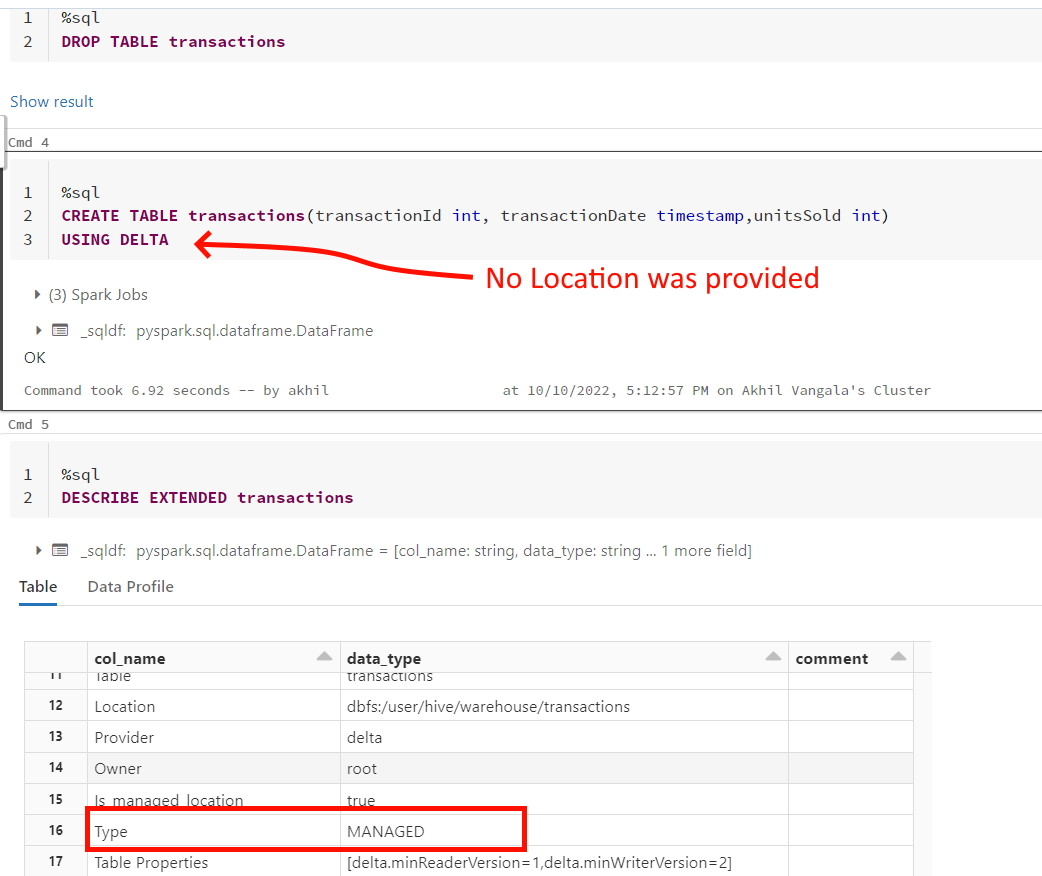
CREATE TABLE table\_name ( column column\_data\_type…) USING format LOCATION "dbfs:/"

format -> DELTA, JSON, CSV, PARQUET, TEXT

**I created the table command based on the above question, you can see it created an external table,**



**Let's remove the location keyword and run again, same syntax except for the LOCATION keyword is removed.**



## Question 18

Which of the following command can be used to drop a managed delta table and the underlying files in the storage?

* **DROP TABLE table\_name CASCADE**
* **DROP TABLE table\_name**
* **Use DROP TABLE table\_name command and manually delete files using command dbutils.fs.rm("/path",True)**
* **DROP TABLE table\_name INCLUDE\_FILES**
* **DROP TABLE table and run VACUUM command**

**Explanation**

The answer is DROP TABLE table\_name,

When a managed table is dropped, the table definition is dropped from metastore and everything including data, metadata, and history are also dropped from storage.

## Question 19

Which of the following is the correct statement for a session scoped temporary view?

* **Temporary views are lost once the notebook is detached and re-attached**
* **Temporary views stored in memory**
* **Temporary views can be still accessed even if the notebook is detached and attached**
* **Temporary views can be still accessed even if cluster is restarted**
* **Temporary views are created in local\_temp database**

**Explanation**

The answer is Temporary views are lost once the notebook is detached and attached

There are two types of temporary views that can be created, Session scoped and Global

* A local/session scoped temporary view is only available with a spark session, so another notebook in the same cluster can not access it. if a notebook is detached and reattached local temporary view is lost.
* A global temporary view is available to all the notebooks in the cluster, if a cluster restarts global temporary view is lost.

## Question 20

Which of the following is correct for the global temporary view?

* **global temporary views cannot be accessed once the notebook is detached and attached**
* **global temporary views can be accessed across many clusters**
* **global temporary views can be still accessed even if the notebook is detached and attached**
* **global temporary views can be still accessed even if the cluster is restarted**
* **global temporary views are created in a database called temp database**

**Explanation**

The answer is global temporary views can be still accessed even if the notebook is detached and attached

There are two types of temporary views that can be created Local and Global

· A local temporary view is only available with a spark session, so another notebook in the same cluster can not access it. if a notebook is detached and reattached local temporary view is lost.

· A global temporary view is available to all the notebooks in the cluster, even if the notebook is detached and reattached it can still be accessible but if a cluster is restarted the global temporary view is lost.

## Question 21

You are currently working on reloading customer\_sales tables using the below query

INSERT OVERWRITE customer\_sales

SELECT \* FROM customers c

INNER JOIN sales\_monthly s on s.customer\_id = c.customer\_id

After you ran the above command, the Marketing team quickly wanted to review the old data that was in the table. How does INSERT OVERWRITE impact the data in the customer\_sales table if you want to see the previous version of the data prior to running the above statement?

* **Overwrites the data in the table, all historical versions of the data, you can not time travel to previous versions**
* **Overwrites the data in the table but preserves all historical versions of the data, you can time travel to previous versions**
* **Overwrites the current version of the data but clears all historical versions of the data, so you can not time travel to previous versions.**
* **Appends the data to the current version, you can time travel to previous versions**
* **By default, overwrites the data and schema, you cannot perform time travel**

**Explanation**

The answer is, INSERT OVERWRITE  Overwrites the current version of the data but preserves all historical versions of the data, you can time travel to previous versions.

1. INSERT OVERWRITE customer\_sales
2. SELECT \* FROM customers c
3. INNER JOIN sales s on s.customer\_id = c.customer\_id

Let's just assume that this is the second time you are running the above statement, you can still query the prior version of the data using time travel, and any DML/DDL except DROP TABLE creates new PARQUET files so you can still access the previous versions of data.

SQL Syntax for Time travel

SELECT \* FROM table\_name as of [version number]

with customer\_sales example

SELECT \* FROM customer\_sales as of 1 -- previous version

SELECT \* FROM customer\_sales as of 2 -- current version

You see all historical changes on the table using DESCRIBE HISTORY table\_name

Note: the main difference between INSERT OVERWRITE and CREATE OR REPLACE TABLE(CRAS) is that  CRAS can modify the schema of the table, i.e it can add new columns or change data types of existing columns.  By default  INSERT OVERWRITE  only overwrites the data.

INSERT OVERWRITE  can also be used to update the schema when spark.databricks.delta.schema.autoMerge.enabled is set true if this option is not enabled and if there is a schema mismatch command  INSERT OVERWRITEwill fail.

Any DML/DDL operation(except DROP TABLE) on the Delta table preserves the historical version of the data.

## Question 22

Which of the following SQL statement can be used to query a table by eliminating duplicate rows from the query results?

* **SELECT DISTINCT \* FROM table\_name**
* **SELECT DISTINCT \* FROM table\_name HAVING COUNT(\*) > 1**
* **SELECT DISTINCT\_ROWS (\*) FROM table\_name**
* **SELECT \* FROM table\_name GROUP BY \* HAVING COUNT(\*) < 1**
* **SELECT \* FROM table\_name GROUP BY \* HAVING COUNT(\*) > 1**

**Explanation**

The  answer is  SELECT DISTINCT \* FROM table\_name

## Question 23

Which of the below SQL Statements can be used to create a SQL UDF to convert Celsius to Fahrenheit and vice versa, you need to pass two parameters to this function one, actual temperature, and the second that identifies if its needs to be converted to Fahrenheit or Celcius with a one-word letter F or C?

select udf\_convert(60,'C')  will result in 15.5

select udf\_convert(10,'F')  will result in 50

CREATE UDF FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURNS DOUBLE

RETURN CASE WHEN measure == 'F' then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

CREATE UDF FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURN CASE WHEN measure == 'F' then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

CREATE FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURN CASE WHEN measure == 'F' then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

CREATE FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURNS DOUBLE

RETURN CASE WHEN measure == 'F' then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

CREATE USER FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURNS DOUBLE

RETURN CASE WHEN measure == 'F' then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

**Explanation**

The answer is

CREATE FUNCTION udf\_convert(temp DOUBLE, measure STRING)

RETURNS DOUBLE

RETURN CASE WHEN measure == ‘F’ then (temp \* 9/5) + 32

ELSE (temp – 33 ) \* 5/9

END

## Question 24:

You are trying to calculate total sales made by all the employees by parsing a complex struct data type that stores employee and sales data, how would you approach this in SQL

Table definition,

batchId INT, performance ARRAY<STRUCT<employeeId: BIGINT, sales: INT>>, insertDate TIMESTAMP

Sample data of performance column

[

{ "employeeId":1234

"sales" : 10000},

{ "employeeId":3232

"sales" : 30000}

]

Calculate total sales made by all the employees?

