Contents

[Question 1 2](#_Toc155619313)

[Question 2 5](#_Toc155619314)

[Question 3 6](#_Toc155619315)

[Question 4 7](#_Toc155619316)

[Question 5 8](#_Toc155619317)

[Question 7 10](#_Toc155619318)

[Question 8 11](#_Toc155619319)

[Question 9 12](#_Toc155619320)

[Question 10 13](#_Toc155619321)

[Question 11 14](#_Toc155619322)

[Question 12 15](#_Toc155619323)

[Question 13 16](#_Toc155619324)

[Question 14 17](#_Toc155619325)

[Question 15 18](#_Toc155619326)

[Question 16 20](#_Toc155619327)

[Question 17 21](#_Toc155619328)

[Question 18 23](#_Toc155619329)

[Question 19 24](#_Toc155619330)

[Question 20 26](#_Toc155619331)

[Question 21 27](#_Toc155619332)

[Question 22 28](#_Toc155619333)

[Question 24 30](#_Toc155619334)

[Question 25 31](#_Toc155619335)

[Question 26 32](#_Toc155619336)

[Question 27 33](#_Toc155619337)

[Question 28 34](#_Toc155619338)

[Question 29 36](#_Toc155619339)

[Question 31 39](#_Toc155619340)

[Question 32 40](#_Toc155619341)

[Question 33 42](#_Toc155619342)

[Question 34 44](#_Toc155619343)

[Question 35 46](#_Toc155619344)

[Question 36 48](#_Toc155619345)

[Question 37 50](#_Toc155619346)

[Question 38 51](#_Toc155619347)

[Question 39 52](#_Toc155619348)

[Question 40 53](#_Toc155619349)

[Question 41 55](#_Toc155619350)

[Question 42 57](#_Toc155619351)

[Question 43 58](#_Toc155619352)

[Question 44 59](#_Toc155619353)

[Question 45 60](#_Toc155619354)

## Question 1

How does Lakehouse replace the dependency on using Data lakes and Data warehouses in a Data and Analytics solution?

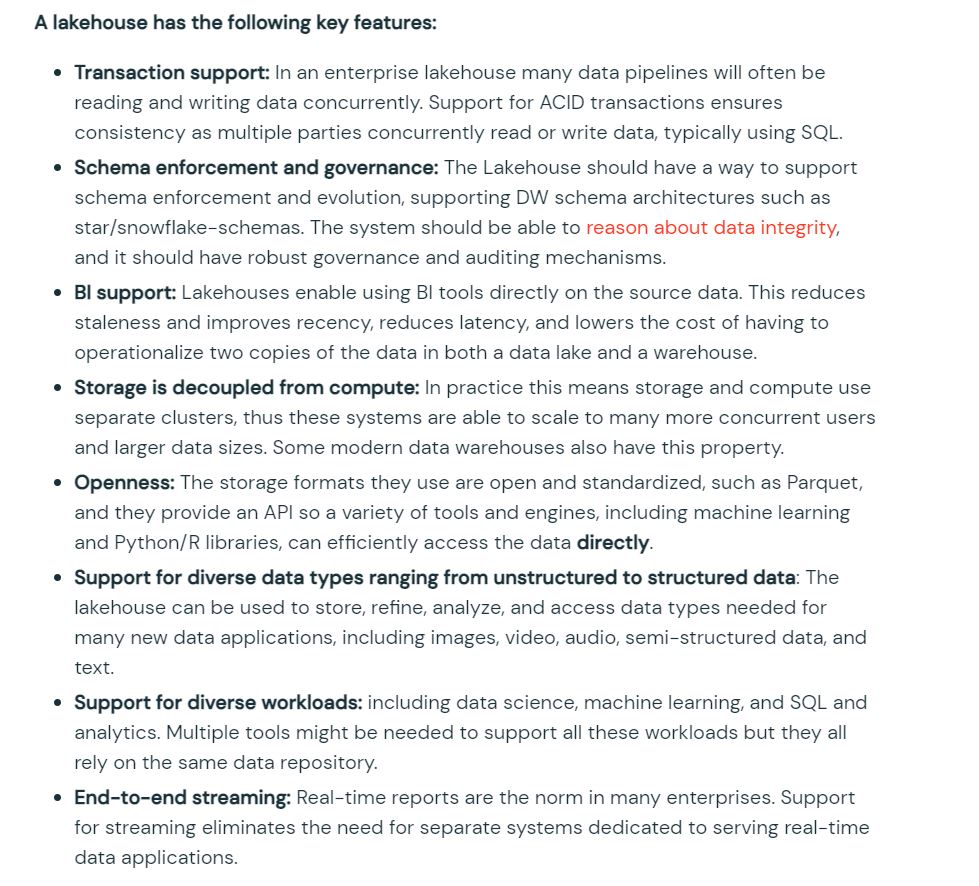
* **Open, direct access to data stored in standard data formats.**
* **Supports ACID transactions.**
* **Supports BI and Machine learning workloads**
* **Support for end-to-end streaming and batch workloads**
* **All the above**

