Contents

[Question 1 3](#_Toc155611774)

[Question 2 4](#_Toc155611775)

[Question 3 6](#_Toc155611776)

[Question 4 7](#_Toc155611777)

[Question 5 8](#_Toc155611778)

[Question 6 9](#_Toc155611779)

[Question 7 10](#_Toc155611780)

[Question 8 11](#_Toc155611781)

[Question 9 12](#_Toc155611782)

[Question 10 13](#_Toc155611783)

[Question 11 14](#_Toc155611784)

[Question 12 15](#_Toc155611785)

[Question 13: 16](#_Toc155611786)

[Question 14 17](#_Toc155611787)

[Question 15 18](#_Toc155611788)

[Question 16 19](#_Toc155611789)

[Question 17 20](#_Toc155611790)

[Question 18 21](#_Toc155611791)

[Question 19 22](#_Toc155611792)

[Question 20 23](#_Toc155611793)

[Question 21 24](#_Toc155611794)

[Question 22 25](#_Toc155611795)

[Question 23 26](#_Toc155611796)

[Question 24 28](#_Toc155611797)

[Question 25 29](#_Toc155611798)

[Question 26 30](#_Toc155611799)

[Question 27 32](#_Toc155611800)

[Question 28 34](#_Toc155611801)

[Question 29: 36](#_Toc155611802)

[Question 30 38](#_Toc155611803)

[Question 31 40](#_Toc155611804)

[Question 32 41](#_Toc155611805)

[Question 33 43](#_Toc155611806)

[Question 34 46](#_Toc155611807)

[Question 35 48](#_Toc155611808)

[Question 36 50](#_Toc155611809)

[Question 37 53](#_Toc155611810)

[Question 38 58](#_Toc155611811)

[Question 39 59](#_Toc155611812)

[Question 40 60](#_Toc155611813)

[Question 41 61](#_Toc155611814)

[Question 42 62](#_Toc155611815)

[Question 43 64](#_Toc155611816)

[Question 44 65](#_Toc155611817)

[Question 45 68](#_Toc155611818)

## Question 1

Which of the statements is correct when choosing between lakehouse and Datawarehouse?

* **Traditional Data warehouses have special indexes which are optimized for Machine learning**
* **Traditional Data warehouses can serve low query latency with high reliability for BI workloads**
* **SQL support is only available for Traditional Datawarehouse’s, Lakehouses support Python and Scala**
* **Traditional Data warehouses are the preferred choice if we need to**

**support ACID, Lakehouse does not support ACID.**

* **Lakehouse replaces the current dependency on data lakes and data warehouses uses an open standard storage format and supports low latency BI workloads.**

**Explanation**

The lakehouse replaces the current dependency on data lakes and data warehouses for modern data companies that desire:

· Open, direct access to data stored in standard data formats.

· Indexing protocols optimized for machine learning and data science.

· Low query latency and high reliability for BI and advanced analytics.

## Question 2

Where are Interactive notebook results stored in Databricks product architecture?

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

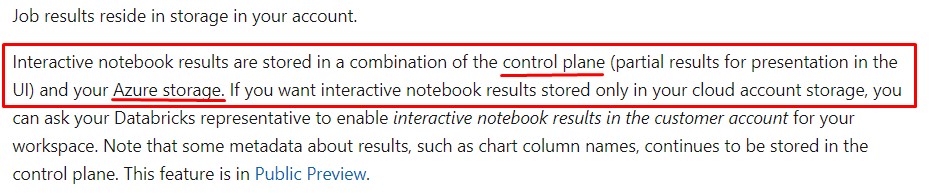
**Explanation**

The answer is Data and Control plane,

Only Job results are stored in Data Plane(your storage), Interactive notebook results are stored in a combination of the control plane (partial results for presentation in the UI) and customer storage.

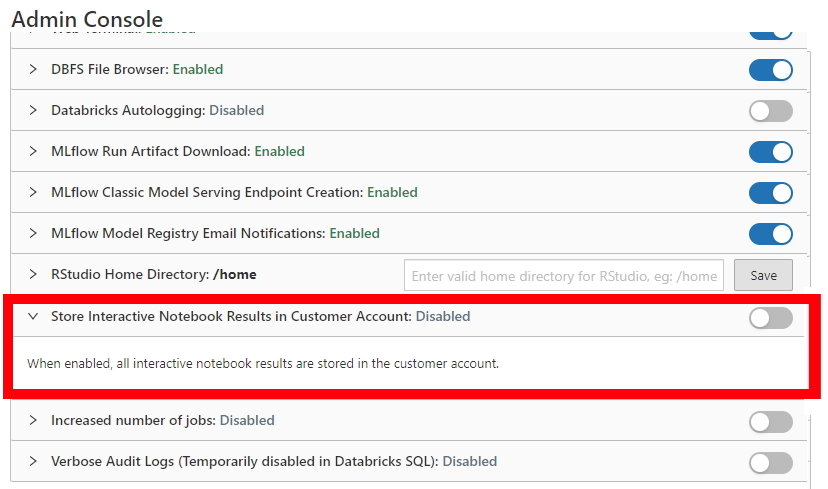
<https://docs.microsoft.com/en-us/azure/databricks/getting-started/overview#--high-level-architecture>

Snippet from the above documentation,



**How to change this behavior?**

You can change this behavior using Workspace/Admin Console settings for that workspace, once enabled all of the interactive results are stored in the customer account(data plane) except the new notebook visualization feature Databricks has recently introduced, this still stores some metadata in the control pane irrespective of the below settings. please refer to the documentation for more details.



**Why is this important to know?**

I recently worked on a project where we had to deal with sensitive information of customers and we had a security requirement that all of the data need to be stored in the data plane including notebook results.

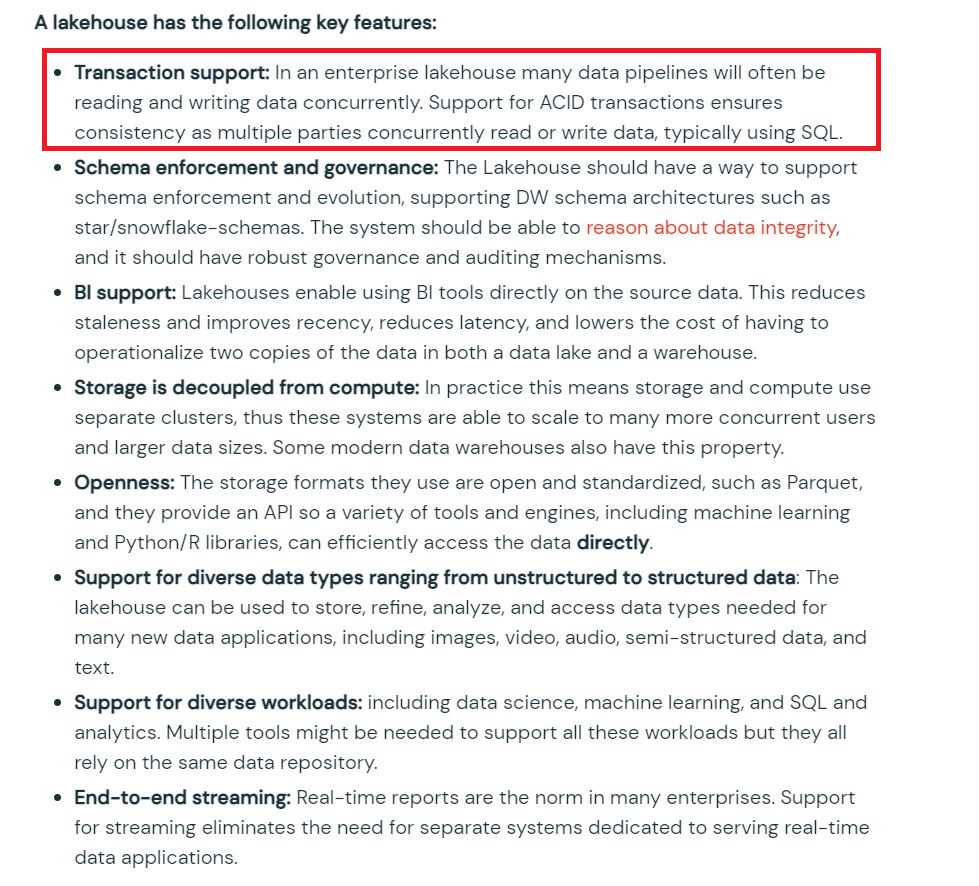
## Question 3

Which of the following statements are true about a lakehouse?

* **Lakehouse only supports Machine learning workloads and Data warehouses support BI workloads**
* **Lakehouse only supports end-to-end streaming workloads and Data warehouses support Batch workloads**
* **Lakehouse does not support ACID**
* **Lakehouse do not support SQL**
* **Lakehouse supports Transactions**

**Explanation**

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



## Question 4

Which of the following SQL command can be used to insert or update or delete rows based on a condition to check if a row(s) exists?

* **MERGE INTO table\_name**
* **COPY INTO table\_name**
* **UPDATE table\_name**
* **INSERT INTO OVERWRITE table\_name**
* **INSERT IF EXISTS table\_name**

**Explanation**

here is the additional documentation for your review.

<https://docs.databricks.com/spark/latest/spark-sql/language-manual/delta-merge-into.html>

MERGE INTO target\_table\_name [target\_alias]

USING source\_table\_reference [source\_alias]

ON merge\_condition

[ WHEN MATCHED [ AND condition ] THEN matched\_action ] [...]

[ WHEN NOT MATCHED [ AND condition ] THEN not\_matched\_action ] [...]

matched\_action

{ DELETE |

UPDATE SET \* |

UPDATE SET { column1 = value1 } [, ...] }

not\_matched\_action

{ INSERT \* |

INSERT (column1 [, ...] ) VALUES (value1 [, ...])

Bottom of Form

## Question 5

When investigating a data issue you realized that a process accidentally updated the table,  you want to query the same table with yesterday's version of the data so you can review what the prior version looks like, what is the best way to query historical data so you can do your analysis?

* **SELECT \* FROM TIME\_TRAVEL(table\_name) WHERE time\_stamp = 'timestamp'**
* **TIME\_TRAVEL FROM table\_name WHERE time\_stamp = date\_sub(current\_date(), 1)**
* **SELECT \* FROM table\_name TIMESTAMP AS OF date\_sub(current\_date(), 1)**
* **DISCRIBE HISTORY table\_name AS OF date\_sub(current\_date(), 1)**
* **SHOW HISTORY table\_name AS OF date\_sub(current\_date(), 1)**

**Explanation**

The answer is   SELECT \* FROM table\_name TIMESTAMP as of date\_sub(current\_date(), 1)

FYI, Time travel supports two ways one is using timestamp and the second way is using version number,

**Timestamp:**

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"

**Version Number:**

1. SELECT count(\*) FROM my\_table VERSION AS OF 5238
2. SELECT count(\*) FROM my\_table@v5238
3. SELECT count(\*) FROM delta.`/path/to/my/table@v5238`

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

## Question 6

While investigating a data issue, you wanted to review yesterday's version of the table using below command, while querying the previous version of the table using time travel you realized that you are no longer able to view the historical data in the table and you could see it the table was updated yesterday based on the table history(DESCRIBE HISTORY table\_name) command what could be the reason why you can not access this data?

SELECT \* FROM table\_name TIMESTAMP AS OF date\_sub(current\_date(), 1)

* **You currently do not have access to view historical data**
* **By default, historical data is cleaned every 180 days in DELTA**
* **A command VACUUM table\_name RETAIN 0 was ran on the table**
* **Time travel is disabled**
* **Time travel must be enabled before you query previous data**

**Explanation**

The answer is, VACUUM table\_name RETAIN 0 was ran

The VACUUM command recursively vacuums directories associated with the Delta table and removes data files that are no longer in the latest state of the transaction log for the table and are older than a retention threshold. The default is 7 Days.

When VACUUM table\_name RETAIN 0 is ran all of the historical versions of data are lost time travel can only provide the current state.

## Question 7

You have accidentally deleted records from a table called transactions, what is the easiest way to restore the records deleted or the previous state of the table? Prior to deleting the version of the table is 3 and after delete the version of the table is 4.

* **RESTORE TABLE transactions FROM VERSION as of 4**
* **RESTORE TABLE transactions TO VERSION as of 3**
* INSERT INTO OVERWRITE transactions

SELECT \* FROM transactions VERSION AS OF 3

MINUS

SELECT \* FROM transactions

* INSERT INTO OVERWRITE transactions

SELECT \* FROM transactions VERSION AS OF 4

INTERSECT

SELECT \* FROM transactions

* **COPY OVERWRITE transactions from VERSION as of 3**

**Explanation**

[RESTORE (Databricks SQL) | Databricks on AWS](https://docs.databricks.com/sql/language-manual/delta-restore.html)

1. RESTORE [TABLE] table\_name [TO] time\_travel\_version

Time travel supports using timestamp or version number

1. time\_travel\_version
2. { TIMESTAMP AS OF timestamp\_expression |
3. VERSION AS OF version }

* timestamp\_expression can be any one of:
  + '2018-10-18T22:15:12.013Z', that is, a string that can be cast to a timestamp
  + cast('2018-10-18 13:36:32 CEST' as timestamp)
  + '2018-10-18', that is, a date string
  + current\_timestamp() - interval 12 hours
  + date\_sub(current\_date(), 1)
  + Any other expression that is or can be cast to a timestamp

## Question 8

Create a schema called bronze using location ‘/mnt/delta/bronze’, and check if the schema exists before creating.