Sample data with create table syntax for the data:

1. create or replace table sales as
2. select 1 as batchId ,
3. from\_json('[{ "employeeId":1234,"sales" : 10000 },{ "employeeId":3232,"sales" : 30000 }]',
4. 'ARRAY<STRUCT<employeeId: BIGINT, sales: INT>>') as performance,
5. current\_timestamp() as insertDate
6. union all
7. select 2 as batchId ,
8. from\_json('[{ "employeeId":1235,"sales" : 10500 },{ "employeeId":3233,"sales" : 32000 }]',
9. 'ARRAY<STRUCT<employeeId: BIGINT, sales: INT>>') as performance,
10. current\_timestamp() as insertDate

* 
  1. WITH CTE as (SELECT EXPLODE (performance) FROM table\_name)
  2. SELECT SUM (performance.sales) FROM CTE
* 
  1. WITH CTE as (SELECT FLATTEN (performance) FROM table\_name)
  2. SELECT SUM (sales) FROM CTE
* 
  1. select aggregate(flatten(collect\_list(performance.sales)), 0, (x, y) -> x + y)
  2. as total\_sales from sales

**(Correct)**

* 

**SELECT SUM(SLICE (performance, sales)) FROM employee**

* 
  1. select reduce(flatten(collect\_list(performance:sales)), 0, (x, y) -> x + y)
  2. as total\_sales from sales

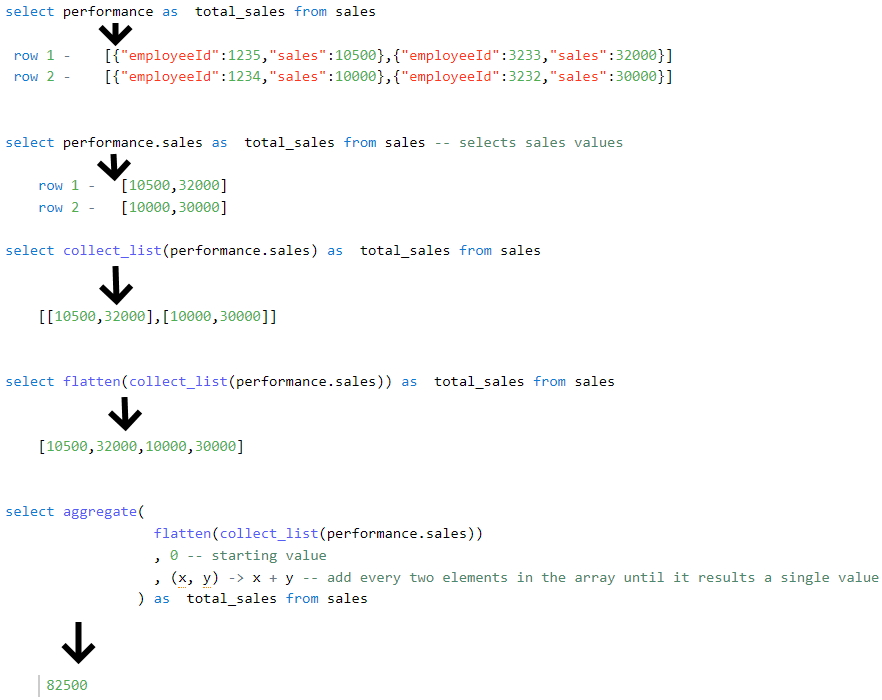
**Explanation**

The answer is

1. select aggregate(flatten(collect\_list(performance.sales)), 0, (x, y) -> x + y)
2. as total\_sales from sales

Nested Struct can be queried using the **.** notation performance.sales will give you access to all the sales values in the performance column.

Note: option D is wrong because it uses performance**:**sales not performance.sales. ":" this is only used when referring to JSON data but here we are dealing with a struct data type.  for the exam please make sure to understand if you are dealing with JSON data or Struct data.



Here are some additional examples

<https://docs.databricks.com/spark/latest/spark-sql/language-manual/functions/dotsign.html>

**Other solutions:**

we can also use reduce instead of aggregate

select reduce(flatten(collect\_list(performance.sales)), 0, (x, y) -> x + y) as  total\_sales from sales

we can also use explode and sum instead of using any higher-order funtions.

1. with cte as (
2. select
3. explode(flatten(collect\_list(performance.sales))) sales from sales
4. )
5. select
6. sum(sales) from cte

Sample data with create table syntax for the data:

1. create or replace table sales as
2. select 1 as batchId ,
3. from\_json('[{ "employeeId":1234,"sales" : 10000 },{ "employeeId":3232,"sales" : 30000 }]',
4. 'ARRAY<STRUCT<employeeId: BIGINT, sales: INT>>') as performance,
5. current\_timestamp() as insertDate
6. union all
7. select 2 as batchId ,
8. from\_json('[{ "employeeId":1235,"sales" : 10500 },{ "employeeId":3233,"sales" : 32000 }]',
9. 'ARRAY<STRUCT<employeeId: BIGINT, sales: INT>>') as performance,
10. current\_timestamp() as insertDate

## Question 25

Which of the following statements can be used to test the functionality of code to test number of rows in the table equal to 10 in python?

row\_count = spark.sql("select count(\*) from table").collect()[0][0]

* **assert (row\_count = 10, "Row count did not match")**
* **assert if (row\_count = 10, "Row count did not match")**
* **assert row\_count == 10, "Row count did not match"**
* **assert if row\_count == 10, "Row count did not match"**
* **assert row\_count = 10, "Row count did not match"**

**Explanation**

The answer is assert row\_count == 10, "Row count did not match"

Review below documentation

[Assert Python](https://www.w3schools.com/python/ref_keyword_assert.asp)

## Question 26

How do you handle failures gracefully when writing code in Pyspark,  fill in the blanks to complete the below statement

\_\_\_\_\_

Spark.read.table("table\_name").select("column").write.mode("append").SaveAsTable("new\_table\_name")

\_\_\_\_\_

print(query failed")

* **try: failure:**
* **try: catch:**
* **try: except:**
* **try: fail:**
* **try: error:**

**Explanation**

The answer is try: and except:

## Question 27

You are working on a process to query the table based on batch date, and batch date is an input parameter and expected to change every time the program runs, what is the best way to we can parameterize the query to run without manually changing the batch date?

* **Create a notebook parameter for batch date and assign the value to a python variable and use a spark data frame to filter the data based on the python variable**
* **Create a dynamic view that can calculate the batch date automatically and use the view to query the data**
* **There is no way we can combine python variable and spark code**
* **Manually edit code every time to change the batch date**
* **Store the batch date in the spark configuration and use a spark data frame to filter the data based on the spark configuration.**

**Explanation**

The answer is,  Create a notebook parameter for batch date and assign the value to a python variable and use a spark data frame to filter the data based on the python variable

## Question 28

Which of the following commands results in the successful creation of a view on top of the delta stream(stream on delta table)?

* **Spark.read.format("delta").table("sales").createOrReplaceTempView("streaming\_vw")**
* **Spark.readStream.format("delta").table("sales").createOrReplaceTempView("streaming\_vw")**
* **Spark.read.format("delta").table("sales").mode("stream").createOrReplaceTempView("streaming\_vw")**
* **Spark.read.format("delta").table("sales").trigger("stream").createOrReplaceTempView("streaming\_vw")**
* **Spark.read.format("delta").stream("sales").createOrReplaceTempView("streaming\_vw")**
* **You can not create a view on streaming data source.**

**Explanation**

The answer is

Spark.readStream.table("sales").createOrReplaceTempView("streaming\_vw")

When you load a Delta table as a stream source and use it in a streaming query, the query processes all of the data present in the table as well as any new data that arrives after the stream is started.

You can load both paths and tables as a stream, you also have the ability to ignore deletes and changes(updates, Merge, overwrites) on the delta table.

Here is more information,

<https://docs.databricks.com/delta/delta-streaming.html#delta-table-as-a-source>

## Question 29

Which of the following techniques structured streaming uses to create an end-to-end fault tolerance?

* **Checkpointing and Water marking**
* **Write ahead logging and water marking**
* **Checkpointing and idempotent sinks**
* **Write ahead logging and idempotent sinks**
* **Stream will failover to available nodes in the cluster**

**Explanation**

The answer is Checkpointing and idempotent sinks

How does structured streaming achieves end to end fault tolerance:

* First, Structured Streaming uses checkpointing and write-ahead logs to record the offset range of data being processed during each trigger interval.
* Next, the streaming sinks are designed to be \_idempotent\_—that is, multiple writes of the same data (as identified by the offset) do *not* result in duplicates being written to the sink.

Taken together, replayable data sources and idempotent sinks allow Structured Streaming to ensure **end-to-end, exactly-once semantics** under any failure condition.

## Question 30

Which of the following two options are supported in identifying the arrival of new files, and incremental data from Cloud object storage using Auto Loader?

* **Directory listing, File notification**
* **Checking pointing, watermarking**
* **Writing ahead logging, read head logging**
* **File hashing, Dynamic file lookup**
* **Checkpointing and Write ahead logging**

**Explanation**

The answer is A, Directory listing, File notifications

**Directory listing**: Auto Loader identifies new files by listing the input directory.

**File notification**: Auto Loader can automatically set up a notification service and queue service that subscribe to file events from the input directory.

[Choosing between file notification and directory listing modes | Databricks on AWS](https://docs.databricks.com/ingestion/auto-loader/file-detection-modes.html)

## Question 31

Which of the following data workloads will utilize a Bronze table as its destination?

* **A job that aggregates cleaned data to create standard summary statistics**
* **A job that queries aggregated data to publish key insights into a dashboard**
* **A job that ingests raw data from a streaming source into the Lakehouse**
* **A job that develops a feature set for a machine learning application**
* **A job that enriches data by parsing its timestamps into a human-readable format**

**Explanation**

The answer is A job that ingests raw data from a streaming source into the Lakehouse.

The ingested data from the raw streaming data source like Kafka is first stored in the Bronze layer as first destination before it is further optimized and stored in Silver.

[Medallion Architecture – Databricks](https://databricks.com/glossary/medallion-architecture)

Bronze Layer:

1. Raw copy of ingested data

2. Replaces traditional data lake

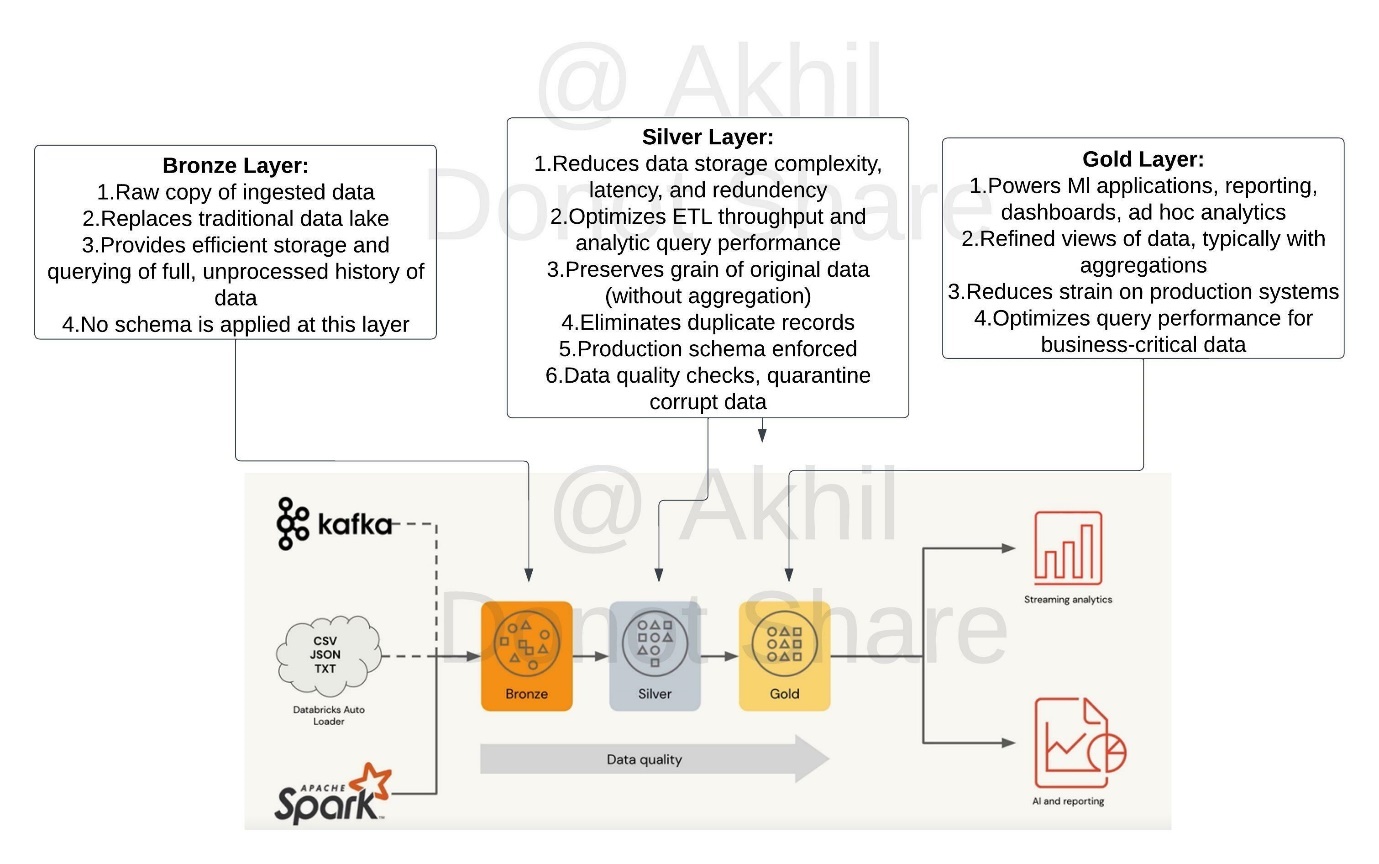
3. Provides efficient storage and querying of full, unprocessed history of data

4. No schema is applied at this layer

***Exam focus: Please review the below image and understand the role of each layer(bronze, silver, gold) in medallion architecture, you will see varying questions targeting each layer and its purpose.***

*Sorry I had to add the watermark some people in Udemy are copying my content.*

Purpose of each layer in medallion architecture



## Question 32

Which of the following data workloads will utilize a silver table as its source?

* **A job that enriches data by parsing its timestamps into a human-readable format**
* **A job that queries aggregated data that already feeds into a dashboard**
* **A job that ingests raw data from a streaming source into the Lakehouse**
* **A job that aggregates cleaned data to create standard summary statistics**
* **A job that cleans data by removing malformatted records**

**Explanation**

The answer is,  A job that aggregates cleaned data to create standard summary statistics

Silver zone maintains the grain of the original data, in this scenario a job is taking data from the silver zone as the **source**and aggregating and storing them in the gold zone.

[Medallion Architecture – Databricks](https://databricks.com/glossary/medallion-architecture)

Silver Layer:

1. Reduces data storage complexity, latency, and redundency

2. Optimizes ETL throughput and analytic query performance

3. Preserves grain of original data (without aggregation)

4. Eliminates duplicate records

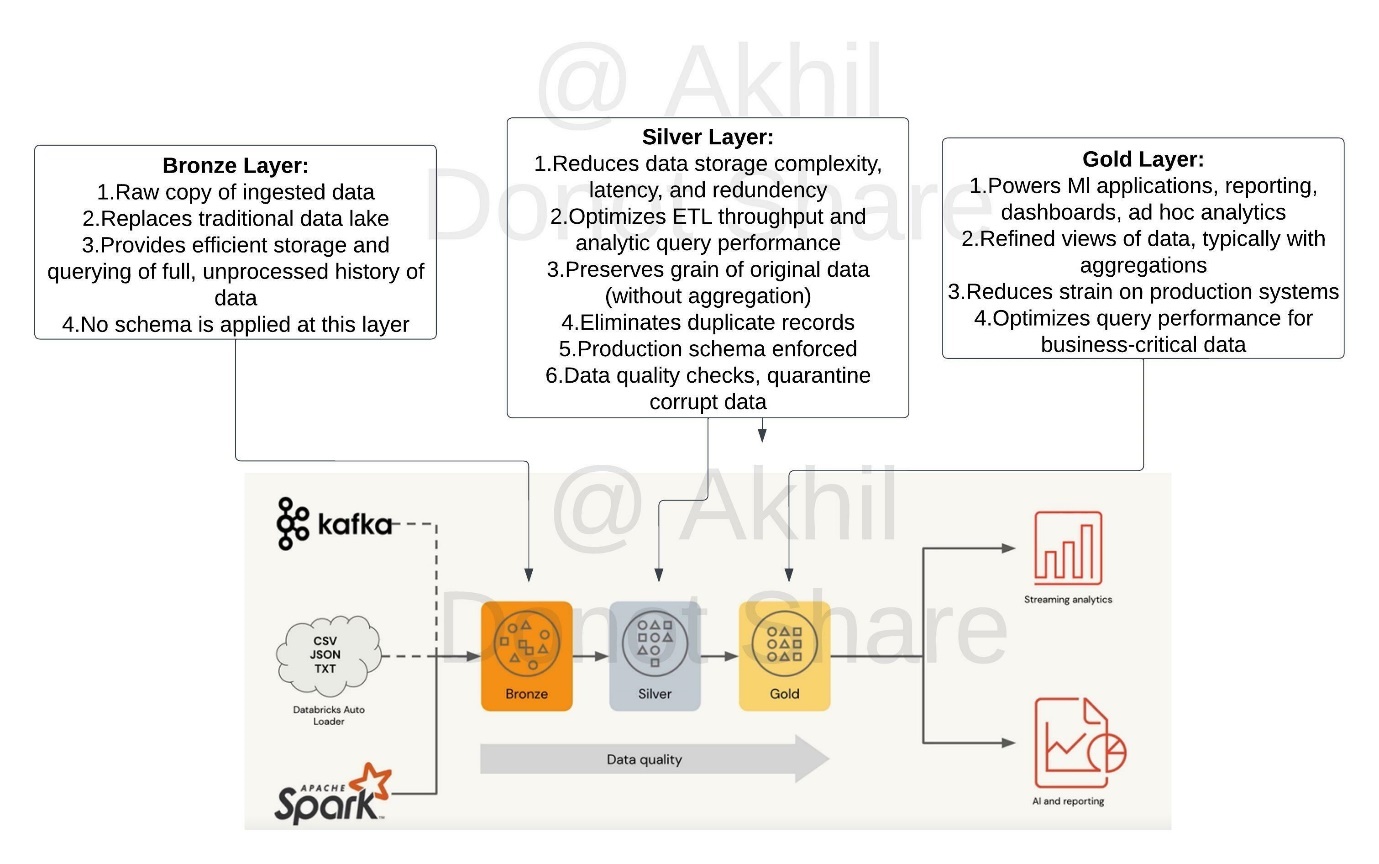
5. production schema enforced

6. Data quality checks, quarantine corrupt data

***Exam focus: Please review the below image and understand the role of each layer(bronze, silver, gold) in medallion architecture, you will see varying questions targeting each layer and its purpose.***

*Sorry I had to add the watermark some people in Udemy are copying my content.*

Purpose of each layer in medallion architecture



## Question 33

Which of the following data workloads will utilize a gold table as its source?

* **A job that enriches data by parsing its timestamps into a human-readable format**
* **A job that queries aggregated data that already feeds into a dashboard**
* **A job that ingests raw data from a streaming source into the Lakehouse**
* **A job that aggregates cleaned data to create standard summary statistics**
* **A job that cleans data by removing malformatted records**

**Explanation**

The answer is,  A job that queries aggregated data that already feeds into a dashboard

The gold layer is used to store aggregated data, which are typically used for dashboards and reporting.

Review the below link for more info,

[Medallion Architecture – Databricks](https://databricks.com/glossary/medallion-architecture)

Gold Layer:

1. Powers Ml applications, reporting, dashboards, ad hoc analytics

2. Refined views of data, typically with aggregations

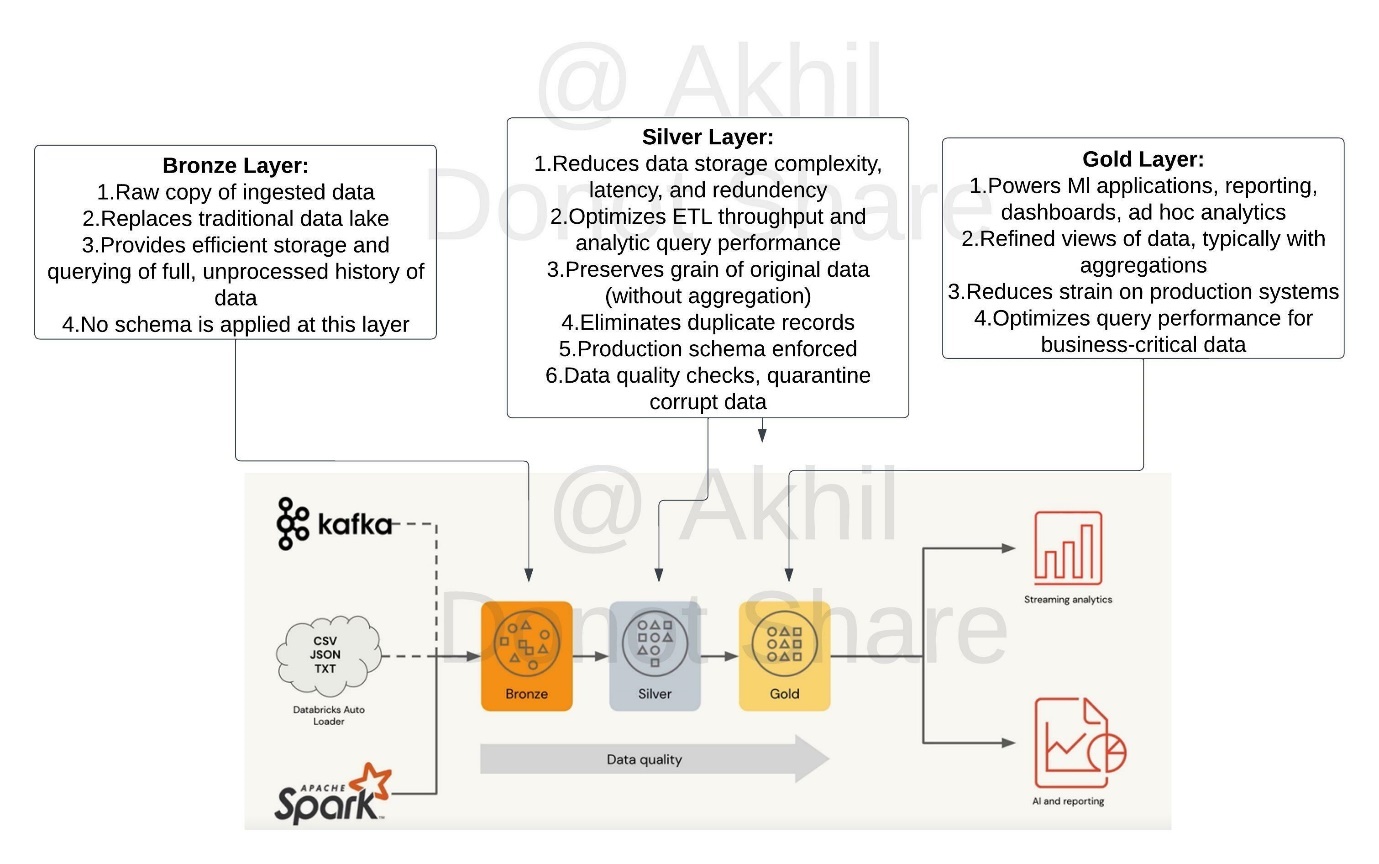
3. Reduces strain on production systems

4. Optimizes query performance for business-critical data

***Exam focus: Please review the below image and understand the role of each layer(bronze, silver, gold) in medallion architecture, you will see varying questions targeting each layer and its purpose.***

*Sorry I had to add the watermark some people in Udemy are copying my content.*

Purpose of each layer in medallion architecture



## Question 34

You are currently asked to work on building a data pipeline, you have noticed that you are currently working with a data source that has a lot of data quality issues and you need to monitor data quality and enforce it as part of the data ingestion process, which of the following tools can be used to address this problem?

* **AUTO LOADER**
* **DELTA LIVE TABLES**
* **JOBS and TASKS**
* **UNITY Catalog and Data Governance**
* **STRUCTURED STREAMING with MULTI HOP**

**Explanation**

The answer is, DELTA LIVE TABLES

Delta live tables expectations can be used to identify and quarantine bad data, all of the data quality metrics are stored in the event logs which can be used to later analyze and monitor.

[DELTA LIVE Tables expectations](https://docs.microsoft.com/en-us/azure/databricks/data-engineering/delta-live-tables/delta-live-tables-expectations)

Below are three types of expectations, make sure to pay attention differences between these three.

**Retain invalid records:**

Use the expect operator when you want to keep records that violate the expectation. Records that violate the expectation are added to the target dataset along with valid records:

Python

1. @dlt.expect("valid timestamp", "col(“timestamp”) > '2012-01-01'")

SQL

1. CONSTRAINT valid\_timestamp EXPECT (timestamp > '2012-01-01')

**Drop invalid records:**

Use the expect or drop operator to prevent the processing of invalid records. Records that violate the expectation are dropped from the target dataset:

Python

1. @dlt.expect\_or\_drop("valid\_current\_page", "current\_page\_id IS NOT NULL AND current\_page\_title IS NOT NULL")

SQL

1. CONSTRAINT valid\_current\_page EXPECT (current\_page\_id IS NOT NULL and current\_page\_title IS NOT NULL) ON VIOLATION DROP ROW

**Fail on invalid records:**

When invalid records are unacceptable, use the expect or fail operator to halt execution immediately when a record fails validation. If the operation is a table update, the system atomically rolls back the transaction:

Python

1. @dlt.expect\_or\_fail("valid\_count", "count > 0")

SQL

1. CONSTRAINT valid\_count EXPECT (count > 0) ON VIOLATION FAIL UPDATE

## Question 35

When building a DLT s pipeline you have two options to create a live tables, what is the main difference between CREATE STREAMING LIVE TABLE vs CREATE LIVE TABLE?

* **CREATE STREAMING LIVE table is used in MULTI HOP Architecture**
* **CREATE LIVE TABLE is used when working with Streaming data sources and Incremental data**
* **CREATE STREAMING LIVE TABLE is used when working with Streaming data sources and Incremental data**
* **There is no difference both are the same, CREATE STRAMING LIVE will be deprecated soon**
* **CREATE LIVE TABLE is used in DELTA LIVE TABLES, CREATE STREAMING LIVE can only used in Structured Streaming applications**

**Explanation**

The answer is, CREATE STREAMING LIVE TABLE is used when working with Streaming data sources and Incremental data

Bottom of Form

## Question 36

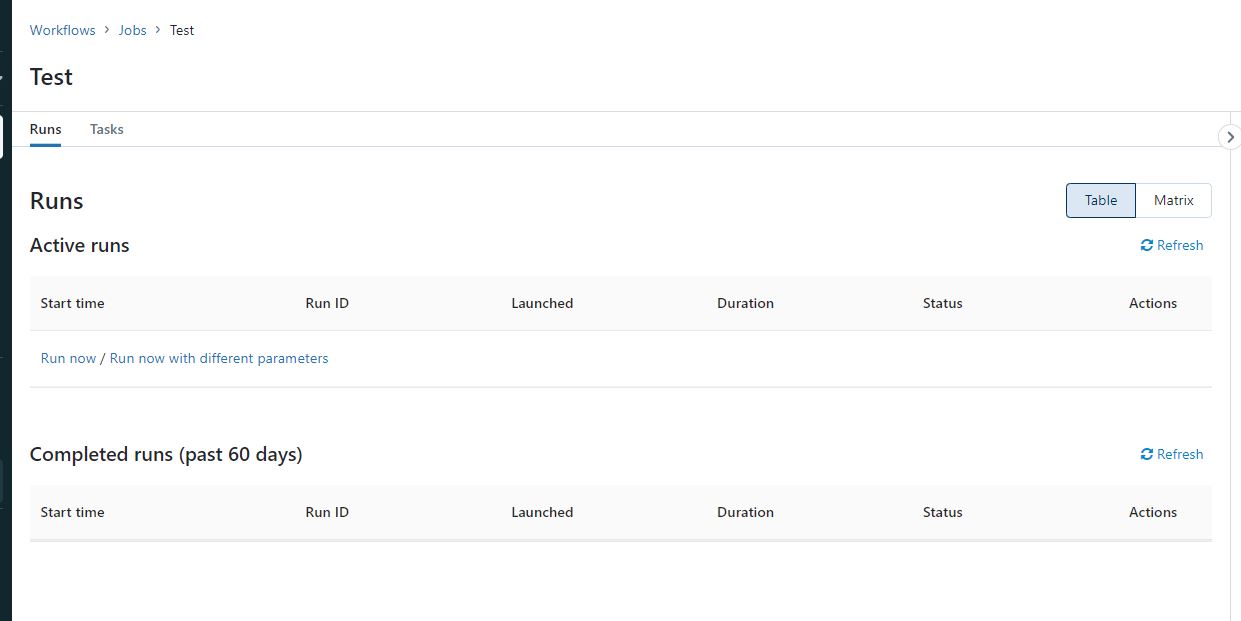
A particular job seems to be performing slower and slower over time, the team thinks this started to happen when a recent production change was implemented, you were asked to take look at the job history and see if we can identify trends and root cause, where in the workspace UI can you perform this analysis?

* **Under jobs UI select the job you are interested, under runs we can see current active runs and last 60 days historical run**
* **Under jobs UI select the job cluster, under spark UI select the application job logs, then you can access last 60 day historical runs**
* **Under Workspace logs, select job logs and select the job you want to monitor to view the last 60 day historical runs**
* **Under Compute UI, select Job cluster and select the job cluster to see last 60 day historical runs**
* **Historical job runs can only be accessed by REST API**

**Explanation**

The answer is,

Under jobs UI select the job you are interested, under runs we can see current active runs and last 60 days historical run



## Question 37

What are the different ways you can schedule a job in Databricks workspace?

* **Continuous, Incremental**
* **On-Demand runs, File notification from Cloud object storage**
* **Cron, On Demand runs**
* **Cron, File notification from Cloud object storage**
* **Once, Continuous**

**Explanation**

The answer is, Cron, On-Demand runs

Supports running job immediately or using can be scheduled using CRON syntax

[Jobs in Databricks](https://docs.databricks.com/data-engineering/jobs/jobs.html#run-a-job)

## Question 38

You have noticed that Databricks SQL queries are running slow, you are asked to look reason why queries are running slow and identify steps to improve the performance, when you looked at the issue you noticed all the queries are running in parallel and using a SQL endpoint(SQL Warehouse) with a single cluster. Which of the following steps can be taken to improve the performance/response times of the queries?

\*Please note Databricks recently renamed SQL endpoint to SQL warehouse.

* + **They can turn on the Serverless feature for the SQL endpoint(SQL warehouse).**
  + **They can increase the maximum bound of the SQL endpoint(SQL warehouse)’s scaling range**
  + **They can increase the warehouse size from 2X-Smal to 4XLarge of the SQL endpoint(SQL warehouse).**
  + **They can turn on the Auto Stop feature for the SQL endpoint(SQL warehouse).**
  + **They can turn on the Serverless feature for the SQL endpoint(SQL warehouse) and change the Spot Instance Policy to “Reliability Optimized.”**