**Explanation**

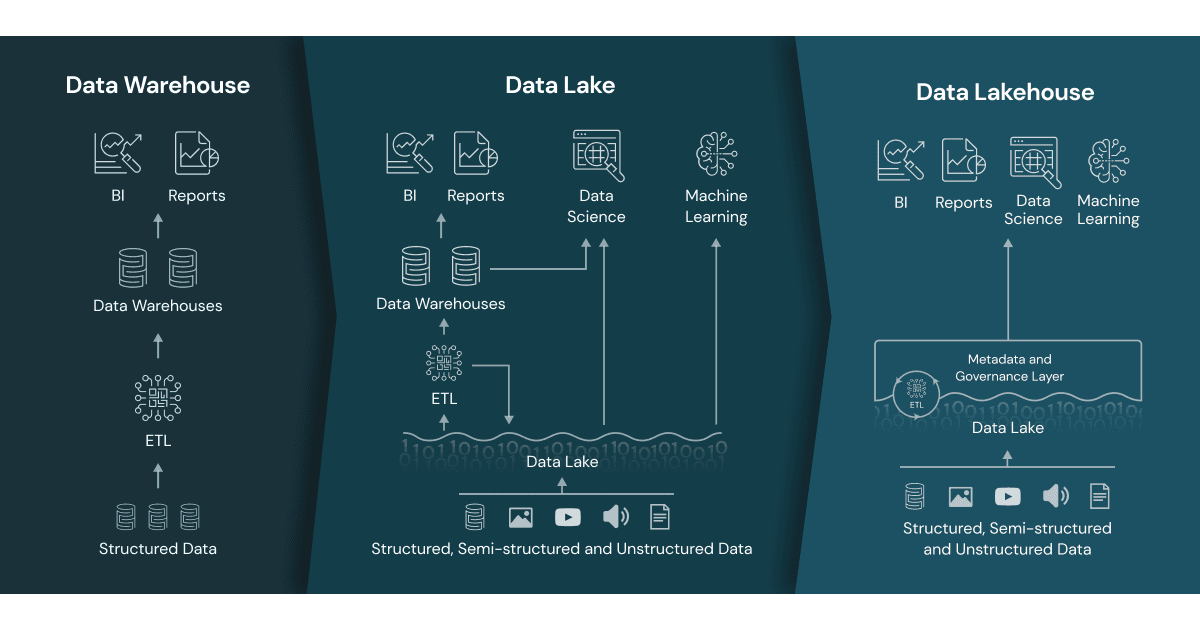
Lakehouse combines the benefits of a data warehouse and data lakes,

**Lakehouse = Data Lake + DataWarehouse**

Here are some of the major benefits of a lakehouse



**Lakehouse = Data Lake + DataWarehouse**



## Question 2

You are currently working on storing data you received from different customer surveys, this data is highly unstructured and changes over time,  why Lakehouse is a better choice compared to a Data warehouse?