* **CREATE SCHEMA IF NOT EXISTS bronze LOCATION '/mnt/delta/bronze'**
* **CREATE SCHEMA bronze IF NOT EXISTS LOCATION '/mnt/delta/bronze'**
* **if IS\_SCHEMA('bronze'): CREATE SCHEMA bronze LOCATION '/mnt/delta/bronze'**
* **Schema creation is not available in metastore, it can only be done in Unity catalog UI**
* **Cannot create schema without a database**

**Explanation**

<https://docs.databricks.com/sql/language-manual/sql-ref-syntax-ddl-create-schema.html>

1. CREATE SCHEMA [ IF NOT EXISTS ] schema\_name [ LOCATION schema\_directory ]

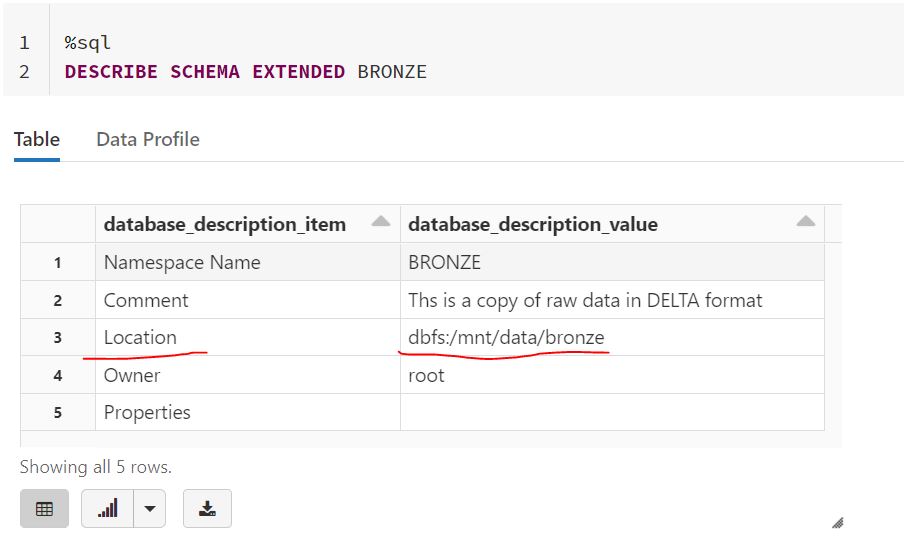
## Question 9

How do you check the location of an existing schema in Delta Lake?

* **Run SQL command SHOW LOCATION schema\_name**
* **Check unity catalog UI**
* **Use Data explorer**
* **Run SQL command DESCRIBE SCHEMA EXTENDED schema\_name**
* **Schemas are internally in-store external hive meta stores like MySQL or SQL Server**

**Explanation**

Here is an example of how it looks



## Question 10

Which of the below SQL commands create a Global temporary view?

* CREATE OR REPLACE TEMPORARY VIEW view\_name

AS SELECT \* FROM table\_name

* CREATE OR REPLACE LOCAL TEMPORARY VIEW view\_name

AS SELECT \* FROM table\_name

* CREATE OR REPLACE GLOBAL TEMPORARY VIEW view\_name

AS SELECT \* FROM table\_name

* CREATE OR REPLACE VIEW view\_name

AS SELECT \* FROM table\_name

* CREATE OR REPLACE LOCAL VIEW view\_name

AS SELECT \* FROM table\_name

**Explanation**

CREATE OR REPLACE GLOBAL TEMPORARY VIEW view\_name

AS SELECT \* FROM table\_name

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

* A 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 but if a cluster restarts a global temporary view is lost.

## Question 11

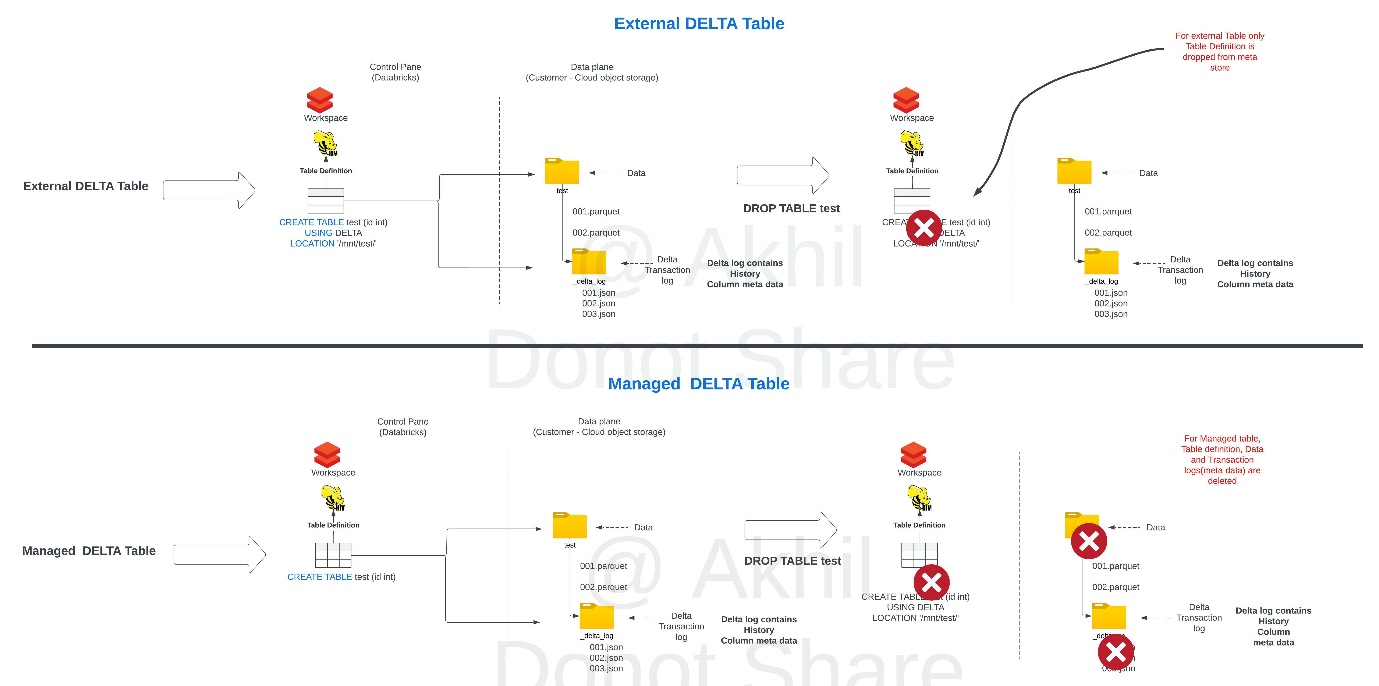
When you drop a managed table using SQL syntax DROP TABLE table\_name how does it impact metadata, history, and data stored in the table?

* **Drops table from meta store, drops metadata, history, and data in storage.**
* **Drops table from meta store and data from storage but keeps metadata and history in storage**
* **Drops table from meta store, meta data and history but keeps the data in storage**
* **Drops table but keeps meta data, history and data in storage**
* **Drops table and history but keeps meta data and data in storage**

**Explanation**

For a managed table, a drop command will drop everything from metastore and storage.

See the below image to understand the differences between dropping an external table.



## Question 12

The team has decided to take advantage of table properties to identify a business owner for each table, which of the following table DDL syntax allows you to populate a table property identifying the business owner of a table

* **CREATE TABLE inventory (id INT, units FLOAT)**

**SET TBLPROPERTIES business\_owner = 'supply chain'**

* **CREATE TABLE inventory (id INT, units FLOAT)**

**TBLPROPERTIES (business\_owner = 'supply chain')**

* **CREATE TABLE inventory (id INT, units FLOAT)**

**SET (business\_owner = ‘supply chain’)**

* **CREATE TABLE inventory (id INT, units FLOAT)**

**SET PROPERTY (business\_owner = ‘supply chain’)**

* **CREATE TABLE inventory (id INT, units FLOAT)**

**SET TAG (business\_owner = ‘supply chain’)**

**Explanation**

CREATE TABLE inventory (id INT, units FLOAT) TBLPROPERTIES (business\_owner = ‘supply chain’)

[Table properties and table options (Databricks SQL) | Databricks on AWS](https://docs.databricks.com/sql/language-manual/sql-ref-syntax-ddl-tblproperties.html#tblproperties)

Alter table command can used to update the TBLPROPERTIES

ALTER TABLE inventory SET TBLPROPERTIES(business\_owner , 'operations')

## Question 13:

Data science team has requested they are missing a column in the table called average price, this can be calculated using units sold and sales amt, which of the following SQL statements allow you to reload the data with additional column

* INSERT OVERWRITE sales

SELECT \*, salesAmt/unitsSold as avgPrice FROM sales

* CREATE OR REPLACE TABLE sales

AS SELECT \*, salesAmt/unitsSold as avgPrice FROM sales

* **MERGE INTO sales USING (SELECT \*, salesAmt/unitsSold as avgPrice FROM sales)**
* **OVERWRITE sales AS SELECT \*, salesAmt/unitsSold as avgPrice FROM sales**
* **COPY INTO SALES AS SELECT \*, salesAmt/unitsSold as avgPrice FROM sales**

**Explanation**

1. CREATE OR REPLACE TABLE sales
2. AS SELECT \*, salesAmt/unitsSold as avgPrice FROM sales

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 overwrite schema, only when spark.databricks.delta.schema.autoMerge.enabled is set true if this option is not enabled and if there is a schema mismatch command will fail.

## Question 14

You are working on a process to load external CSV files into a delta table by leveraging the COPY INTO command, but after running the command for the second time no data was loaded into the table name, why is that?

COPY INTO table\_name

FROM 'dbfs:/mnt/raw/\*.csv'

FILEFORMAT = CSV

* **COPY INTO only works one time data load**
* **Run REFRESH TABLE sales before running COPY INTO**
* **COPY INTO did not detect new files after the last load**
* **Use incremental = TRUE option to load new files**
* **COPY INTO does not support incremental load, use AUTO LOADER**

**Explanation**

The answer is COPY INTO did not detect new files after the last load,

COPY INTO keeps track of files that were successfully loaded into the table, the next time when the COPY INTO runs it skips them.

FYI, you can change this behavior by using COPY\_OPTIONS 'force'= 'true', when this option is enabled all files in the path/pattern are loaded.

COPY INTO table\_identifier

FROM [ file\_location | (SELECT identifier\_list FROM file\_location) ]

FILEFORMAT = data\_source

[FILES = [file\_name, ... | PATTERN = 'regex\_pattern']

[FORMAT\_OPTIONS ('data\_source\_reader\_option' = 'value', ...)]

[COPY\_OPTIONS 'force' = ('false'|'true')]

## Question 15

What is the main difference between the below two commands?

INSERT OVERWRITE table\_name

SELECT \* FROM table

CREATE OR REPLACE TABLE table\_name

AS SELECT \* FROM table

* **INSERT OVERWRITE replaces data by default, CREATE OR REPLACE replaces data and Schema by default**
* **INSERT OVERWRITE replaces data and schema by default, CREATE OR REPLACEreplaces data by default**
* **INSERT OVERWRITE maintains historical data versions by default, CREATE OR REPLACEclears the historical data versions by default**
* **INSERT OVERWRITE clears historical data versions by default, CREATE OR REPLACE maintains the historical data versions by default**
* **Both are same and results in identical outcomes**

**Explanation**

The answer is, INSERT OVERWRITE replaces data, CRAS replaces data and Schema

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 overwrite schema, only when spark.databricks.delta.schema.autoMerge.enabled is set true if this option is not enabled and if there is a schema mismatch command will fail.

## Question 16

Which of the following functions can be used to convert JSON string to Struct data type?

* **TO\_STRUCT (json value)**
* **FROM\_JSON (json value)**
* **FROM\_JSON (json value, schema of json)**
* **CONVERT (json value, schema of json)**
* **CAST (json value as STRUCT)**

**Explanation**

**Syntax**

Copy

1. from\_json(jsonStr, schema [, options])

**Arguments**

* jsonStr: A STRING expression specifying a row of CSV data.
* schema: A STRING literal or invocation of [schema\_of\_json function (Databricks SQL)](https://docs.microsoft.com/en-us/azure/databricks/sql/language-manual/functions/schema_of_json).
* options: An optional MAP<STRING,STRING> literal specifying directives.

Refer documentation for more details,

<https://docs.microsoft.com/en-us/azure/databricks/sql/language-manual/functions/from_json>

## Question 17

You are working on a marketing team request to identify customers with the same information between two tables CUSTOMERS\_2021 and CUSTOMERS\_2020 each table contains 25 columns with the same schema, You are looking to identify rows that match between two tables across all columns, which of the following can be used to perform in SQL

* SELECT \* FROM CUSTOMERS\_2021

UNION

SELECT \* FROM CUSTOMERS\_2020

* SELECT \* FROM CUSTOMERS\_2021

UNION ALL

SELECT \* FROM CUSTOMERS\_2020

* SELECT \* FROM CUSTOMERS\_2021 C1

INNER JOIN CUSTOMERS\_2020 C2

ON C1.CUSTOMER\_ID = C2.CUSTOMER\_ID

* SELECT \* FROM CUSTOMERS\_2021

INTERSECT

SELECT \* FROM CUSTOMERS\_2020

* SELECT \* FROM CUSTOMERS\_2021

EXCEPT

SELECT \* FROM CUSTOMERS\_2020

**Explanation**

Answer is,

SELECT \* FROM CUSTOMERS\_2021

INTERSECT

SELECT \* FROM CUSTOMERS\_2020

To compare all the rows between both the tables across **all the columns** using intersect will help us achieve that, an inner join is only going to check if the same column value exists across both the tables on a single column.