**Explanation**

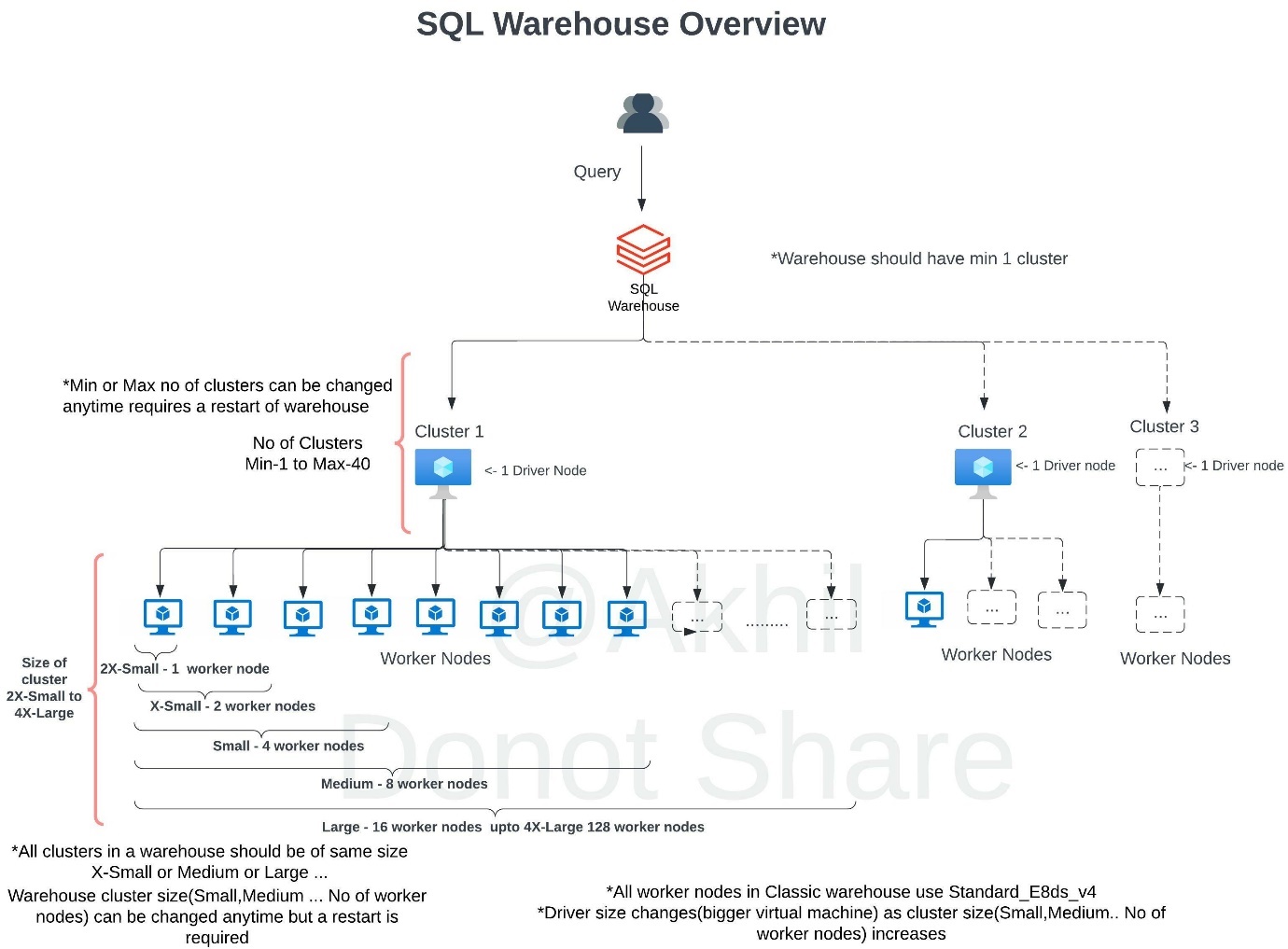
The answer is, They can increase the maximum bound of the SQL endpoint’s scaling range when you increase the max scaling range more clusters are added so queries instead of waiting in the queue can start running using available clusters, see below for more explanation.

**The question is looking to test your ability to know how to scale a SQL Endpoint(SQL Warehouse) and you have to look for cue words or need to understand if the queries are running sequentially or concurrently. if the queries are running sequentially then scale up(Size of the cluster from 2X-Small to 4X-Large) if the queries are running concurrently or with more users then scale out(add more clusters).**

**SQL Endpoint(SQL Warehouse) Overview: (Please read all of the below points and the below diagram to understand )**

1. A SQL Warehouse should have at least one cluster
2. A cluster comprises one driver node and one or many worker nodes
3. No of worker nodes in a cluster is determined by the size of the cluster (2X -Small ->1 worker, X-Small ->2 workers.... up to 4X-Large -> 128 workers) this is called **Scale up**
4. A single cluster irrespective of cluster size(2X-Smal.. to ...4XLarge) can only run 10 queries at any given time if a user submits 20 queries all at once to a warehouse with 3X-Large cluster size and cluster scaling (min 1, max1) while 10 queries will start running the remaining 10 queries wait in a queue for these 10 to finish.
5. Increasing the Warehouse cluster size can improve the performance of a query, for example, if a query runs for 1 minute in a 2X-Small warehouse size it may run in 30 Seconds if we change the warehouse size to X-Small. this is due to 2X-Small having 1 worker node and X-Small having 2 worker nodes so the query has more tasks and runs faster (note: this is an ideal case example, the scalability of a query performance depends on many factors, it can not always be linear)
6. A warehouse can have more than one cluster this is called **Scale out**. If a warehouse is configured with X-Small cluster size with cluster scaling(Min1, Max 2) Databricks spins up an additional cluster if it detects queries are waiting in the queue, If a warehouse is configured to run 2 clusters(Min1, Max 2), and let's say a user submits 20 queries, 10 queriers will start running and holds the remaining in the queue and databricks will automatically start the second cluster and starts redirecting the 10 queries waiting in the queue to the second cluster.
7. A single query will not span more than one cluster, once a query is submitted to a cluster it will remain in that cluster until the query execution finishes irrespective of how many clusters are available to scale.

Please review the below diagram to understand the above concepts:

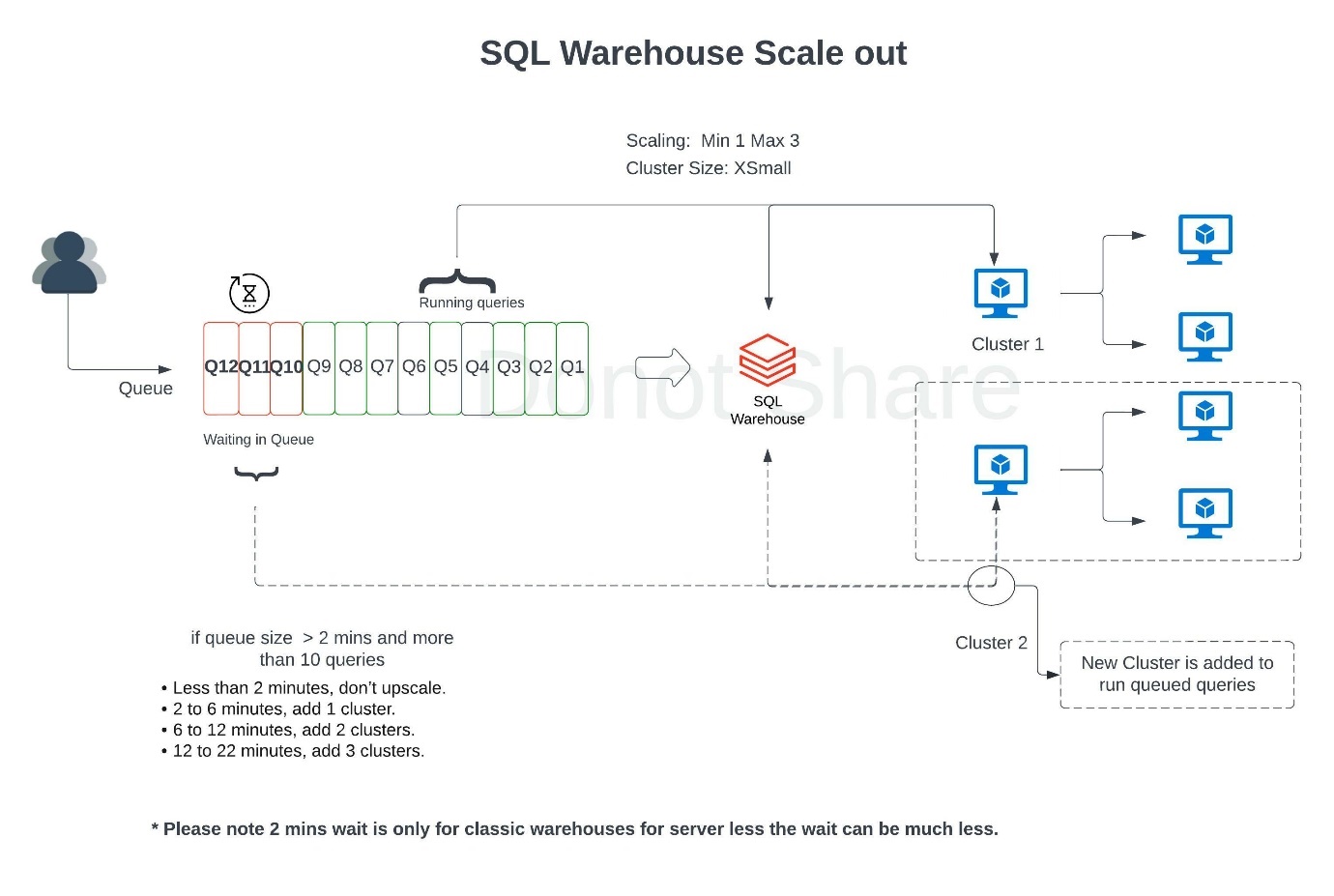


SQL endpoint(SQL Warehouse) scales horizontally(scale-out) and vertical (scale-up), you have to understand when to use what.

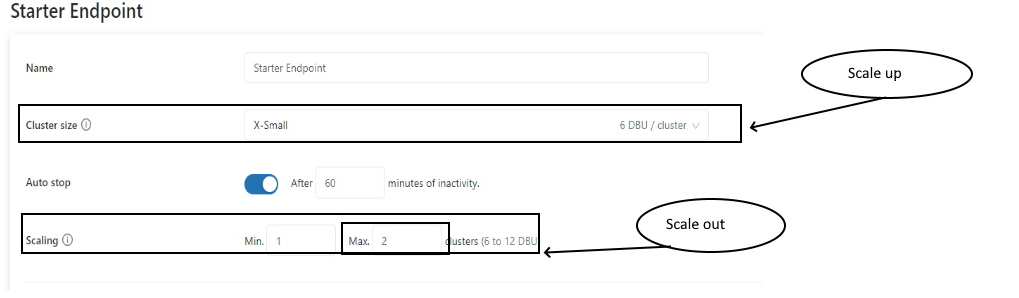
**Scale-out** -> to add more clusters for a SQL endpoint, change max number of clusters

If you are trying to improve the throughput, being able to run as many queries as possible then having an additional cluster(s) will improve the performance.

Databricks SQL automatically scales as soon as it detects queries are in queuing state, in this example scaling is set for min 1 and max 3 which means the warehouse can add three clusters if it detects queries are waiting.