* **Lakehouse supports schema enforcement and evolution, traditional data warehouses lack schema evolution.**
* **Lakehouse supports SQL**
* **Lakehouse supports ACID**
* **Lakehouse enforces data integrity**
* **Lakehouse supports primary and foreign keys like a data warehouse**

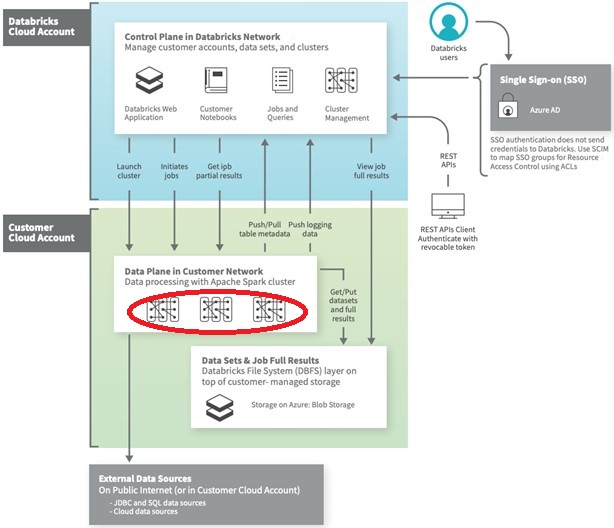
## Question 3

Which of the following locations hosts the driver and worker nodes of a Databricks-managed cluster?

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

**Explanation**

The answer is Data Plane, which is where compute(all-purpose, Job Cluster, DLT) are stored this is generally a customer cloud account,  there is one exception SQL Warehouses, currently there are 3 types of SQL Warehouse compute available(classic, pro, serverless), in classic and pro compute is located in customer cloud account but serverless computed is located in Databricks cloud account.



## Question 4

You have written a notebook to generate a summary data set for reporting, Notebook was scheduled using the job cluster, but you realized it takes an average of 8 minutes to start the cluster, what feature can be used to start the cluster in a timely fashion?

* **Setup an additional job to run ahead of the actual job so the cluster is running second job starts**
* **Use the Databricks cluster pools feature to reduce the startup time**
* **Use Databricks Premium edition instead of Databricks standard edition**
* **Pin the cluster in the cluster UI page so it is always available to the jobs**
* **Disable auto termination so the cluster is always running**

**Explanation**

Cluster pools allow us to reserve VM's ahead of time, when a new job cluster is created VM are grabbed from the pool. Note: when the VM's are waiting to be used by the cluster only cost incurred is Azure. Databricks run time cost is only billed once VM is allocated to a cluster.

Here is a demo of how to setup and follow some best practices,

<https://www.youtube.com/watch?v=FVtITxOabxg&ab_channel=DatabricksAcademy>

## Question 5

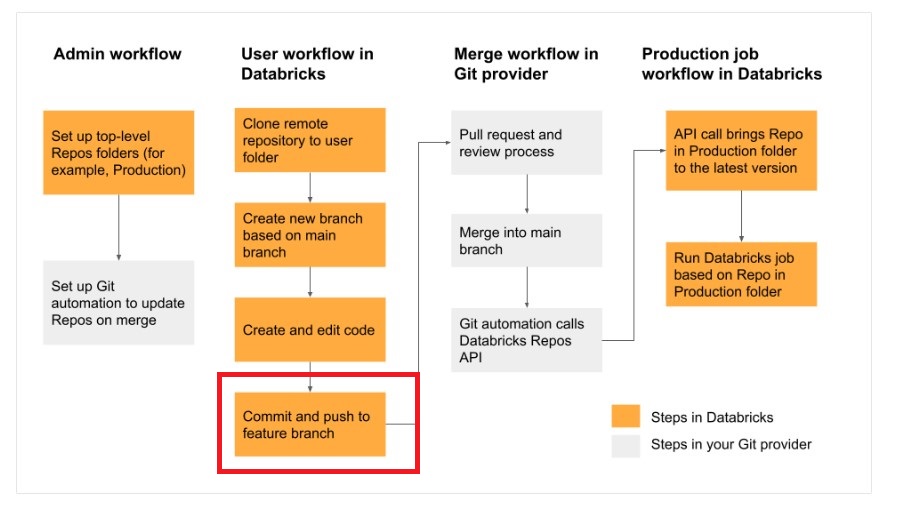
Which of the following statement is true about Databricks repos?

* **You can approve the pull request if you are the owner of Databricks repos**
* **A workspace can only have one instance of git integration**
* **Databricks Repos and Notebook versioning are the same features**
* **You cannot create a new branch in Databricks repos**
* **Databricks repos allow you to comment and commit code changes and push them to a remote branch**

**Explanation**

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

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



Bottom of Form

Top of Form

Question 6: **Correct**

Which of the statement is correct about the cluster pools?