**INTERSECT [ALL | DISTINCT]**

* Returns the set of rows which are in both subqueries.

If ALL is specified a row that appears multiple times in the subquery1 as well as in subquery will be returned multiple times.

If DISTINCT is specified the result does not contain duplicate rows. This is the default.

## Question 18

You are looking to process the data based on two variables, one to check if the department is supply chain and second to check if process flag is set to True

* **if department = “supply chain” & process:**
* **if department == “supply chain” && process:**
* **if department == “supply chain” & process == TRUE:**
* **if department == “supply chain” & if process == TRUE:**
* **if department == "supply chain" and process:**

## Question 19

You were asked to create a notebook that can take department as a parameter and process the data accordingly, which is the following statements result in storing the notebook parameter into a python variable

* **SET department = dbutils.widget.get("department")**
* **ASSIGN department == dbutils.widget.get("department")**
* **department = dbutils.widget.get("department")**
* **department = notebook.widget.get("department")**
* **department = notebook.param.get("department")**

**Explanation**

The answer is department = dbutils.widget.get("department")

Refer to additional documentation here

<https://docs.databricks.com/notebooks/widgets.html>

## Question 20

Which of the following statements can successfully read the notebook widget and pass the python variable to a SQL statement in a Python notebook cell?

order\_date = dbutils.widgets.get("widget\_order\_date")

spark.sql(f"SELECT \* FROM sales WHERE orderDate = 'f{order\_date }'")

order\_date = dbutils.widgets.get("widget\_order\_date")

spark.sql(f"SELECT \* FROM sales WHERE orderDate = 'order\_date' ")

order\_date = dbutils.widgets.get("widget\_order\_date")

spark.sql(f”SELECT \* FROM sales WHERE orderDate = '${order\_date }' ")

order\_date = dbutils.widgets.get("widget\_order\_date")

spark.sql(f"SELECT \* FROM sales WHERE orderDate = '{order\_date}' ")

order\_date = dbutils.widgets.get("widget\_order\_date")

spark.sql("SELECT \* FROM sales WHERE orderDate = order\_date")

## Question 21

The below spark command is looking to create a summary table based customerId and the number of times the customerId is present in the event\_log delta table and write a one-time micro-batch to a summary table, fill in the blanks to complete the query.

spark.\_\_\_\_\_\_\_\_\_

.format("delta")

.table("events\_log")

.groupBy("customerId")

.count()

.\_\_\_\_\_\_\_

.format("delta")

.outputMode("complete")

.option("checkpointLocation", "/tmp/delta/eventsByCustomer/\_checkpoints/")

.trigger(\_\_\_\_\_\_)

.table("target\_table")

* **writeStream, readStream, once**
* **readStream, writeStream, once**
* **writeStream, processingTime = once**
* **writeStream, readStream, once = True**
* **readStream, writeStream, once = True**

**Explanation**

The answer is readStream, writeStream, once = True.

spark.**readStream**

  .format("delta")

  .table("events\_log")

  .groupBy("customerId")

  .count()

  .**writeStream**

  .format("delta")

  .outputMode("complete")

  .option("checkpointLocation", "/tmp/delta/eventsByCustomer/\_checkpoints/")

  .trigger(**once = True**)

  .table("target\_table")

## Question 22

You would like to build a spark streaming process to read from a Kafka queue and write to a Delta table every 15 minutes, what is the correct trigger option

* **trigger("15 minutes")**
* **trigger(process "15 minutes")**
* **trigger(processingTime = 15)**
* **trigger(processingTime = "15 Minutes")**
* **trigger(15)**

**Explanation**

The answer is trigger(processingTime = "15 Minutes")

Triggers:

* **Unspecified**

This is the default. This is equivalent to using processingTime="500ms"

* ***Fixed interval micro-batches   .trigger(processingTime="2 minutes")***

The query will be executed in micro-batches and kicked off at the user-specified intervals

* **One-time micro-batch .trigger(once=True)**

The query will execute a single micro-batch to process all the available data and then stop on its own

* **One-time micro-batch.trigger .trigger(availableNow=True)  -- New feature a better version of (once=True)**

Databricks supports trigger(availableNow=True) in Databricks Runtime 10.2 and above for Delta Lake and Auto Loader sources. This functionality combines the batch processing approach of trigger once with the ability to configure batch size, resulting in multiple parallelized batches that give greater control for right-sizing batches and the resultant files.

## Question 23

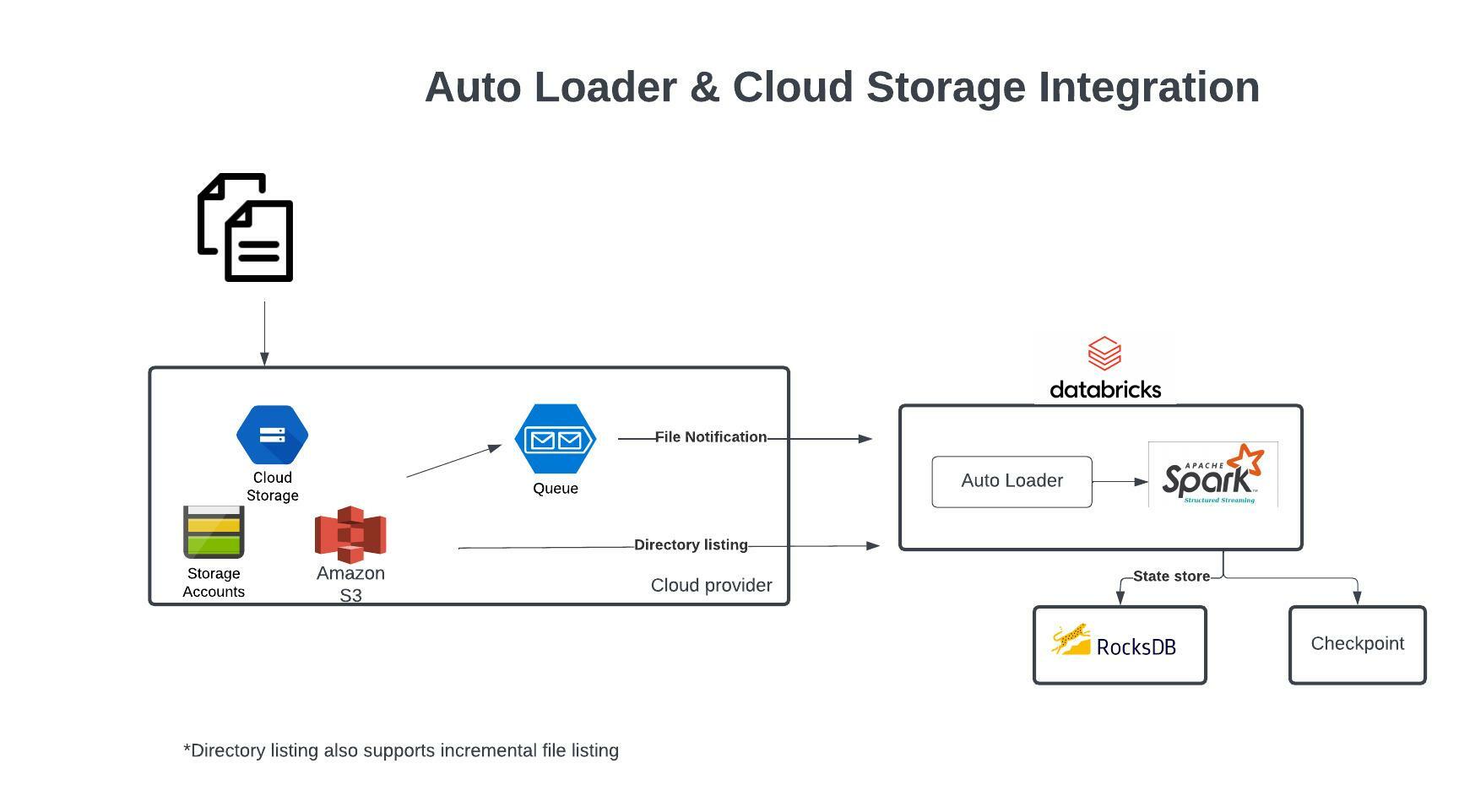
Which of the following scenarios is the best fit for the AUTO LOADER solution?

* **Efficiently process new data incrementally from cloud object storage**
* **Incrementally process new streaming data from Apache Kafa into delta lake**
* **Incrementally process new data from relational databases like MySQL**
* **Efficiently copy data from data lake location to another data lake location**
* **Efficiently move data incrementally from one delta table to another delta table**

**Explanation**

The answer is, Efficiently process new data incrementally from cloud object storage.

Please note:  AUTO LOADER only works on data/files located in cloud object storage like S3 or Azure Blob Storage it does not have the ability to read other data sources, although AUTO LOADER is built on top of structured streaming it only supports files in the cloud object storage. If you want to use Apache Kafka then you can just use structured streaming.



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.

Refer to more documentation here,

https://docs.microsoft.com/en-us/azure/databricks/ingestion/auto-loader

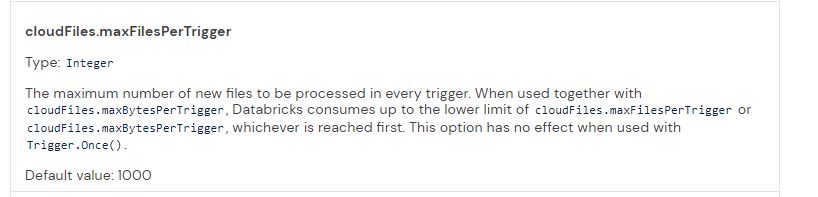
## Question 24

You had AUTO LOADER to process millions of files a day and noticed slowness in load process, so you scaled up the Databricks cluster but realized the performance of the Auto loader is still not improving, what is the best way to resolve this.

* **AUTO LOADER is not suitable to process millions of files a day**
* **Setup a second AUTO LOADER process to process the data**
* **Increase the maxFilesPerTrigger option to a sufficiently high number**
* **Copy the data from cloud storage to local disk on the cluster for faster access**
* **Merge files to one large file**

**Explanation**

The default value of maxFilesPerTrigger is 1000 it can be increased to a much higher number but will require a much larger compute to process.



<https://docs.databricks.com/ingestion/auto-loader/options.html>

## Question 25

The current ELT pipeline is receiving data from the operations team once a day so you had setup an AUTO LOADER process to run once a day using trigger (Once = True) and scheduled a job to run once a day, operations team recently rolled out a new feature that allows them to send data every 1 min, what changes do you need to make to AUTO LOADER to process the data every 1 min.

* **Convert AUTO LOADER to structured streaming**
* **Change AUTO LOADER trigger to .trigger(ProcessingTime = "1 minute")**
* **Setup a job cluster run the notebook once a minute**
* **Enable stream processing**
* **Change AUTO LOADER trigger to ("1 minute")**

## Question 26

What is the purpose of the bronze layer in a Multi-hop Medallion architecture?

* **Copy of raw data, easy to query and ingest data for downstream processes.**
* **Powers ML applications**
* **Data quality checks, corrupt data quarantined**
* **Contain aggregated data that is to be consumed into Silver**
* **Reduces data storage by compressing the data**

**Explanation**

The answer is, copy of raw data, easy to query and ingest data for downstream processes,

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

Here are the typical role of Bronze Layer in a medallion architecture.