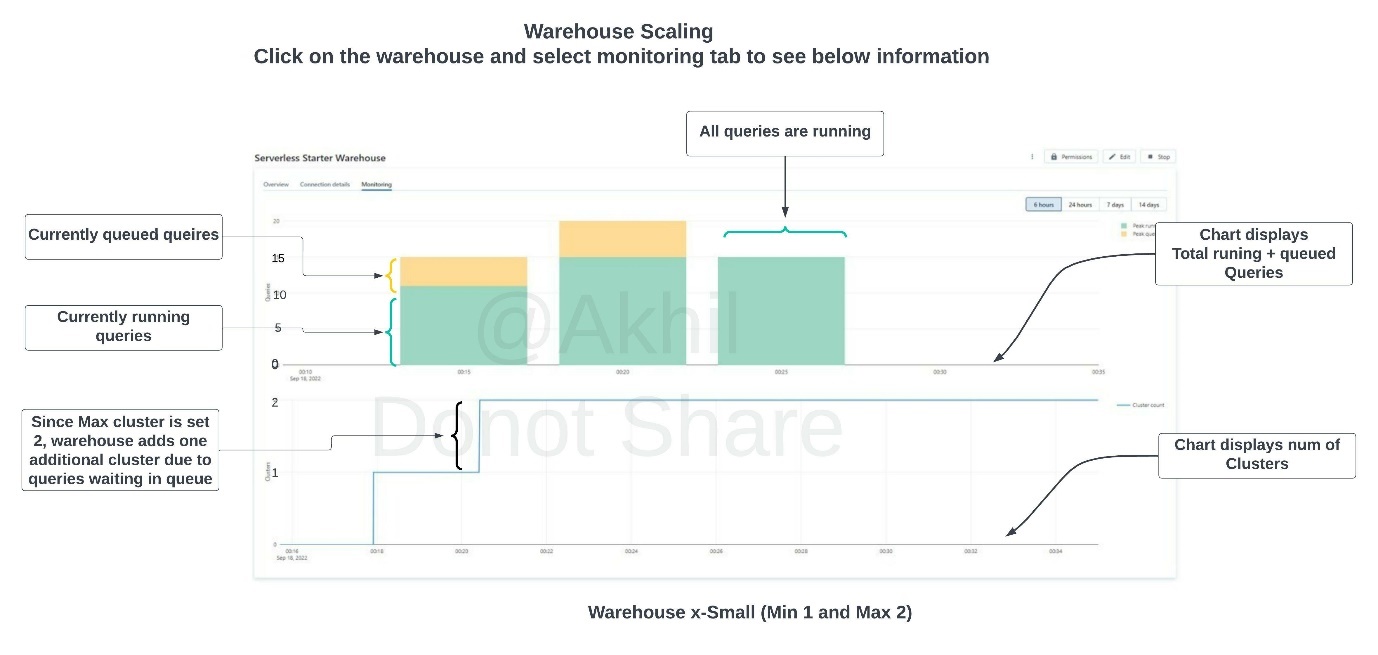


During the warehouse creation or after you have the ability to change the warehouse size (2X-Small....to ...4XLarge) to improve query performance and the maximize scaling range to add more clusters on a SQL Endpoint(SQL Warehouse) scale-out, if you are changing an existing warehouse you may have to restart the warehouse to make the changes effective.



**How do you know how many clusters you need(How to set Max cluster size)?**

When you click on an existing warehouse and select the monitoring tab, you can see warehouse utilization information(see below), there are two graphs that provide important information on how the warehouse is being utilized, if you see queries are being queued that means your warehouse can benefit from additional clusters. Please review the additional DBU cost associated with adding clusters so you can take a well balanced decision between cost and performance.



## Question 39

You currently working with the marketing team to setup a dashboard for ad campaign analysis, since the team is not sure how often the dashboard should be refreshed they have decided to do a manual refresh on an as needed basis. Which of the following steps can be taken to reduce the overall cost of the compute when the team is not using the compute?

\*Please note that Databricks recently change the name of SQL Endpoint to SQL Warehouses.

* **They can turn on the Serverless feature for the SQL endpoint(SQL Warehouse).**
* **They can decrease the maximum bound of the SQL endpoint(SQL Warehouse) scaling range.**
* **They can decrease the cluster size of the SQL endpoint(SQL Warehouse).**
* **They can turn on the Auto Stop feature for the SQL endpoint(SQL Warehouse).**
* **They can turn on the Serverless feature for the SQL endpoint(SQL Warehouse) and change the Spot Instance Policy from “Reliability Optimized” to  “Cost optimized”**

**Explanation**

The answer is, They can turn on the Auto Stop feature for the SQL endpoint(SQL Warehouse).

Use auto stop to automatically terminate the cluster when you are not using it.

## Question 40:

You had worked with the Data analysts team to set up a SQL Endpoint(SQL warehouse) point so they can easily query and analyze data in the gold layer, but once they started consuming the SQL Endpoint(SQL warehouse)  you noticed that during the peak hours as the number of users increase you are seeing queries taking longer to finish, which of the following steps can be taken to resolve the issue?

\*Please note Databricks recently renamed SQL endpoint to SQL warehouse.

* **They can turn on the Serverless feature for the SQL endpoint(SQL warehouse).**
* **They can increase the maximum bound of the SQL endpoint(SQL warehouse) ’s scaling range.**
* **They can increase the cluster size from 2X-Small to 4X-Large of the SQL endpoint(SQL warehouse) .**
* **They can turn on the Auto Stop feature for the SQL endpoint(SQL warehouse) .**
* **They can turn on the Serverless feature for the SQL endpoint(SQL warehouse)  and change the Spot Instance Policy from “Cost optimized” to “Reliability Optimized.”**

**Explanation**

the answer is,

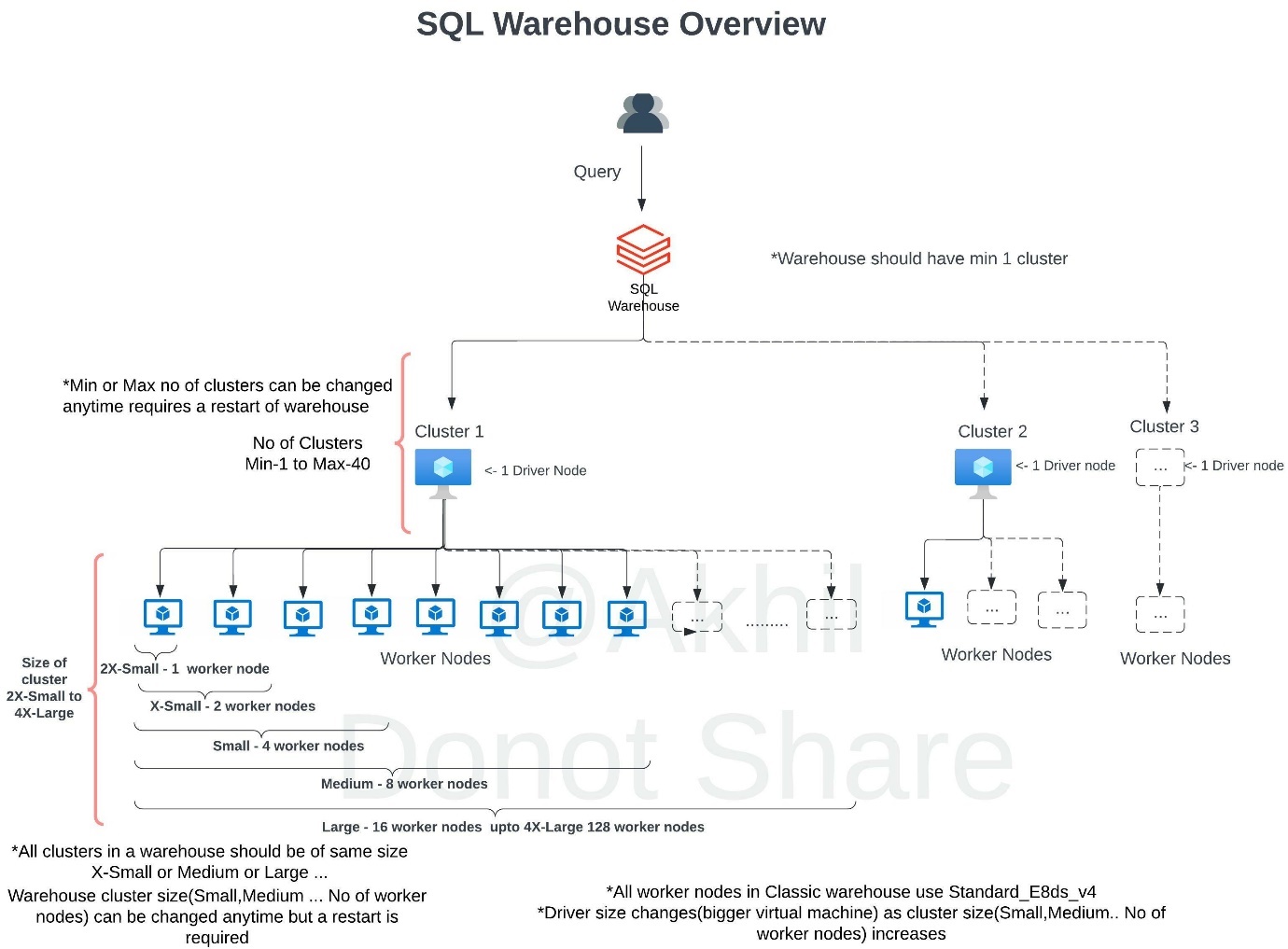
They can increase the maximum bound of the SQL endpoint’s scaling range, when you increase the maximum bound you can add more clusters to the warehouse which can then run additional queries that are waiting in the queue to run, focus on the below explanation that talks about Scale-out.

**The question is looking to test your ability to know how to scale a SQL Endpoint(SQL Warehouse) and you have to look for cue words or need to understand if the queries are running sequentially or concurrently. if the queries are running sequentially then scale up(Size of the cluster from 2X-Small to 4X-Large) if the queries are running concurrently or with more users then scale out(add more clusters).**

**SQL Endpoint(SQL Warehouse) Overview: (Please read all of the below points and the below diagram to understand )**

1. A SQL Warehouse should have at least one cluster
2. A cluster comprises one driver node and one or many worker nodes
3. No of worker nodes in a cluster is determined by the size of the cluster (2X -Small ->1 worker, X-Small ->2 workers.... up to 4X-Large -> 128 workers) this is called **Scale up**
4. A single cluster irrespective of cluster size(2X-Smal.. to ...4XLarge) can only run 10 queries at any given time if a user submits 20 queries all at once to a warehouse with 3X-Large cluster size and cluster scaling (min 1, max1) while 10 queries will start running the remaining 10 queries wait in a queue for these 10 to finish.
5. Increasing the Warehouse cluster size can improve the performance of a query, example if a query runs for 1 minute in a 2X-Small warehouse size, it may run in 30 Seconds if we change the warehouse size to X-Small. this is due to 2X-Small has 1 worker node and X-Small has 2 worker nodes so the query has more tasks and runs faster (note: this is an ideal case example, the scalability of a query performance depends on many factors, it can not always be linear)
6. A warehouse can have more than one cluster this is called **Scale out**. If a warehouse is configured with X-Small cluster size with cluster scaling(Min1, Max 2) Databricks spins up an additional cluster if it detects queries are waiting in the queue, If a warehouse is configured to run 2 clusters(Min1, Max 2), and let's say a user submits 20 queries, 10 queriers will start running and holds the remaining in the queue and databricks will automatically start the second cluster and starts redirecting the 10 queries waiting in the queue to the second cluster.
7. A single query will not span more than one cluster, once a query is submitted to a cluster it will remain in that cluster until the query execution finishes irrespective of how many clusters are available to scale.