* **Cluster pools allow you to perform load balancing**
* **Cluster pools allow you to create a cluster**
* **Cluster pools allow you to save time when starting a new cluster**
* **Cluster pools are used to share resources among multiple teams**
* **Cluster pools allow you to have all the nodes in the cluster from single physical server rack**

## Question 7

Once a cluster is deleted, below additional actions need to performed by the administrator

* **Remove virtual machines but storage and networking are automatically dropped**
* **Drop storage disks but Virtual machines and networking are automatically dropped**
* **Remove networking but Virtual machines and storage disks are automatically dropped**
* **Remove logs**
* **No action needs to be performed. All resources are automatically removed.**

## Question 8

How does a Delta Lake differ from a traditional data lake?

* **Delta lake is Datawarehouse service on top of data lake that can provide reliability, security, and performance**
* **Delta lake is a caching layer on top of data lake that can provide reliability, security, and performance**
* **Delta lake is an open storage format like parquet with additional capabilities that can provide reliability, security, and performance**
* **Delta lake is an open storage format designed to replace flat files with additional capabilities that can provide reliability, security, and performance**
* **Delta lake is proprietary software designed by Databricks that can provide reliability, security, and performance**

**Explanation**

What is Delta?

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 9

How VACCUM and OPTIMIZE commands can be used to manage the DELTA lake?

* **VACCUM command can be used to compact small parquet files, and the OPTIMZE command can be used to delete parquet files that are marked for deletion/unused.**
* **VACCUM command can be used to delete empty/blank parquet files in a delta table. OPTIMIZE command can be used to update stale statistics on a delta table.**
* **VACCUM command can be used to compress the parquet files to reduce the size of the table, OPTIMIZE command can be used to cache frequently delta tables for better performance.**
* **VACCUM command can be used to delete empty/blank parquet files in a delta table, OPTIMIZE command can be used to cache frequently delta tables for better performance.**
* **OPTIMIZE command can be used to compact small parquet files, and the VACCUM command can be used to delete parquet files that are marked for deletion/unused.**

**Explanation**

**VACCUM:**

You can remove files no longer referenced by a Delta table and are older than the retention threshold by running the vacuum command on the table. vacuum is not triggered automatically. The default retention threshold for the files is 7 days. To change this behavior, see [Configure data retention for time travel](https://docs.databricks.com/delta/history.html#data-retention).

**OPTIMIZE:**

Using OPTIMIZE you can compact data files on Delta Lake, this can improve the speed of read queries on the table. Too many small files can significantly degrade the performance of the query.

## Question 10

Which of the below commands can be used to drop a DELTA table?

* **DROP DELTA table\_name**
* **DROP TABLE table\_name**
* **DROP TABLE table\_name FORMAT DELTA**
* **DROP  table\_name**

## Question 11

Delete records from the transactions Delta table where transactionDate is greater than current timestamp?

* **DELETE FROM transactions FORMAT DELTA where transactionDate > currenct\_timestmap()**
* **DELETE FROM transactions if transctionDate > current\_timestamp()**
* **DELETE FROM transactions where transactionDate > current\_timestamp()**
* **DELETE FROM transactions where transactionDate > current\_timestamp() KEEP\_HISTORY**
* **DELET FROM transactions where transactionDate GE current\_timestamp()**

## Question 12

Identify one of the below statements that can query a delta table in PySpark Dataframe API

* **Spark.read.mode("delta").table("table\_name")**
* **Spark.read.table.delta("table\_name")**
* **Spark.read.table("table\_name")**
* **Spark.read.format("delta").LoadTableAs("table\_name")**
* **Spark.read.format("delta").TableAs("table\_name")**

## Question 13

The default threshold of VACUUM is 7 days, internal audit team asked to certain tables to maintain at least 365 days as part of compliance requirement, which of the below setting is needed to implement.