Bronze Layer:

1. Raw copy of ingested data

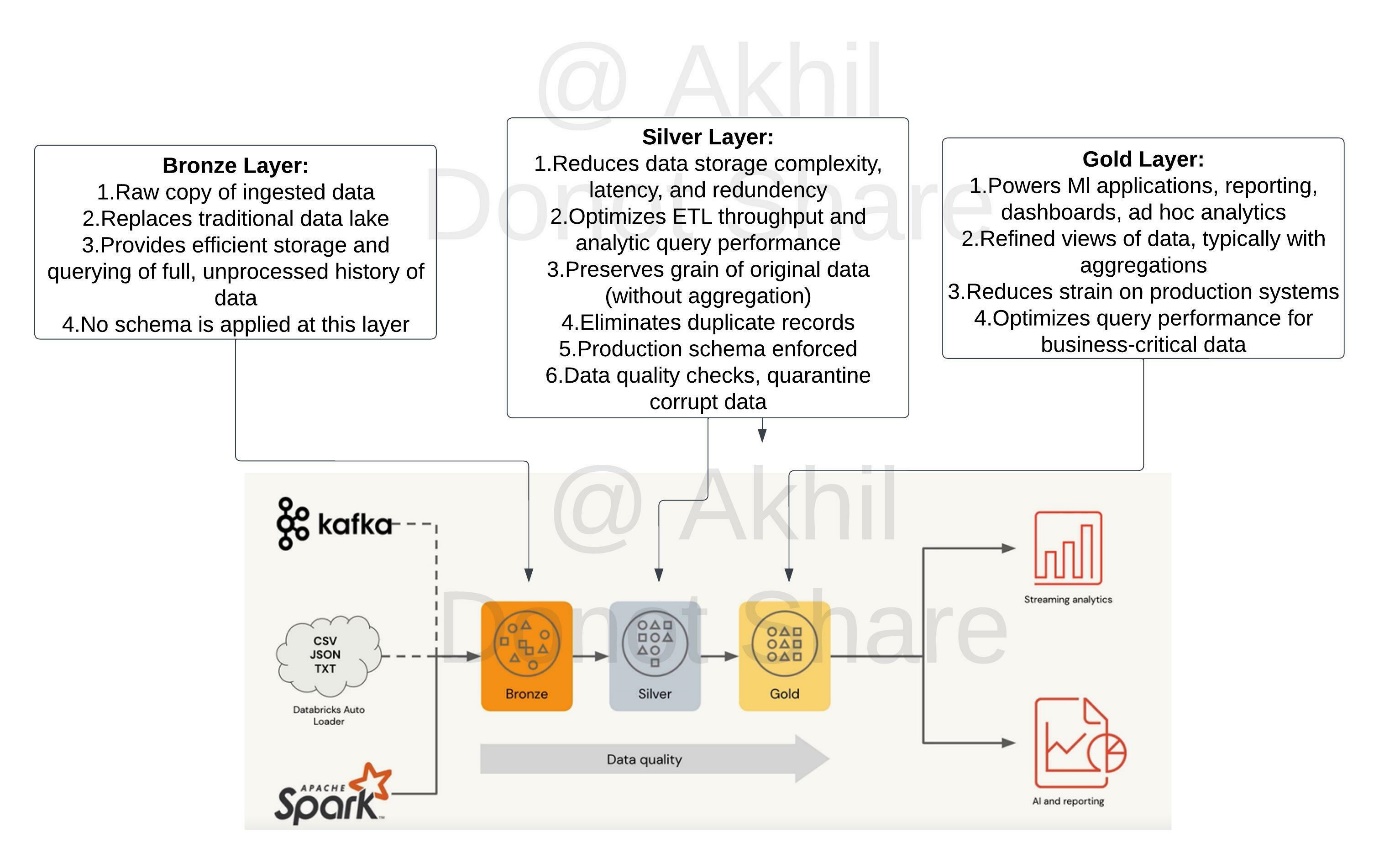
2. Replaces traditional data lake

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.*



## Question 27

What is the purpose of the silver layer in a Multi hop architecture?

* **Replaces a traditional data lake**
* **Efficient storage and querying of full, unprocessed history of data**
* **Eliminates duplicate data, quarantines bad data**
* **Refined views with aggregated data**
* **Optimized query performance for business-critical data**

**Explanation**

[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)

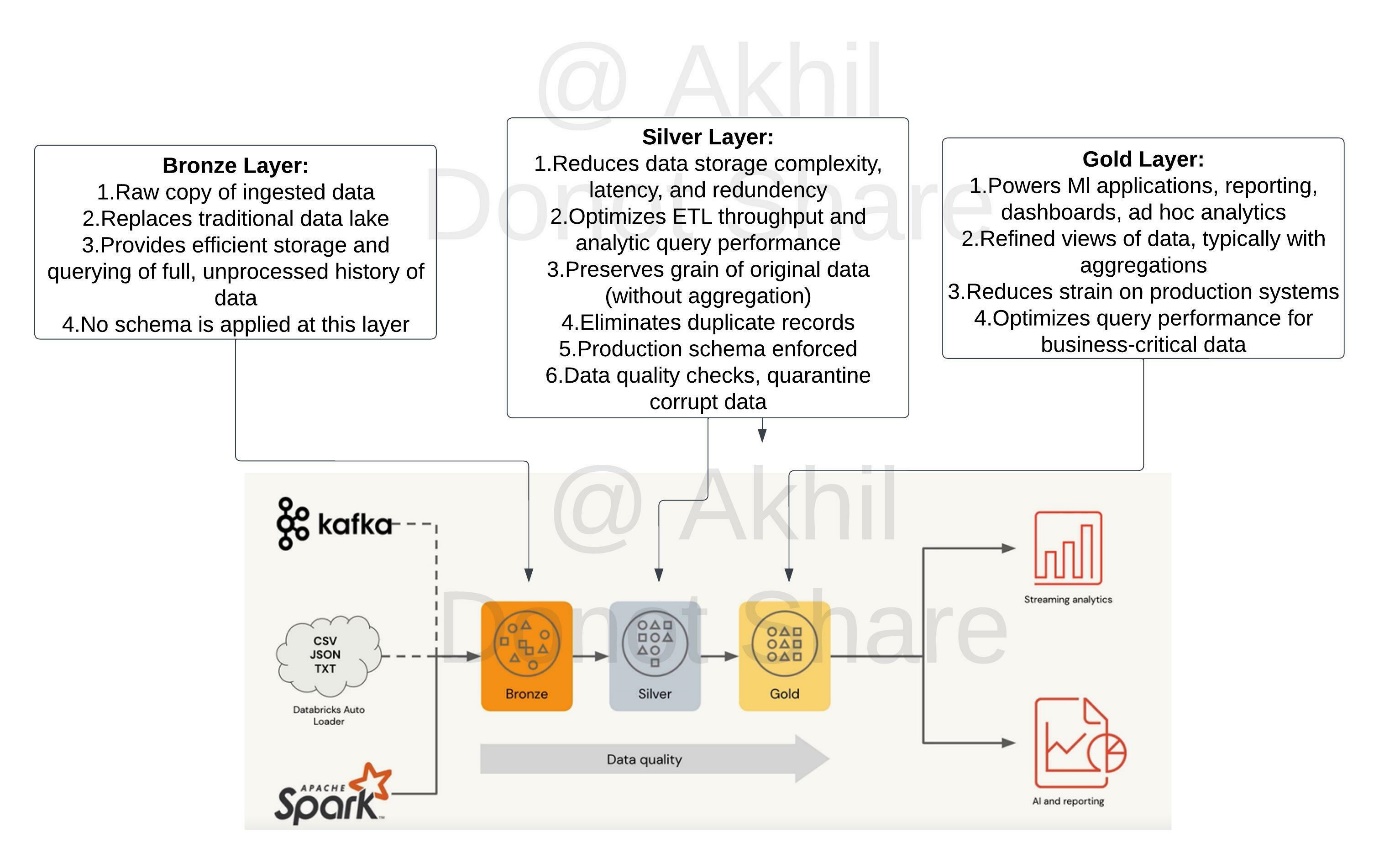
4. Eliminates duplicate records

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.



## Question 28

What is the purpose of gold layer in Multi hop architecture?

* **Optimizes ETL throughput and analytic query performance**
* **Eliminate duplicate records**
* **Preserves grain of original data, without any aggregations**
* **Data quality checks and schema enforcement**
* **Optimized query performance for business-critical data**

**Explanation**

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

Gold Layer:

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

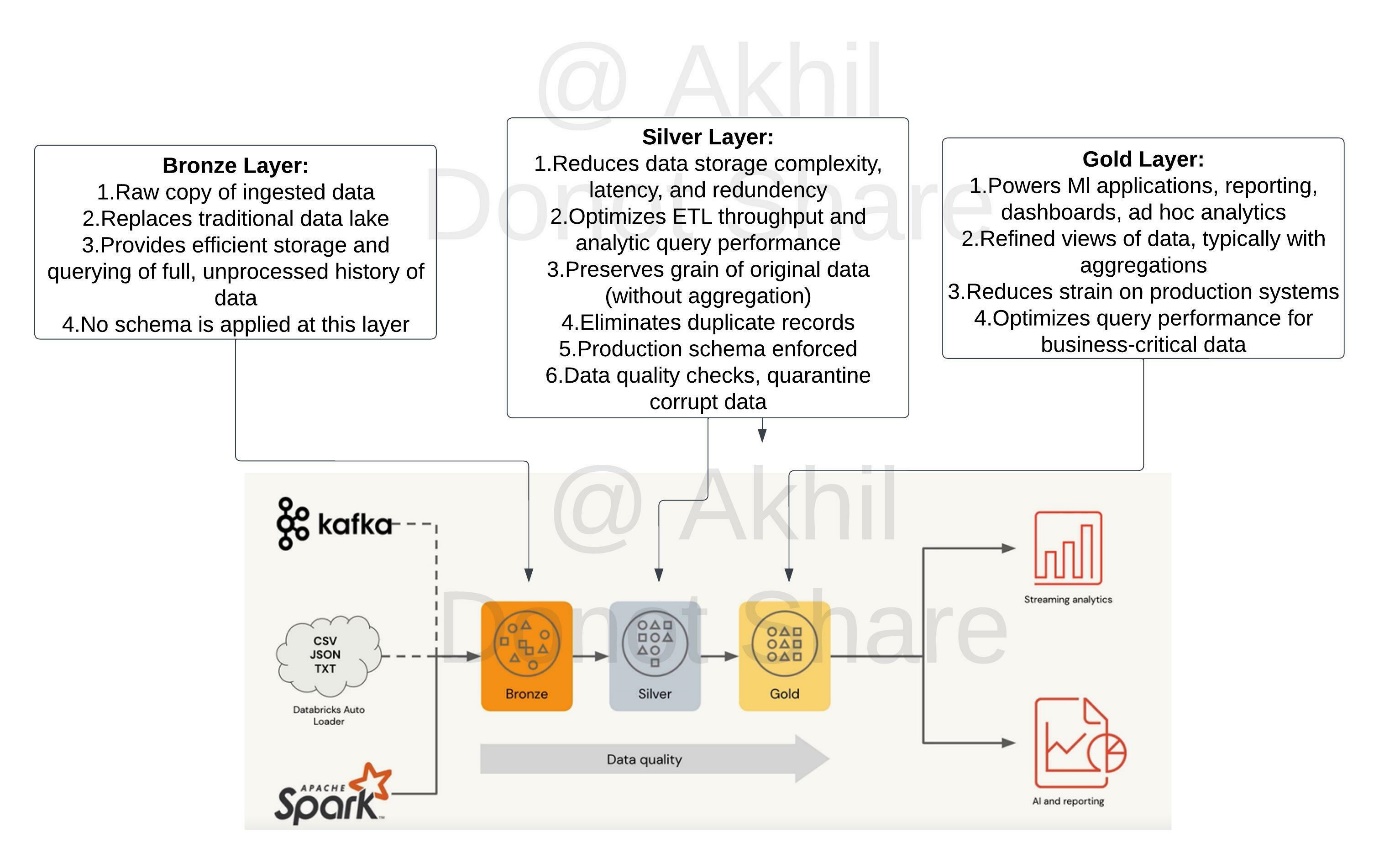
2. Refined views of data, typically with aggregations

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.



## Question 29:

The Delta Live Tables Pipeline is configured to run in Development mode using the Triggered Pipeline Mode. what is the expected outcome after clicking Start to update the pipeline?

* **All datasets will be updated once and the pipeline will shut down. The compute resources will be terminated**
* **All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will be deployed for the update and terminated when the pipeline is stopped**
* **All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will persist after the pipeline is stopped to allow for additional development and testing**
* **All datasets will be updated once and the pipeline will shut down. The compute resources will persist to allow for additional development and testing**
* **All datasets will be updated continuously and the pipeline will not shut down. The compute resources will persist with the pipeline**

**Explanation**

The answer is All datasets will be updated once and the pipeline will shut down. The compute resources will persist to allow for additional testing.

DLT pipeline supports two modes Development and Production, you can switch between the two based on the stage of your development and deployment lifecycle.

**Development and production modes**

When you run your pipeline in development mode, the Delta Live Tables system:

* Reuses a cluster to avoid the overhead of restarts.
* Disables pipeline retries so you can immediately detect and fix errors.

In production mode, the Delta Live Tables system:

* Restarts the cluster for specific recoverable errors, including memory leaks and stale credentials.
* Retries execution in the event of specific errors, for example, a failure to start a cluster.

Use the  buttons in the Pipelines UI to switch between development and production modes. By default, pipelines run in development mode.

Switching between development and production modes only controls cluster and pipeline execution behavior. Storage locations must be configured as part of pipeline settings and are not affected when switching between modes.

Please review additional DLT concepts using below link

https://docs.databricks.com/data-engineering/delta-live-tables/delta-live-tables-concepts.html#delta-live-tables-concepts

## Question 30

The Delta Live Table Pipeline is configured to run in Production mode using the continuous Pipeline Mode. what is the expected outcome after clicking Start to update the pipeline?

* **All datasets will be updated once and the pipeline will shut down. The compute resources will be terminated**
* **All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will be deployed for the update and terminated when the pipeline is stopped**
* **All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will persist after the pipeline is stopped to allow for additional testing**
* **All datasets will be updated once and the pipeline will shut down. The compute resources will persist to allow for additional testing**
* **All datasets will be updated continuously and the pipeline will not shut down. The compute resources will persist with the pipeline**

**Explanation**

The answer is,

All datasets will be updated continuously and the pipeline will not shut down. The compute resources will persist with the pipeline until it is shut down since the execution mode is chosen to be continuous. It does not matter if the pipeline mode is development or production, pipeline mode only matters during the pipeline initialization.

DLT pipeline supports two modes Development and Production, you can switch between the two based on the stage of your development and deployment lifecycle.

**Development and production modes**

**Development:**

When you run your pipeline in development mode, the Delta Live Tables system:

* Reuses a cluster to avoid the overhead of restarts.
* Disables pipeline retries so you can immediately detect and fix errors.