Please review the below diagram to understand the above concepts:

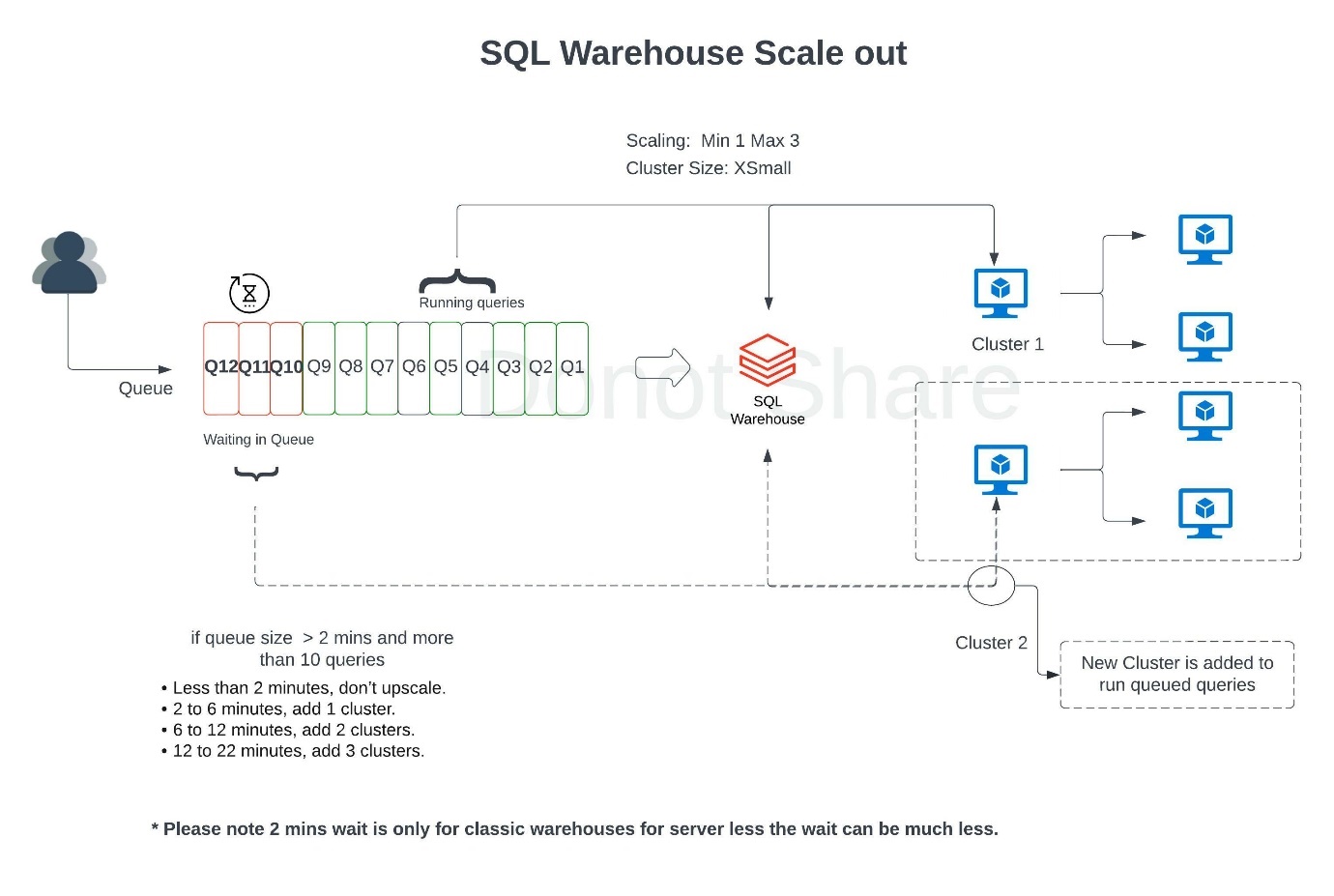


SQL endpoint(SQL Warehouse) scales horizontally(scale-out) and vertical (scale-up), you have to understand when to use what.

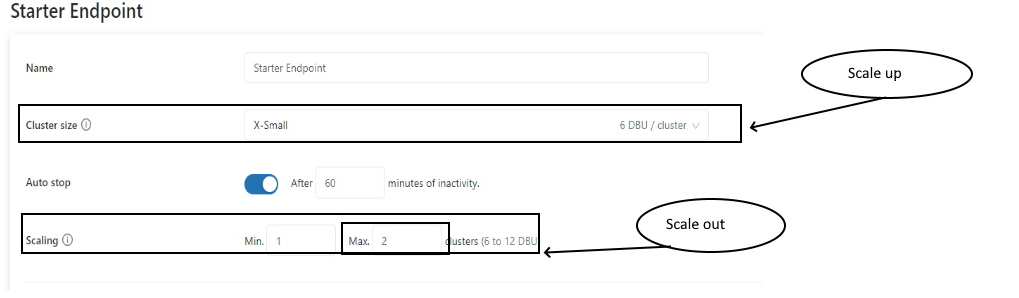
**Scale-out** -> to add more clusters for a SQL endpoint, change max number of clusters

If you are trying to improve the throughput, being able to run as many queries as possible then having an additional cluster(s) will improve the performance.

Databricks SQL automatically scales as soon as it detects queries are in queuing state, in this example scaling is set for min 1 and max 3 which means the warehouse can add three clusters if it detects queries are waiting.

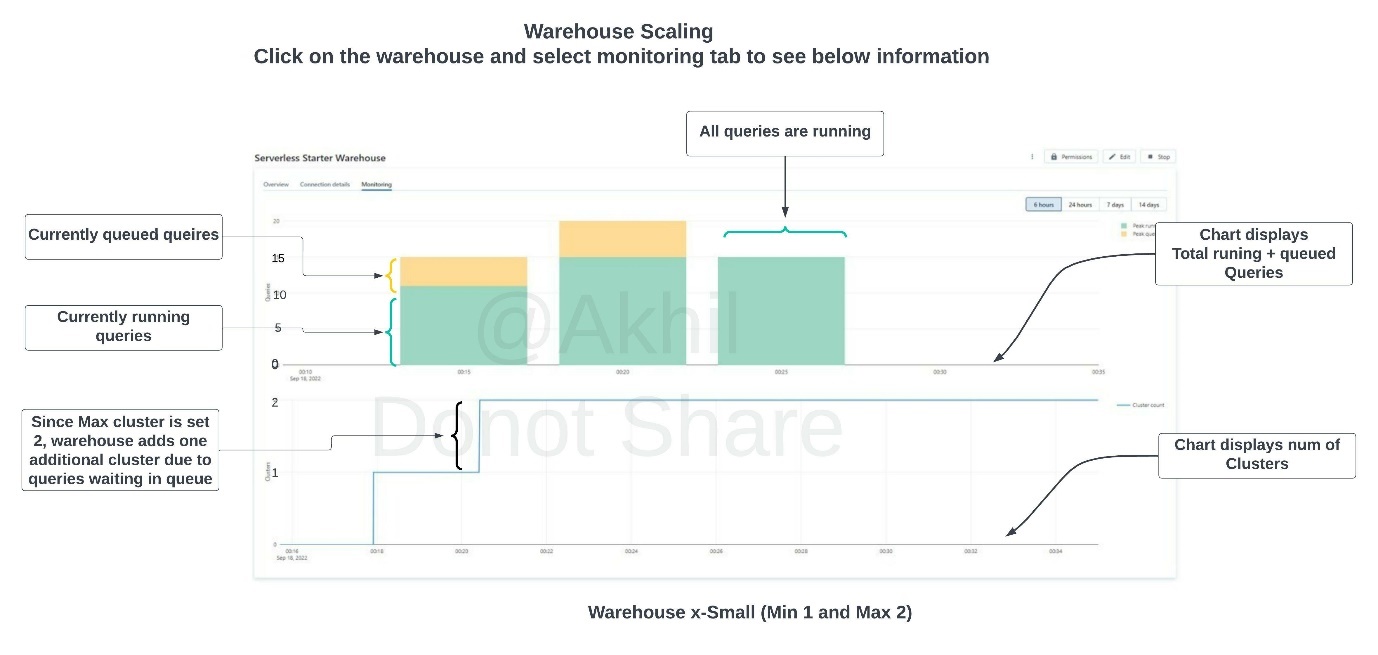


During the warehouse creation or after you have the ability to change the warehouse size (2X-Small....to ...4XLarge) to improve query performance and the maximize scaling range to add more clusters on a SQL Endpoint(SQL Warehouse) scale-out, if you are changing an existing warehouse you may have to restart the warehouse to make the changes effective.



**How do you know how many clusters you need(How to set Max cluster size)?**

When you click on an existing warehouse and select the monitoring tab, you can see warehouse utilization information(see below), there are two graphs that provide important information on how the warehouse is being utilized, if you see queries are being queued that means your warehouse can benefit from additional clusters. Please review the additional DBU cost associated with adding clusters so you can take a well balanced decision between cost and performance.



## Question 41

The research team has put together a funnel analysis query to monitor the customer traffic on the e-commerce platform, the query takes about 30 mins to run on a small SQL endpoint cluster with max scaling set to 1 cluster. What steps can be taken to improve the performance of the query?