* **ALTER TABLE table\_name set TBLPROPERTIES (delta.deletedFileRetentionDuration= ‘interval 365 days’)**
* **MODIFY TABLE table\_name set TBLPROPERTY (delta.maxRetentionDays = ‘interval 365 days’)**
* **ALTER TABLE table\_name set EXENDED TBLPROPERTIES (delta.deletedFileRetentionDuration= ‘interval 365 days’)**
* **ALTER TABLE table\_name set EXENDED TBLPROPERTIES (delta.vaccum.duration= ‘interval 365 days’)**

**Explanation**

1. ALTER TABLE table\_name SET TBLPROPERTIES ( property\_key [ = ] property\_val [, ...] )

TBLPROPERTIES allow you to set key-value pairs

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

## Question 14

Which of the following commands can be used to query a delta table?

%python

spark.sql("select \* from table\_name")

%sql

Select \* from table\_name

**Both A & B**

%python

execute.sql("select \* from table")

%python

delta.sql("select \* from table")

**Explanation**

The answer is both options A and B

Options C and D are incorrect because there is no command in Spark called execute.sql or delta.sql

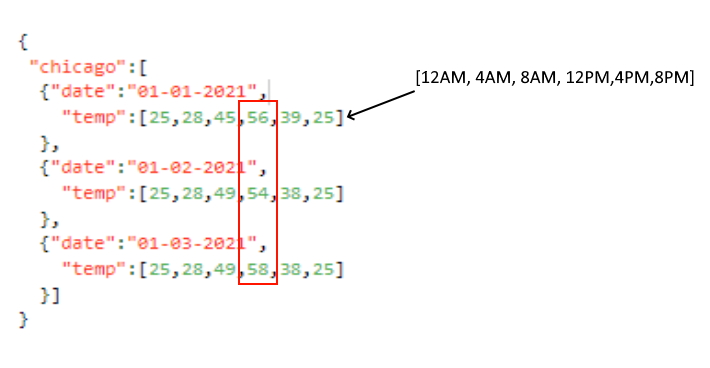
## Question 15

Below table **temp\_data**has one column called **raw** contains JSON data that records temperature for every four hours in the day for the city of **Chicago**, you are asked to calculate the **maximum**temperature that was ever recorded for **12:00 PM** hour across all the days.  Parse the JSON data and use the necessary array function to calculate the max temp.

Table: temp\_date

Column: raw

Datatype: string

****

**Expected output: 58**

* + select max(raw.chicago.temp[3]) from temp\_data
  + select array\_max(raw.chicago[\*].temp[3]) from temp\_data
  + select array\_max(from\_json(raw['chicago'].temp[3],'array<int>')) from temp\_data
  + select array\_max(from\_json(raw:chicago[\*].temp[3],'array<int>')) from temp\_data
* select max(from\_json(raw:chicago[3].temp[3],'array<int>')) from temp\_data

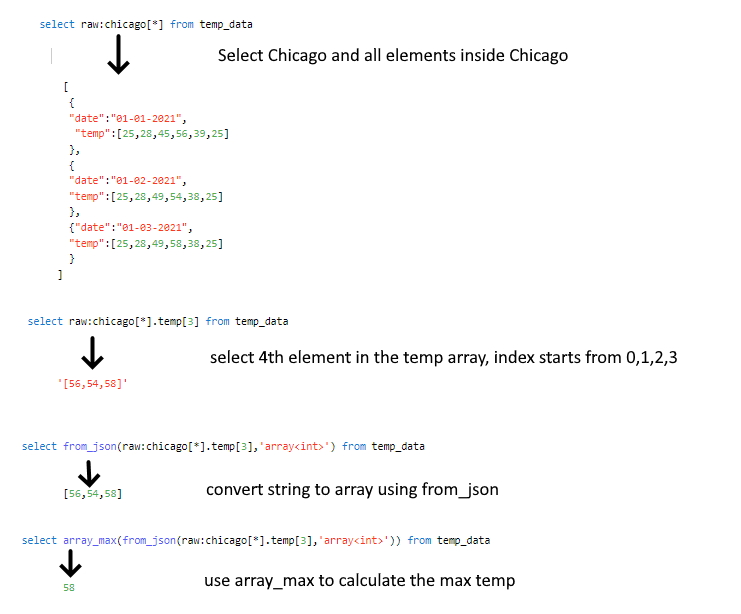
**Explanation**

Note: This is a difficult question, more likely you may see easier questions similar to this but the more you are prepared for the exam easier it is to pass the exam.