**Production:**

*In production mode, the Delta Live Tables system:*

* *Restarts the cluster for specific recoverable errors, including memory leaks and stale credentials.*
* *Retries execution in the event of specific errors, for example, a failure to start a cluster.*

Use the  buttons in the Pipelines UI to switch between development and production modes. By default, pipelines run in development mode.

Switching between development and production modes only controls cluster and pipeline execution behavior. Storage locations must be configured as part of pipeline settings and are not affected when switching between modes.

**Delta Live Tables supports two different modes of execution:**

Triggered pipelines update each table with whatever data is currently available and then stop the cluster running the pipeline. Delta Live Tables automatically analyzes the dependencies between your tables and starts by computing those that read from external sources. Tables within the pipeline are updated after their dependent data sources have been updated.

*Continuous pipelines update tables continuously as input data changes. Once an update is started, it continues to run until manually stopped. Continuous pipelines require an always-running cluster but ensure that downstream consumers have the most up-to-date data*

Please review additional DLT concepts using the below link

https://docs.databricks.com/data-engineering/delta-live-tables/delta-live-tables-concepts.html#delta-live-tables-concepts

## Question 31

You are working to set up two notebooks to run on a schedule, the second notebook is dependent on the first notebook but both notebooks need different types of compute to run in an optimal fashion, what is the best way to set up these notebooks as jobs?

* **Use DELTA LIVE PIPELINES instead of notebook tasks**
* **A Job can only use single cluster, setup job for each notebook and use job dependency to link both jobs together**
* **Each task can use different cluster, add these two notebooks as two tasks in a single job with linear dependency and modify the cluster as needed for each of the tasks**
* **Use a single job to setup both notebooks as individual tasks, but use the cluster API to setup the second cluster before the start of second task**
* **Use a very large cluster to run both the tasks in a single job**

**Explanation**

Tasks in Jobs support different clusters for each task in the same job.

## Question 32

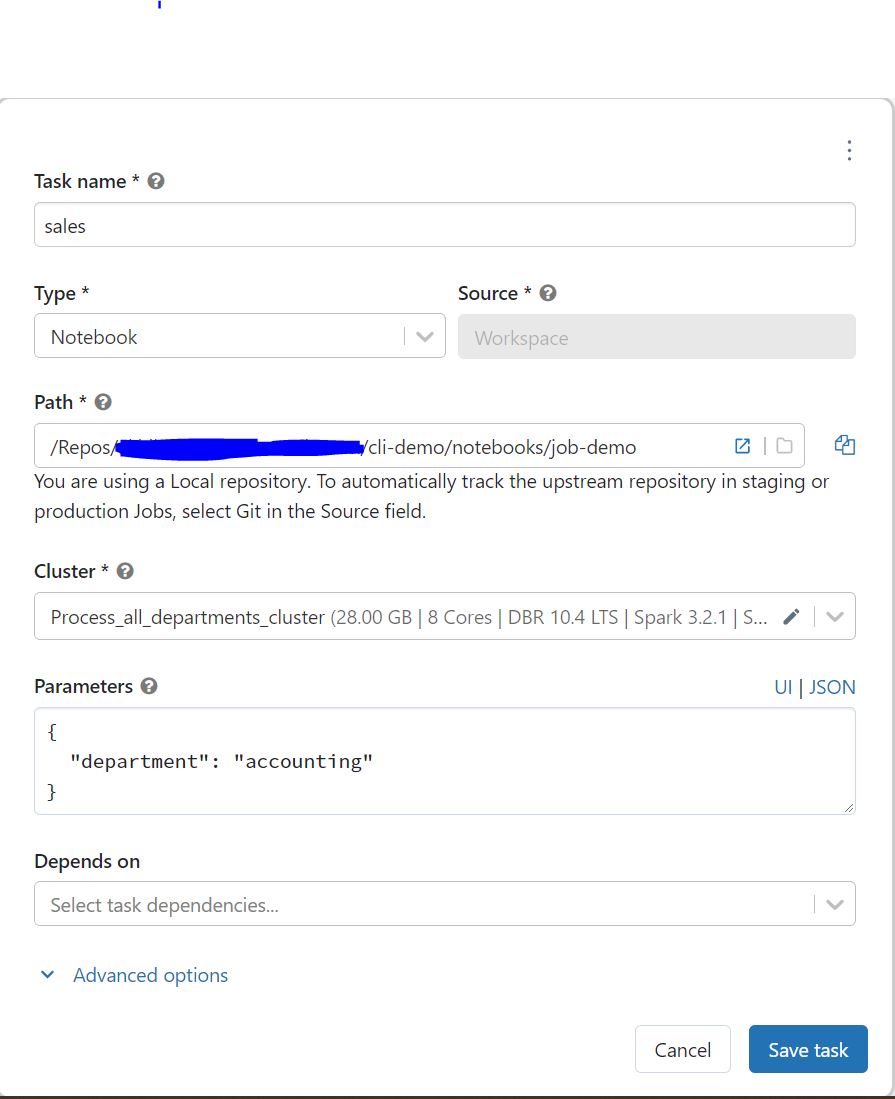
You are tasked to set up a set notebook as a job for six departments and each department can run the task parallelly, the notebook takes an input parameter dept number to process the data by department, how do you go about to setup this up in job?

* **Use a single notebook as task in the job and use dbutils.notebook.run to run each notebook with parameter in a different cell**
* **A task in the job cannot take an input parameter, create six notebooks with hardcoded dept number and setup six tasks with linear dependency in the job**
* **A task accepts key-value pair parameters, creates six tasks pass department number as parameter foreach task with no dependency in the job as they can all run in parallel.**
* **A parameter can only be passed at the job level, create six jobs pass department number to each job with linear job dependency**
* **A parameter can only be passed at the job level, create six jobs pass department number to each job with no job dependency**

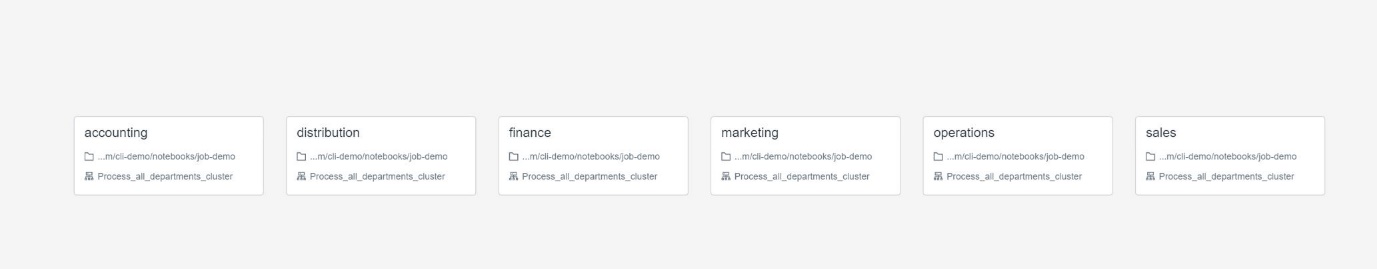
**Explanation**

Here is how you setup

Create a single job and six tasks with the same notebook and assign a different parameter for each task ,



All tasks are added in a single job and can run parallel either using single shared cluster or with individual clusters.



## Question 33

You are asked to setup two tasks in a databricks job, the first task runs a notebook to download the data from a remote system, and the second task is a DLT pipeline that can process this data, how do you plan to configure this in Jobs UI

* **Single job cannot have a notebook task and DLT Pipeline task, use two different jobs with linear dependency.**
* **Jobs UI does not support DTL pipeline, setup the first task using jobs UI and setup the DLT to run in continuous mode.**
* **Jobs UI does not support DTL pipeline, setup the first task using jobs UI and setup the DLT to run in trigger mode.**
* **Single job can be used to setup both notebook and DLT pipeline, use two different tasks with linear dependency.**
* **Add first step in the DLT pipeline and run the DLT pipeline as triggered mode in JOBS UI**

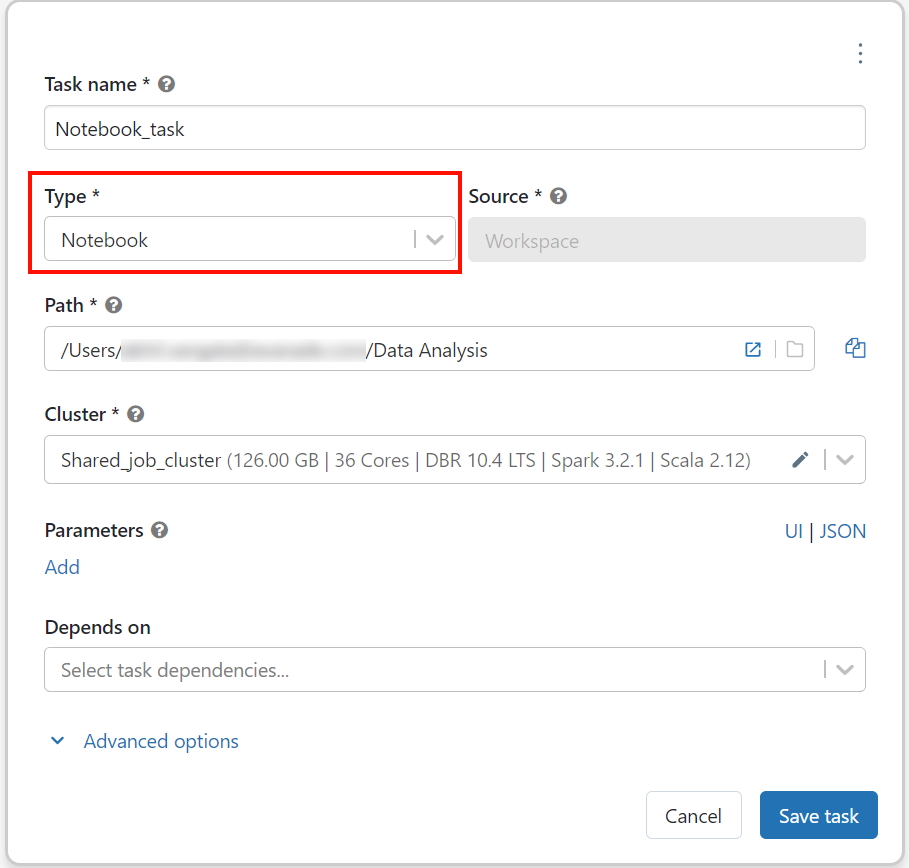
**Explanation**

The answer is Single job can be used to set up both notebook and DLT pipeline, use two different tasks with linear dependency,

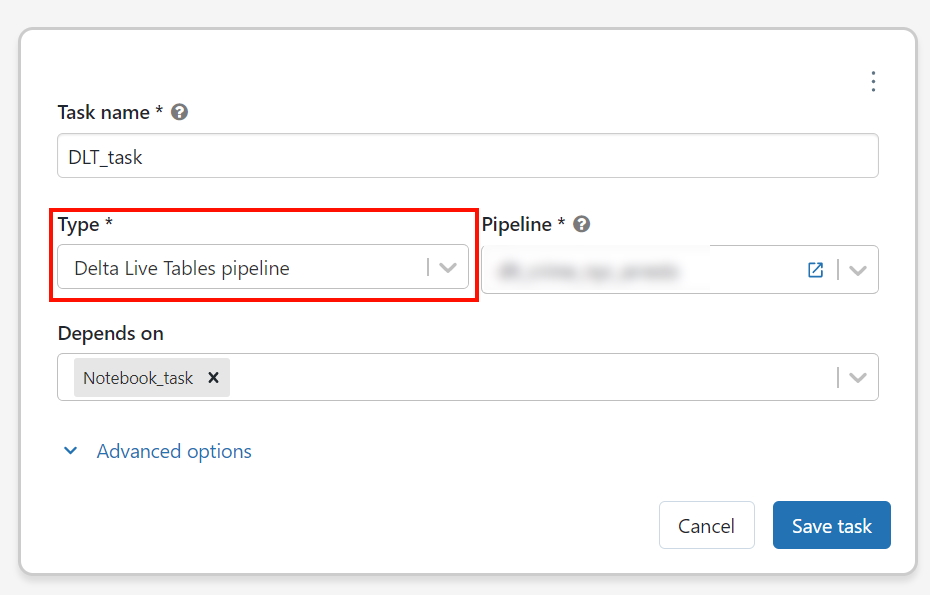
Here is the JOB UI

1. Create a notebook task
2. Create DLT task
   1. add notebook task as dependency
3. Final view

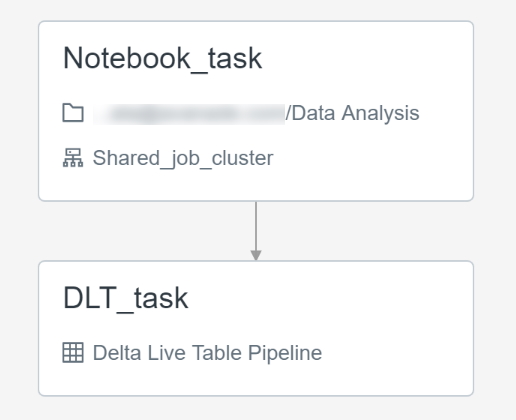
Create the notebook task



DLT task



Final view



## Question 34

You are asked to set up an alert to notify in an email every time a KPI indicater increases beyond a threshold value, team also asked you to include the actual value in the alert email notification.