* **They can turn on the Serverless feature for the SQL endpoint.**
* **They can increase the maximum bound of the SQL endpoint’s scaling range anywhere from between 1 to 100 to review the performance and select the size that meets the required SLA.**
* **They can increase the cluster size anywhere from X small to 3XL to review the performance and select the size that meets the required SLA.**
* **They can turn off the Auto Stop feature for the SQL endpoint to more than 30 mins.**
* **They can turn on the Serverless feature for the SQL endpoint and change the Spot Instance Policy from “Cost optimized” to “Reliability Optimized.”**

**Explanation**

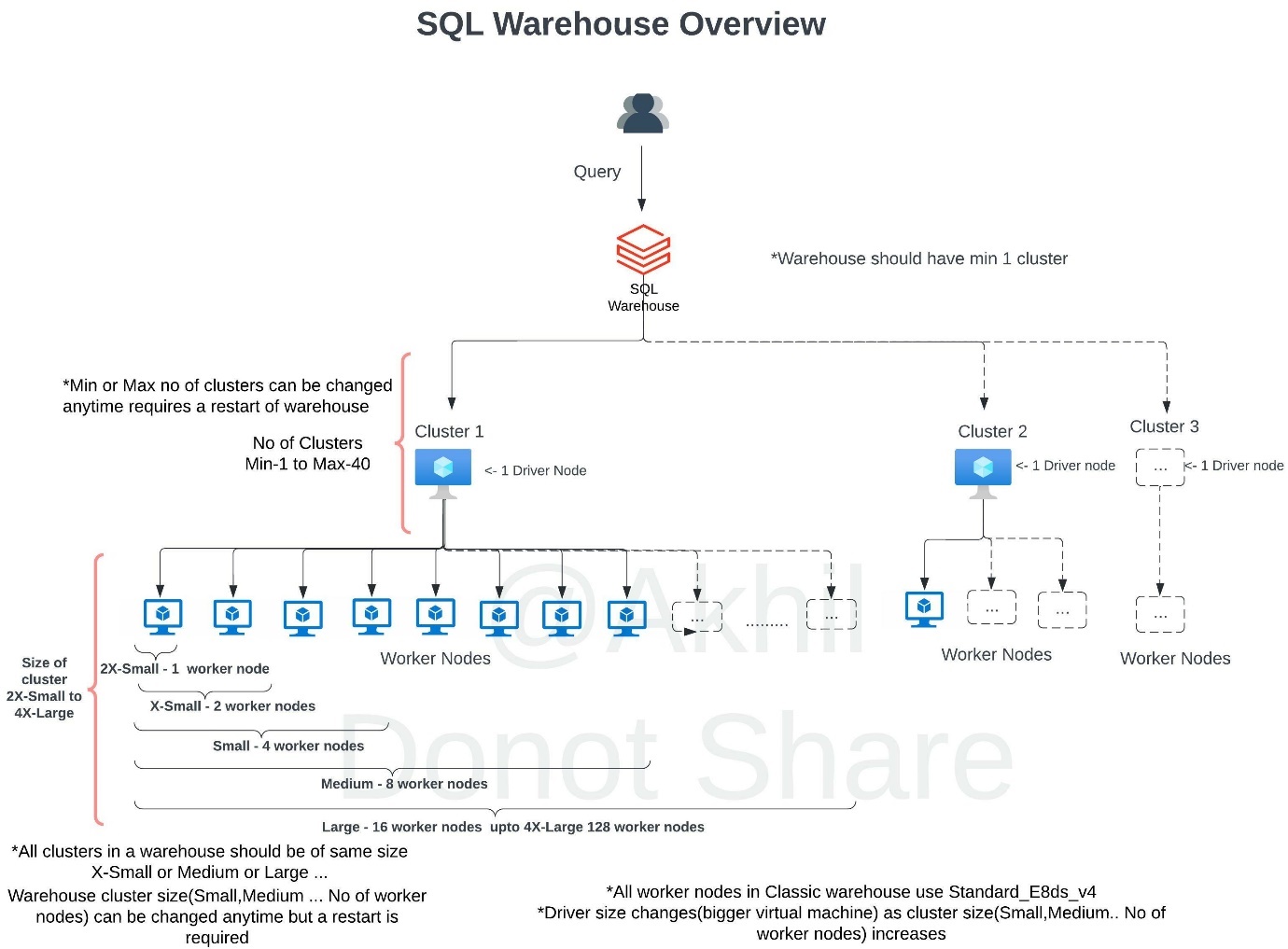
The answer is, They can increase the cluster size anywhere from 2X-Small to 4XL(Scale Up) to review the performance and select the size that meets your SLA. If you are trying to improve the performance of a single query at a time having additional memory, additional worker nodes mean that more tasks can run in a cluster which will improve the performance of that query.

**The question is looking to test your ability to know how to scale a SQL Endpoint(SQL Warehouse) and you have to look for cue words or need to understand if the queries are running sequentially or concurrently. if the queries are running sequentially then scale up(Size of the cluster from 2X-Small to 4X-Large) if the queries are running concurrently or with more users then scale out(add more clusters).**

**SQL Endpoint(SQL Warehouse) Overview: (Please read all of the below points and the below diagram to understand )**

1. A SQL Warehouse should have at least one cluster
2. A cluster comprises one driver node and one or many worker nodes
3. No of worker nodes in a cluster is determined by the size of the cluster (2X -Small ->1 worker, X-Small ->2 workers.... up to 4X-Large -> 128 workers) this is called **Scale Up**
4. A single cluster irrespective of cluster size(2X-Smal.. to ...4XLarge) can only run 10 queries at any given time if a user submits 20 queries all at once to a warehouse with 3X-Large cluster size and cluster scaling (min 1, max1) while 10 queries will start running the remaining 10 queries wait in a queue for these 10 to finish.
5. Increasing the Warehouse cluster size can improve the performance of a query, example if a query runs for 1 minute in a 2X-Small warehouse size, it may run in 30 Seconds if we change the warehouse size to X-Small. this is due to 2X-Small has 1 worker node and X-Small has 2 worker nodes so the query has more tasks and runs faster (note: this is an ideal case example, the scalability of a query performance depends on many factors, it can not always be linear)
6. A warehouse can have more than one cluster this is called **Scale Out**. If a warehouse is configured with X-Small cluster size with cluster scaling(Min1, Max 2) Databricks spins up an additional cluster if it detects queries are waiting in the queue, If a warehouse is configured to run 2 clusters(Min1, Max 2), and let's say a user submits 20 queries, 10 queriers will start running and holds the remaining in the queue and databricks will automatically start the second cluster and starts redirecting the 10 queries waiting in the queue to the second cluster.
7. A single query will not span more than one cluster, once a query is submitted to a cluster it will remain in that cluster until the query execution finishes irrespective of how many clusters are available to scale.

Please review the below diagram to understand the above concepts:



**Scale-up**-> Increase the size of the SQL endpoint, change cluster size from 2X-Small to up to 4X-Large

If you are trying to improve the performance of a single query having additional memory, additional worker nodes and cores will result in more tasks running in the cluster will ultimately improve the performance.

During the warehouse creation or after, you have the ability to change the warehouse size (2X-Small....to ...4XLarge) to improve query performance and the maximize scaling range to add more clusters on a SQL Endpoint(SQL Warehouse) scale-out if you are changing an existing warehouse you may have to restart the warehouse to make the changes effective.



## Question 42

Unity catalog simplifies managing multiple workspaces, by storing and managing permissions and ACL at \_\_\_\_\_\_\_ level

* **Workspace**
* **Account**
* **Storage**
* **Data pane**
* **Control pane**

**Explanation**

The answer is, Account Level

The classic access control list (tables, workspace, cluster) is at the workspace level, Unity catalog is at the account level and can manage all the workspaces in an Account.

## Question 43

Which of the following section in the UI can be used to manage permissions and grants to tables?

* **User Settings**
* **Admin UI**
* **Workspace admin settings**
* **User access control lists**
* **Data Explorer**

**Explanation**

The answer is Data Explorer

[Data explorer](https://docs.databricks.com/sql/user/data/index.html)

## Question 44

Which of the following is not a privilege in the Unity catalog?

**SELECT**

**MODIFY**

**DELETE**

**CREATE TABLE**

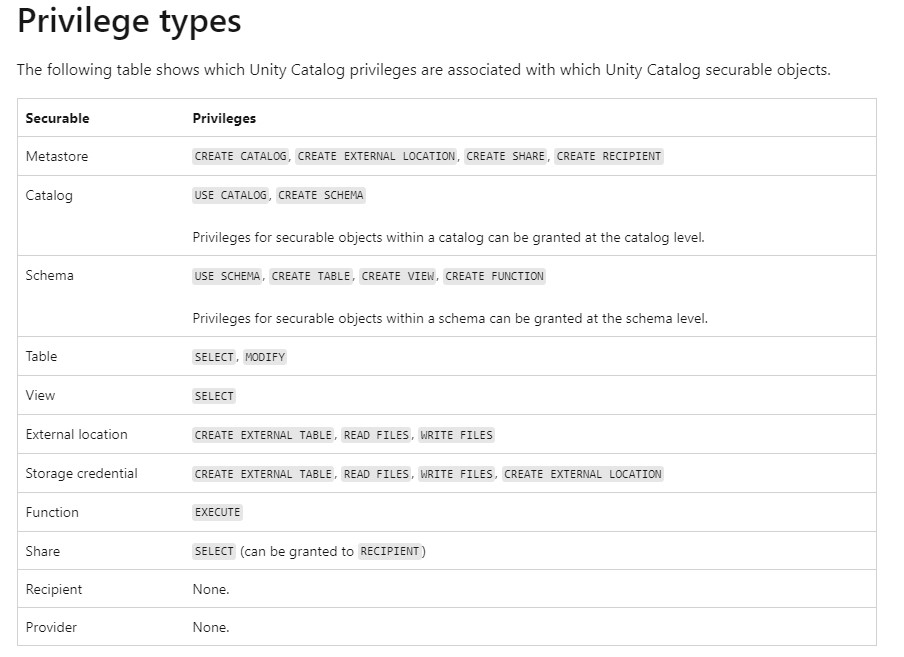
**EXECUTE**

**Explanation**

The Answer is DELETE and UPDATE permissions do not exit, you have to use MODIFY which provides both Update and Delete permissions.

**Please note: TABLE ACL privilege types are different from Unity Catalog privilege types, please read the question carefully.**

Here is the list of all privileges in **Unity Catalog:**



Unity Catalog Privileges

<https://learn.microsoft.com/en-us/azure/databricks/spark/latest/spark-sql/language-manual/sql-ref-privileges#privilege-types>

Table ACL privileges

<https://learn.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges#privileges>

## Question 45

A team member is leaving the team and he/she is currently the owner of the few tables, instead of transfering the ownership to a user you have decided to transfer the ownership to a group so in the future anyone in the group can manage the permissions rather than a single individual, which of the following commands help you accomplish this?

* **ALTER TABLE table\_name OWNER to 'group'**
* **TRANSFER OWNER table\_name to 'group'**
* **GRANT OWNER table\_name to 'group'**
* **ALTER OWNER ON table\_name to 'group'**
* **GRANT OWNER On table\_name to 'group'**

**Explanation**

The answer is ALTER TABLE table\_name OWNER to ‘group’

[Assign owner to object](https://docs.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges#assign-owner-to-object)

Bottom of Form