Use this below link to look for more examples, this will definitely help you,

<https://docs.databricks.com/optimizations/semi-structured.html>

Here is the solution, step by step



Use this below link to look for more examples, this will definitely help you,

<https://docs.databricks.com/optimizations/semi-structured.html>

If you want to try this solution use below DDL,

create or replace table temp\_data

as select ' {

"chicago":[

{"date":"01-01-2021",

"temp":[25,28,45,56,39,25]

},

{"date":"01-02-2021",

"temp":[25,28,49,54,38,25]

},

{"date":"01-03-2021",

"temp":[25,28,49,58,38,25]

}]

}

' as raw

select array\_max(from\_json(raw:chicago[\*].temp[3],'array<int>')) from temp\_data

## Question 16

Which of the following SQL statements can be used to update a transactions table, to set a flag on the table from Y to N

* **MODIFY transactions SET active\_flag = 'N' WHERE active\_flag = 'Y'**
* **MERGE transactions SET active\_flag = 'N' WHERE active\_flag = 'Y'**
* **UPDATE transactions SET active\_flag = 'N' WHERE active\_flag = 'Y'**
* **REPLACE transactions SET active\_flag = 'N' WHERE active\_flag = 'Y'**

**Explanation**

The answer is

UPDATE transactions SET active\_flag = 'N' WHERE active\_flag = 'Y'

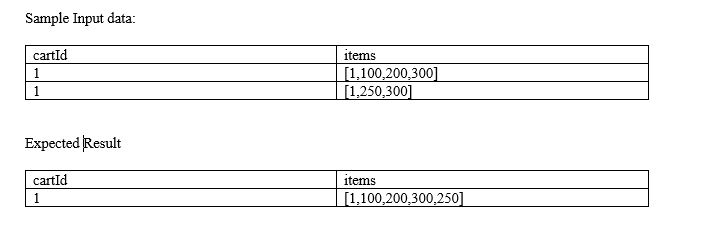
Delta Lake supports UPDATE statements on the delta table, all of the changes as part of the update are ACID compliant.

## Question 17

Below sample input data contains two columns, one cartId also known as session id, and the second column is called items, every time a customer makes a change to the cart this is stored as an array in the table, the Marketing team asked you to create a unique list of item’s that were ever added to the cart by each customer, fill in blanks by choosing the appropriate array function so the query produces below **expected** result as shown below.

Schema: cartId INT, items Array<INT>

Sample Data

****

1. SELECT cartId, \_\_\_ (\_\_\_(items)) as items
2. FROM carts GROUP BY cartId

**Expected result:**

cartId              items

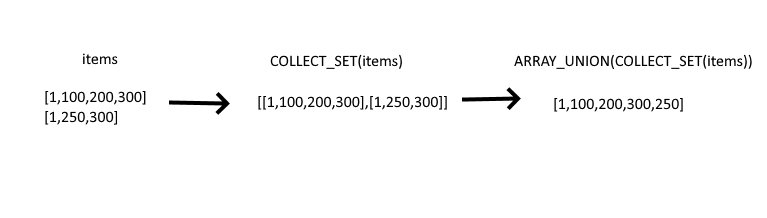
1                 [1,100,200,300,250]

* **FLATTEN, COLLECT\_UNION**
* **ARRAY\_UNION, FLATTEN**
* **ARRAY\_UNION, ARRAY\_DISTINT**
* **ARRAY\_UNION, COLLECT\_SET**
* **ARRAY\_DISTINCT, ARRAY\_UNION**

**Explanation**

COLLECT SET is a kind of aggregate function that combines a column value from all rows into a unique list

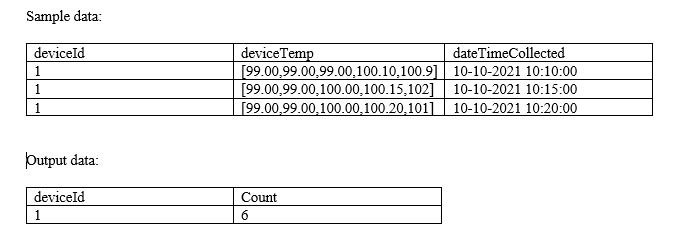
ARRAY\_UNION combines and removes any duplicates,



## Question 18

You were asked to identify number of times a temperature sensor exceed threshold temperature (100.00) by each device, each row contains 5 readings collected every 5 minutes, fill in the blank with the appropriate functions.