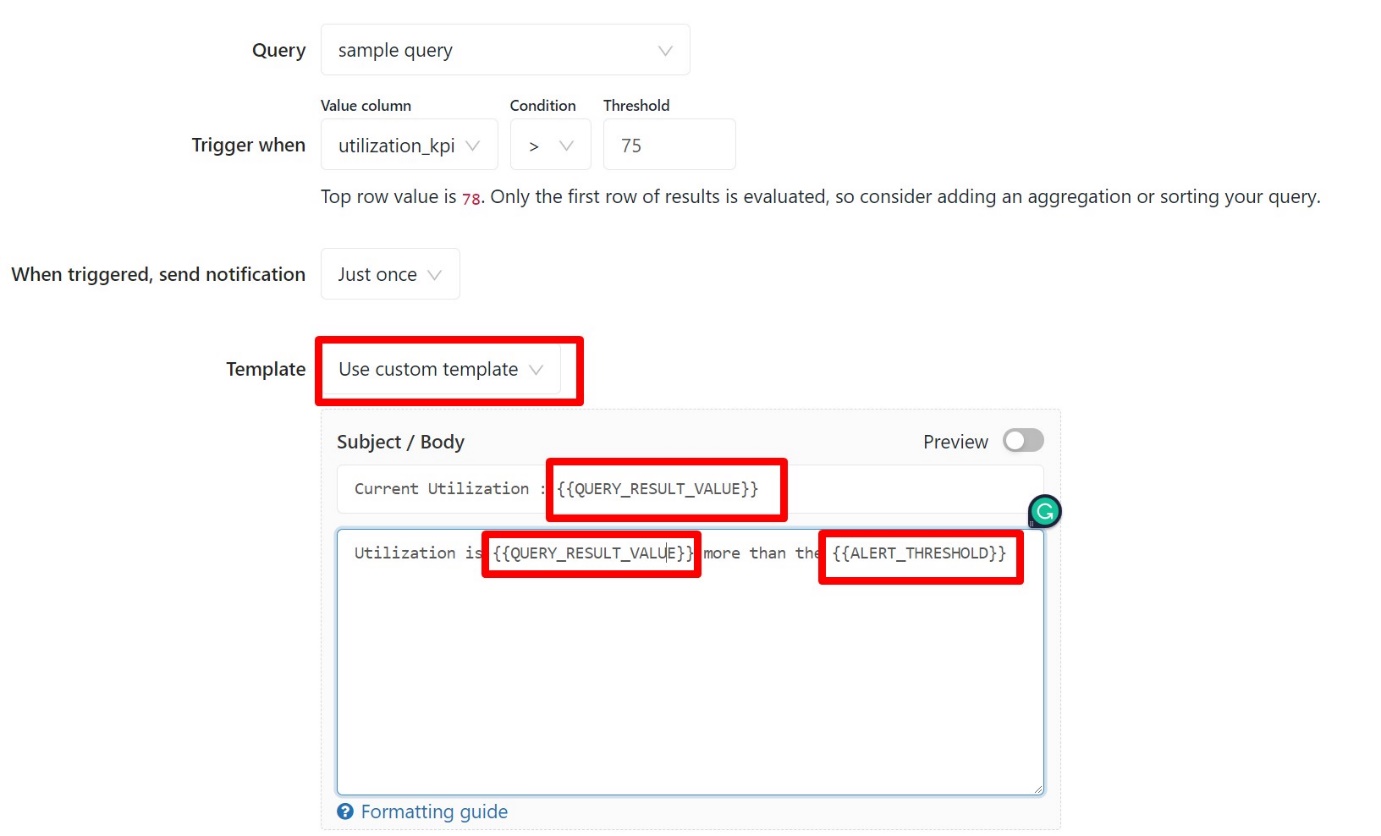
* **Use notebook and python code to run every minute, using python variables to capture send the information in an email**
* **Setup an alert but use the default template to notify the message in email’s subject**
* **Setup an alert but use the custom template to notify the message in email’s subject**
* **Use the webhook destination instead so alert message can be customized**
* **Use custom email hook to customize the message**

**Explanation**

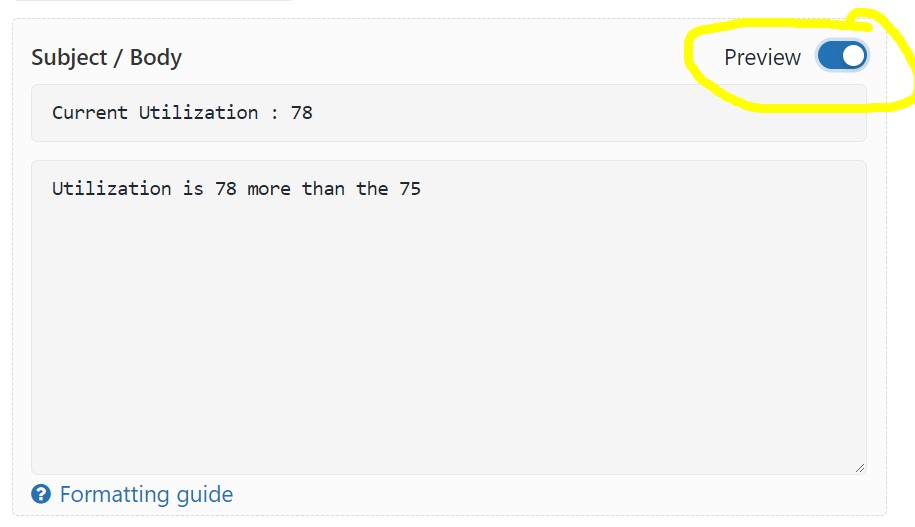
Alerts support custom template supports using variables to customize the default message, set up the query to compare the KPI current value to the threshold and use the variable QUERY\_RESULT\_VALUE to display the value in the email notification.

here is a simple alert, that uses variables in the custom template to present these values in the email notification message, when the alert is fired these variables get replaced with actual values.

Alert with custom template

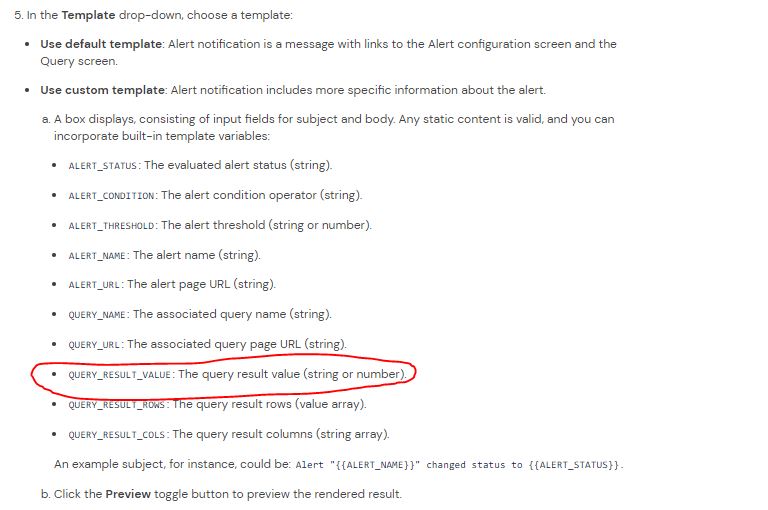


When you enable preview you can see how the alert looks when you substitute the variables.



Below are additional template variables available to you with the custom template.

[Alerts | Databricks on AWS](https://docs.databricks.com/sql/user/alerts/index.html#id2)



## Question 35

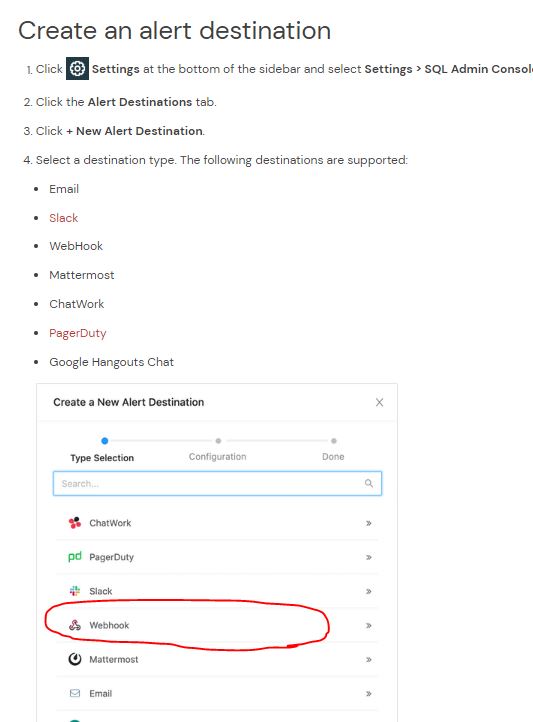
Operations team is using a centralized data quality monitoring system, a user can publish data quality metrics through a webhook, you were asked to develop a process to send messages using a webhook if there is atleast one duplicate record, which of the following approaches can be taken to integrate an alert with current data quality monitoring system

* **Use notebook and Jobs to use python to publish DQ metrics**
* **Setup an alert to send an email, use python to parse email, and publish a webhook message**
* **Setup an alert with custom template**
* **Setup an alert with custom Webhook destination**
* **Setup an alert with dynamic template**

**Explanation**

Alerts supports multiple destinations, email is the default destination.

[Alert destinations | Databricks on AWS](https://docs.databricks.com/sql/admin/alert-destinations.html)



## Question 36

You are currently working with the application team to setup a SQL Endpoint point, once the team started consuming the SQL Endpoint you noticed that during peak hours as the number of concurrent users increases you are seeing degradation in the query performance and the same queries are taking longer to run, which of the following steps can be taken to resolve the issue?

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

**Explanation**

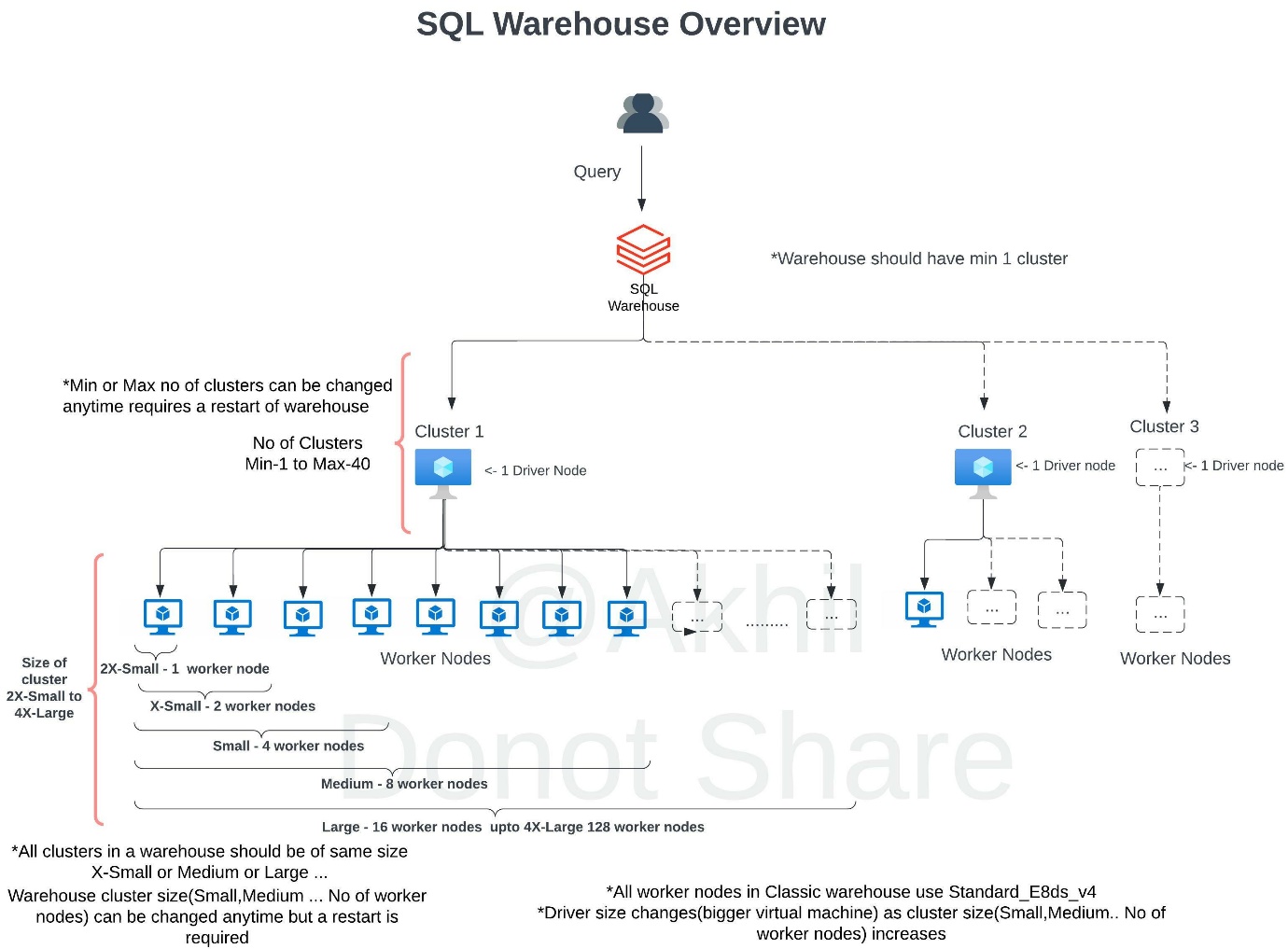
The answer is, They can increase the maximum bound of the SQL endpoint’s scaling range, when you increase the max scaling range more clusters are added so queries instead of waiting in the queue can start running using available clusters, see below for more explanation.

**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, 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:

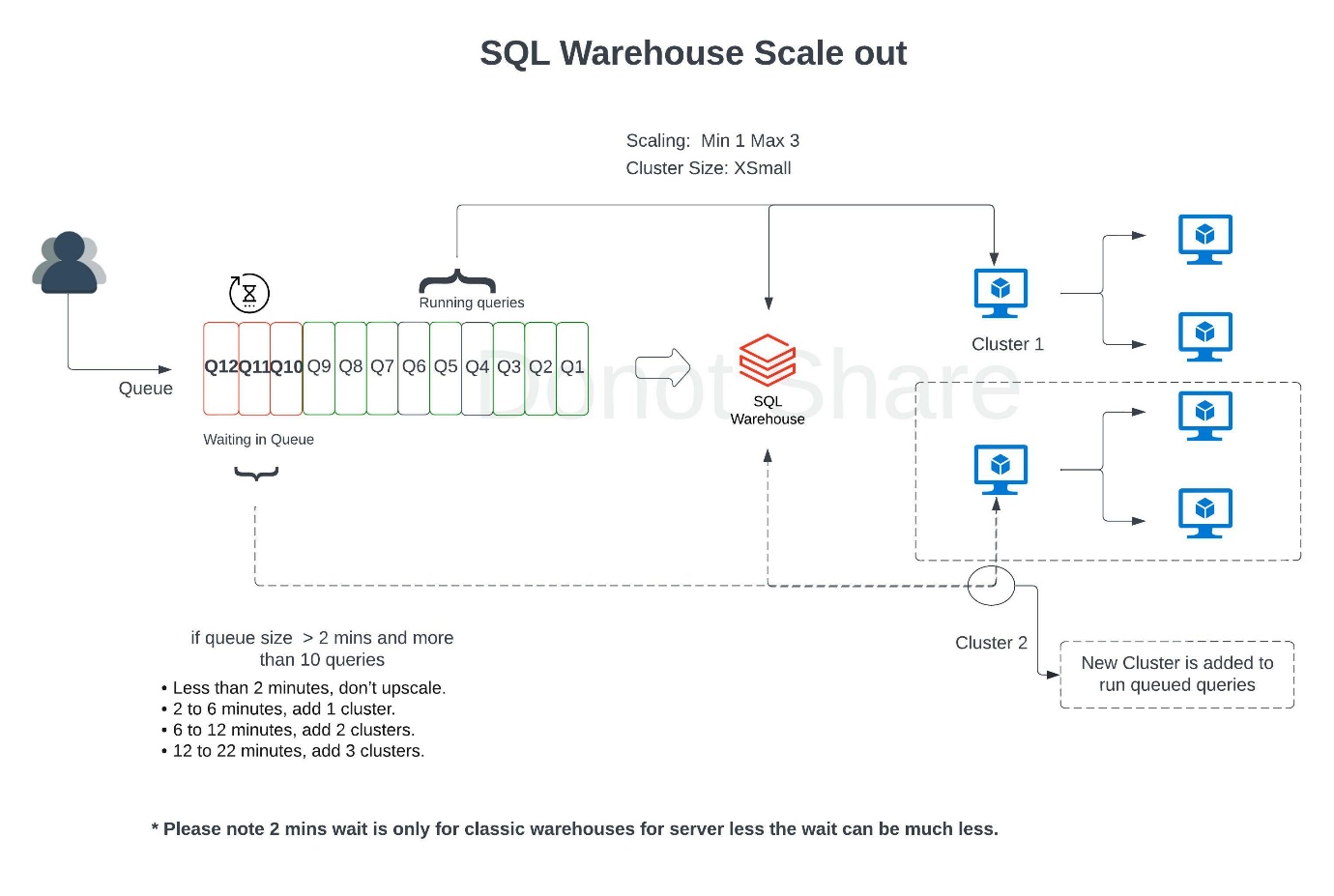


SQL endpoint(SQL Warehouse) scales horizontally(scale-out) and vertical (scale-up), you have to understand when to use what.

**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.



## Question 37

The data engineering team is using a bunch of SQL queries to review data quality and monitor the ETL job every day, which of the following approaches can be used to set up a schedule and automate this process?

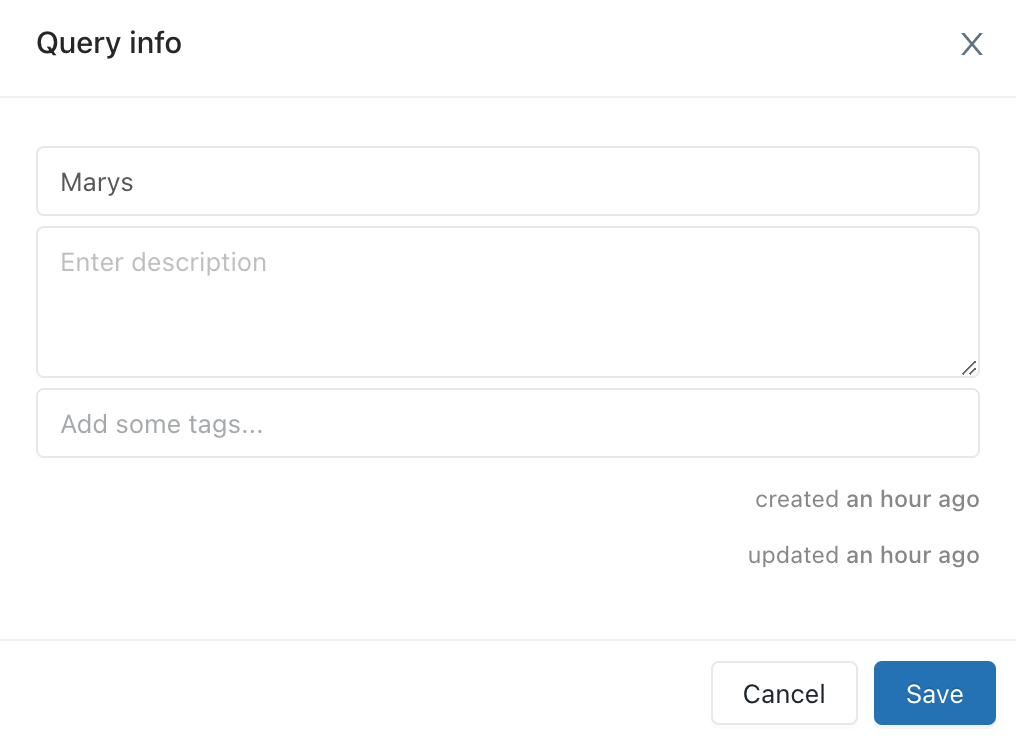
* **They can schedule the query to run every 1 day from the Jobs UI**
* **They can schedule the query to refresh every 1 day from the query’s page in Databricks SQL.**
* **They can schedule the query to run every 12 hours from the Jobs UI.**
* **They can schedule the query to refresh every 1 day from the SQL endpoint’s page in Databricks SQL.**
* **They can schedule the query to refresh every 12 hours from the SQL endpoint’s page in Databricks SQL**

**Explanation**

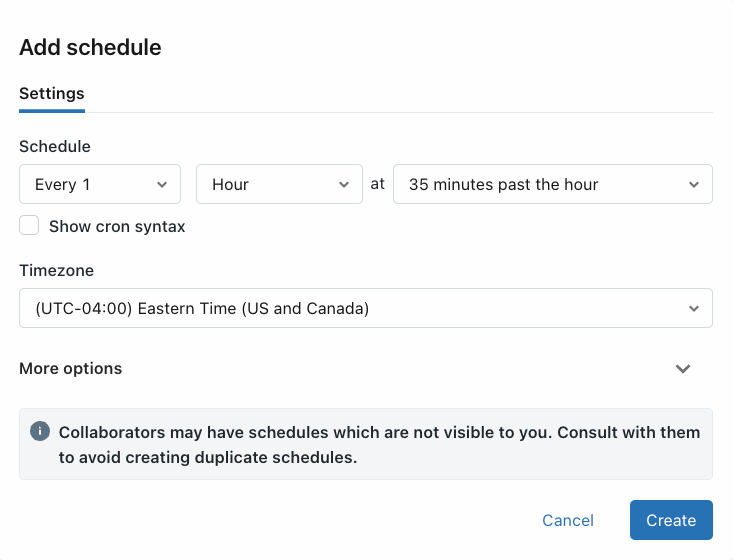
Individual queries can be refreshed on a schedule basis,

To set the schedule:

1. Click the query info tab.



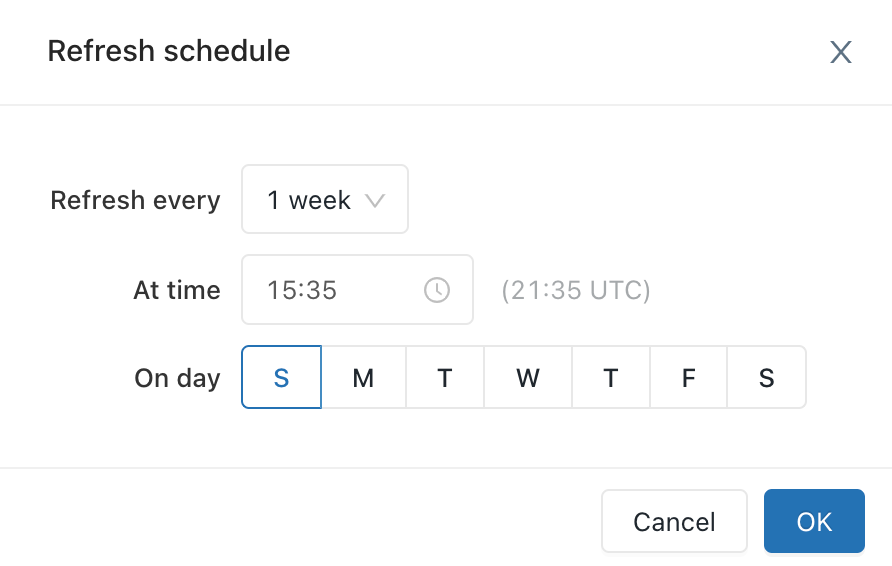
1. Click the link to the right of **Refresh Schedule** to open a picker with schedule intervals.



1. Set the schedule.

The picker scrolls and allows you to choose:

* + An interval: 1-30 minutes, 1-12 hours, 1 or 30 days, 1 or 2 weeks
  + A time. The time selector displays in the picker only when the interval is greater than 1 day and the day selection is greater than 1 week. When you schedule a specific time, Databricks SQL takes input in your computer’s timezone and converts it to UTC. If you want a query to run at a certain time in UTC, you must adjust the picker by your local offset. For example, if you want a query to execute at 00:00 UTC each day, but your current timezone is PDT (UTC-7), you should select 17:00 in the picker:



1. Click **OK**.

Your query will run automatically.

If you experience a scheduled query not executing according to its schedule, you should manually trigger the query to make sure it doesn’t fail. However, you should be aware of the following:

* + If you schedule an interval—for example, “every 15 minutes”—the interval is calculated from the *last successful execution*. If you manually execute a query, the scheduled query will not be executed until the interval has passed.
  + If you schedule a time, Databricks SQL waits for the results to be “outdated”. For example, if you have a query set to refresh every Thursday and you manually execute it on Wednesday, by Thursday the results will still be considered “valid”, so the query wouldn’t be scheduled for a new execution. Thus, for example, when setting a weekly schedule, check the last query execution time and expect the scheduled query to be executed on the selected day after that execution is a week old. Make sure not to manually execute the query during this time.

If a query execution fails, Databricks SQL retries with a back-off algorithm. The more failures the further away the next retry will be (and it might be beyond the refresh interval).

Refer documentation for additional info,

<https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/schedule-query>

## Question 38

In order to use Unity catalog features, which of the following steps needs to be taken on managed/external tables in the Databricks workspace?

* **Enable unity catalog feature in workspace settings**
* **Migrate/upgrade objects in workspace managed/external tables/view to unity catalog**
* **Upgrade to DBR version 15.0**
* **Copy data from workspace to unity catalog**
* **Upgrade workspace to Unity catalog**

**Explanation**

[Upgrade tables and views to Unity Catalog - Azure Databricks | Microsoft Docs](https://docs.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/migrate)

Managed table:[Upgrade a managed to Unity Catalog](https://docs.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/migrate#--upgrade-a-table-to-unity-catalog)

External table: [Upgrade an external table to Unity Catalog](https://docs.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/migrate#--upgrade-an-external-table-to-unity-catalog)

Bottom of Form

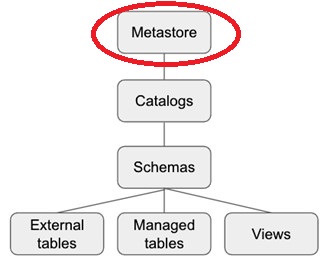
## Question 39

What is the top-level object in unity catalog?