Schema: deviceId INT, deviceTemp ARRAY<double>, dateTimeCollected TIMESTAMP

****

SELECT deviceId, \_\_ (\_\_ (\_\_(deviceTemp], i -> i > 100.00)))

FROM devices

GROUP BY deviceId

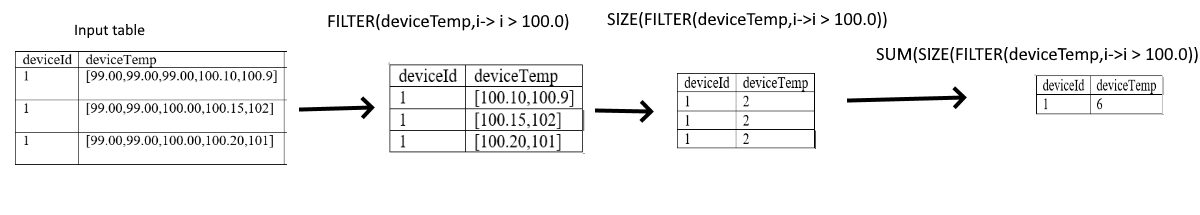
* **SUM, COUNT, SIZE**
* **SUM, SIZE, SLICE**
* **SUM, SIZE, ARRAY\_CONTAINS**
* **SUM, SIZE, ARRAY\_FILTER**
* **SUM, SIZE, FILTER**

**Explanation**

FILER function can be used to filter an array based on an expression

SIZE function can be used to get size of an array

SUM is used to calculate to total by device

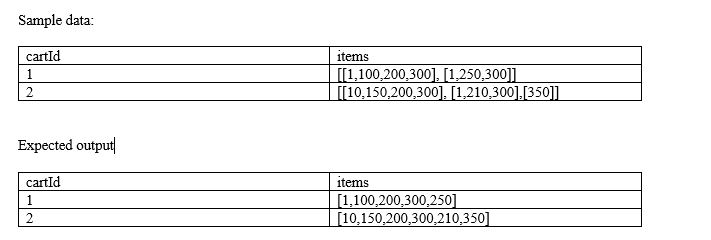


## Question 19

You are currently looking at a table that contains data from an e-commerce platform, each row contains a list of items(Item number) that were present in the cart, when the customer makes a change to the cart the entire information is saved as a separate list and appended to an existing list for the duration of the customer session, to identify all the items customer bought you have to make a unique list of items, you were asked to create a unique item’s list that was added to the cart by the user, **fill in the blanks** of below query by choosing the appropriate higher-order function?

Note: See below sample data and expected output.

Schema: cartId INT, items Array<INT>

****

**Fill in the blanks:**

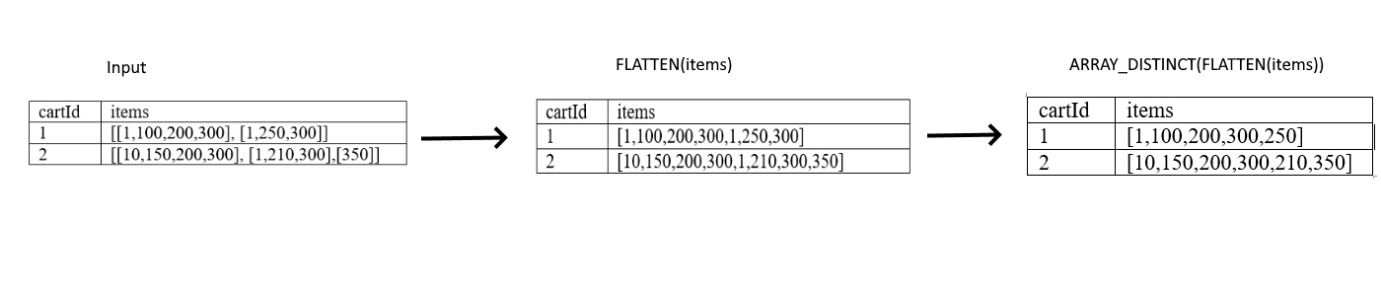
SELECT cartId, \_(\_(items)) FROM carts

* **ARRAY\_UNION, ARRAY\_DISCINT**
* **ARRAY\_DISTINCT, ARRAY\_UNION**
* **ARRAY\_DISTINCT, FLATTEN**
* **FLATTEN, ARRAY\_DISTINCT**
* **ARRAY\_DISTINCT, ARRAY\_FLATTEN**

**Explanation**

FLATTEN -> Transforms an array of arrays into a single array.