* **Catalog**
* **Table**
* **Workspace**
* **Database**
* **Metastore**

**Explanation**

[Key concepts - Azure Databricks | Microsoft Docs](https://docs.microsoft.com/en-us/azure/databricks/data-governance/unity-catalog/key-concepts)



## Question 40

One of the team members Steve who has the ability to create views, created a new view called regional\_sales\_vw on the existing table called sales which is owned by John, and the second team member Kevin who works with regional sales managers wanted to query the data in regional\_sales\_vw, so Steve granted the permission to Kevin using command

GRANT VIEW, USAGE ON regional\_sales\_vw to kevin@company.com but Kevin is still unable to access the view?

* **Kevin needs select access on the table sales**
* **Kevin needs owner access on the view regional\_sales\_vw**
* **Steve is not the owner of the sales table**
* **Kevin is not the owner of the sales table**
* **Table access control is not enabled on the table and view**

**Explanation**

Ownership determines whether or not you can grant privileges on derived objects to other users, since Steve is not the owner of the underlying sales table, he can not grant access to the table or data in the table indirectly.

Only owner(user or group) can grant access to a object

<https://docs.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges#a-user-has-select-privileges-on-a-view-of-table-t-but-when-that-user-tries-to-select-from-that-view-they-get-the-error-user-does-not-have-privilege-select-on-table>

[Data object privileges - Azure Databricks | Microsoft Doc](https://docs.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges#a-user-has-select-privileges-on-a-view-of-table-t-but-when-that-user-tries-to-select-from-that-view-they-get-the-error-user-does-not-have-privilege-select-on-table)

## Question 41

Kevin is the owner of the schema sales, Steve wanted to create new table in sales schema called regional\_sales so Kevin grants the create table permissions to Steve. Steve creates the new table called regional\_sales in sales schema, who is the owner of the table regional\_sales

* **Kevin is the owner of sales schema, all the tables in the schema will be owned by Kevin**
* **Steve is the owner of the table**
* **By default ownership is assigned DBO**
* **By default ownership is assigned to DEFAULT\_OWNER**
* **Kevin and Smith both are owners of table**

**Explanation**

A user who creates the object becomes its owner, does not matter who is the owner of the parent object.

## Question 42

You were asked to setup a new all-purpose cluster, but the cluster is unable to start which of the following steps do you need to take to identify the root cause of the issue and the reason why the cluster was unable to start?

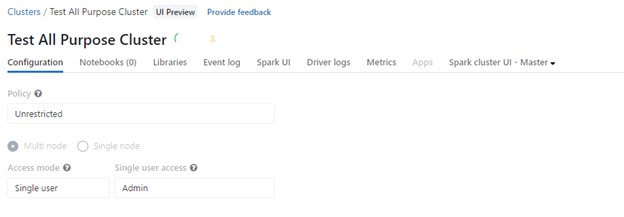
* **Check the cluster driver logs**
* **Check the cluster event logs**
* **Workspace logs**
* **Storage account**
* **Data plane**

**Explanation**

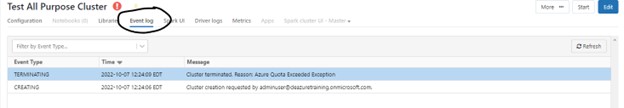
Cluster event logs are very useful, to identify issues pertaining to cluster availability.  Cluster may not start due to resource limitations or issues with the cloud providers.

Some of the common issues include a subnet for compute VM reaching its limits or exceeding the subscription or cloud account CPU quota limit.

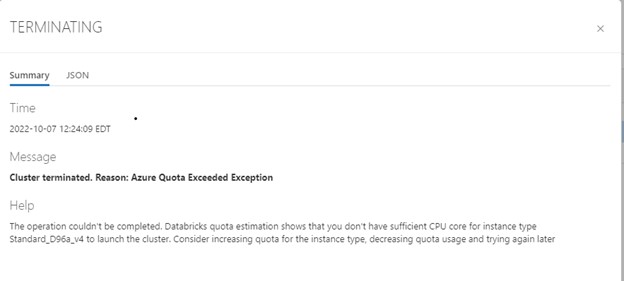
Here is an example where the cluster did not start due to subscription reaching the quota limit on a certain type of cpu cores for a VM type.



Click on event logs



Click on the message to see the detailed error message on why the cluster did not start.



## Question 43

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

* **Delete branch**
* **Trigger Databricks CICD pipeline**
* **Commit and push code**
* **Create a pull request**
* **Approve the pull request**

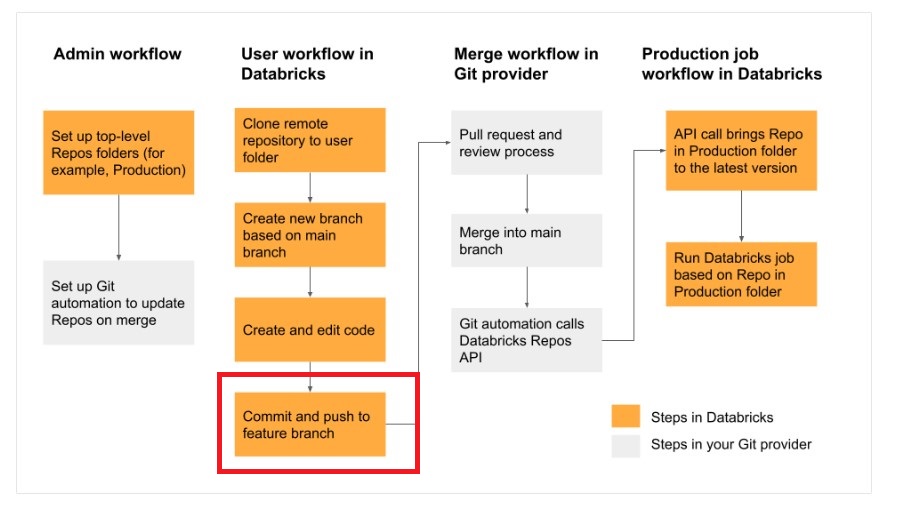
**Explanation**

The answer is Commit and push code.

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.

***Exam focus: Please study the below image carefully to understand all of the steps in the CI/CD flow to understand the tasks that are implemented in Databricks Repo vs Git Provider, exam may ask a different type of questions based on this flow.***



## Question 44

You noticed that a team member started using an all-purpose cluster to develop a notebook and used the same all-purpose cluster to set up a job that can run every 30 mins so they can update underlying tables which are used in a dashboard. What would you recommend for reducing the overall cost of this approach?

* **Reduce the size of the cluster**
* **Reduce the number of nodes and enable auto scale**
* **Enable auto termination after 30 mins**
* **Change the cluster all-purpose to job cluster when scheduling the job**
* **Change the cluster mode from all-purpose to single-mode**

**Explanation**

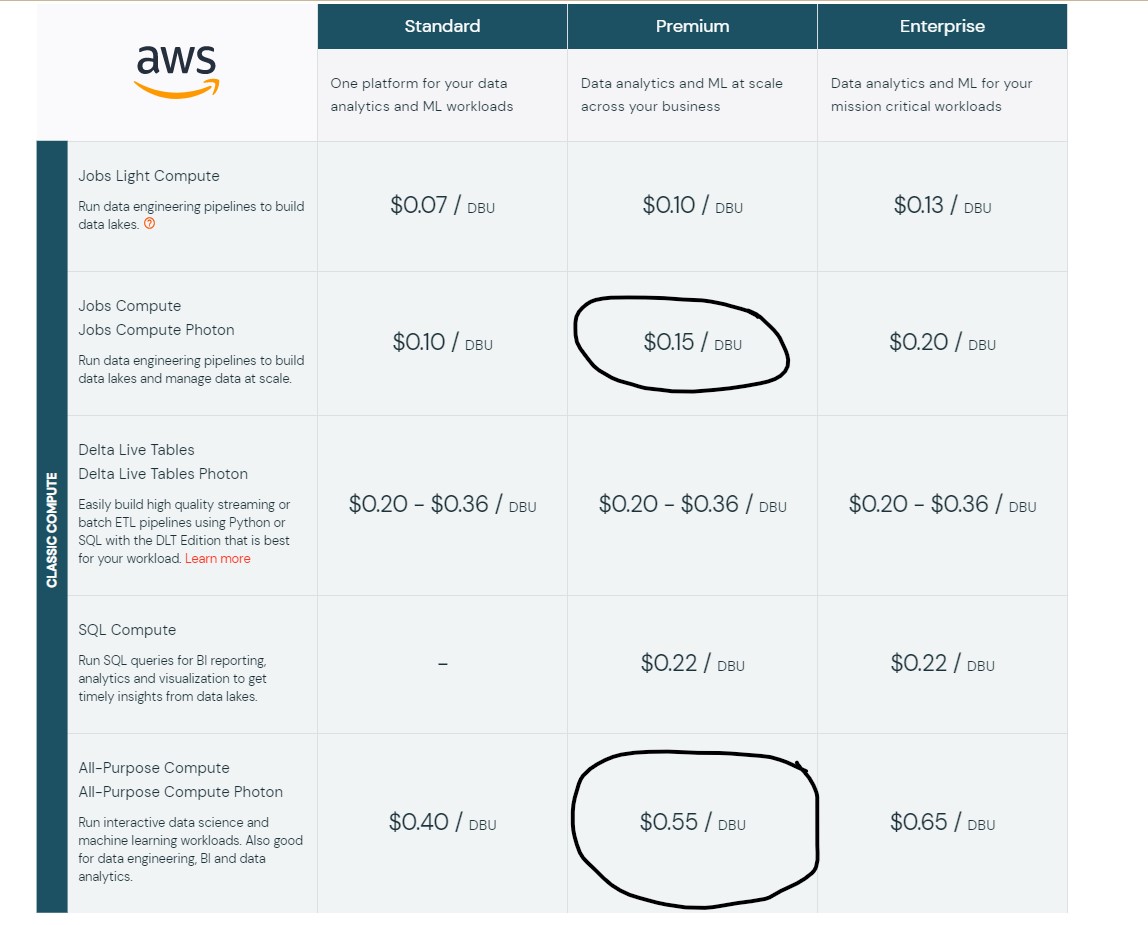
While using an all-purpose cluster is ok during development but anytime you don't need to interact with a notebook, especially for a scheduled job it is less expensive to use a job cluster. Using an all-purpose cluster can be twice as expensive as a job cluster.

Please note: The compute cost you pay the cloud provider for the same cluster type and size between an all-purpose cluster and job cluster is the same the only difference is the DBU cost.

The total cost of cluster = Total cost of VM compute(Azure or AWS or GCP) + Cost per DBU

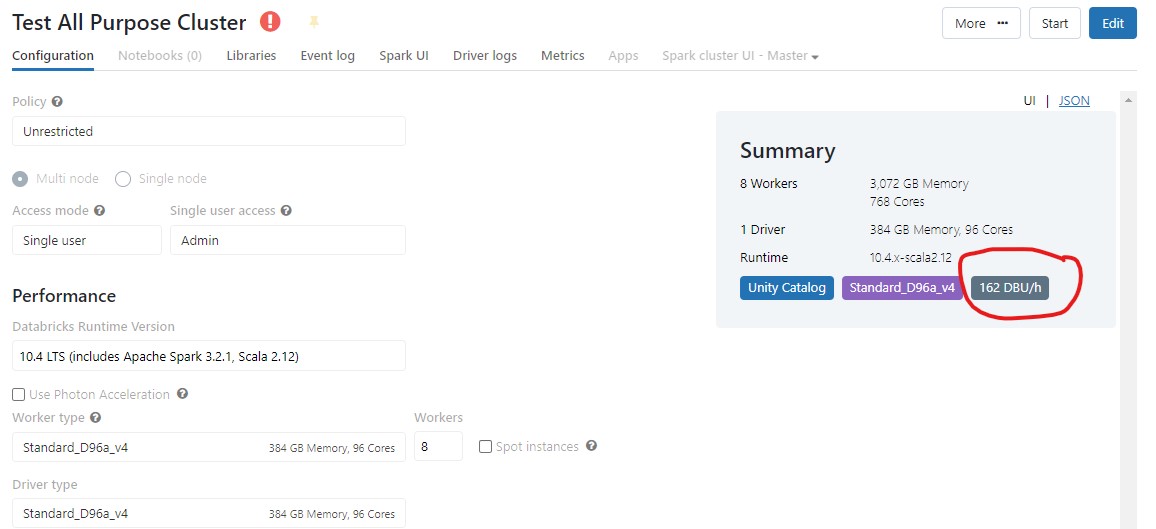
The per DBU cost varies between all-purpose and Job Cluster

Here is the recent cost estimate from AWS between Jobs Cluster and all-purpose Cluster, for jobs compute its $0.15 cents per DBU v$0.55 cents per DBU for all-purpose



How do I check how much the DBU cost for my cluster?

When you click on an exister cluster or when you look at the cluster details you will see this in the top right corner



## Question 45

Which of the following commands can be used to run one notebook from another notebook?

* **notebook.utils.run("full notebook path")**
* **execute.utils.run("full notebook path")**
* **dbutils.notebook.run("full notebook path")**
* **only job clusters can run notebook**
* **spark.notebook.run("full notebook path")**

**Explanation**

The answer is dbutils.notebook.run(" full notebook path ")

Here is the full command with additional options.

**run(path: String, timeout\_seconds: int, arguments: Map): String**

1. dbutils.notebook.run("ful-notebook-name", 60, {"argument": "data", "argument2": "data2", ...})

Bottom of Form