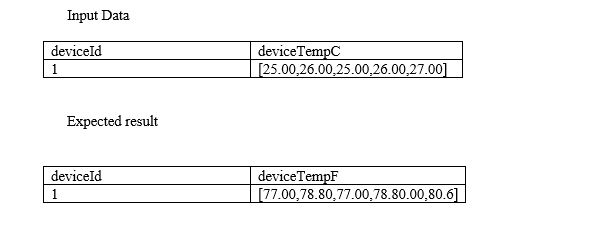
ARRAY\_DISTINCT -> The function returns an array of the same type as the input argument where all duplicate values have been removed.



## Question 20

You are working on IOT data where each device has 5 reading in an array collected in Celsius, you were asked to covert each individual reading from Celsius to Fahrenheit, fill in the blank with an appropriate function that can be used in this scenario.

Schema: deviceId INT, deviceTemp ARRAY<double>

****

SELECT deviceId, \_\_(deviceTempC,i-> (i \* 9/5) + 32) as deviceTempF

FROM sensors

* **APPLY**
* **MULTIPLY**
* **ARRAYEXPR**
* **TRANSFORM**
* **FORALL**

**Explanation**

TRANSFORM -> Transforms elements in an array in expr using the function func.

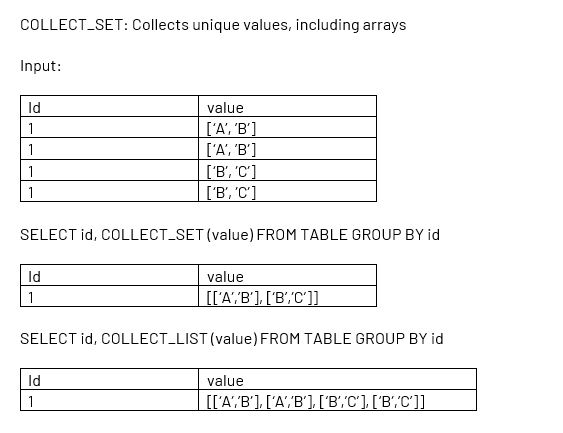
1. transform(expr, func)

## Question 21

Which of the following array functions takes input column return unique list of values in an array?

* + **COLLECT\_LIST**
  + **COLLECT\_SET**
  + **COLLECT\_UNION**
  + **ARRAY\_INTERSECT**
  + **ARRAY\_UNION**

**Explanation**



## Question 22

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

* **if department = “supply chain” | process:**
* **if department == “supply chain” or process = TRUE:**
* **if department == “supply chain” | process == TRUE:**
* **if department == “supply chain” | if process == TRUE:**
* **if department == “supply chain” or process:**

Bottom of Form

Top of Form

Question 23: **Correct**

What is the output of below function when executed with input parameters 1, 3 :

1. def check\_input(x,y):
2. if x < y:
3. x= x+1
4. if x>y:
5. x= x+1
6. if x <y:
7. x = x+1
8. return x

* **1**
* **2**
* **3**
* **4**
* **5**

**Explanation**

The answer is 2

## Question 24

Which of the following python statements can be used to replace the schema name and table name in the query?

table\_name = "sales"

schema\_name = "bronze"

query = f"select \* from schema\_name.table\_name"

table\_name = "sales"

query = "select \* from {schema\_name}.{table\_name}"

table\_name = "sales"

query = f"select \* from {schema\_name}.{table\_name}"

table\_name = "sales"

query = f"select \* from + schema\_name +"."+table\_name"

**Explanation**

The answer is

1. table\_name = "sales"
2. query = f"select \* from {schema\_name}.{table\_name}"

It is always best to use f strings to replace python variables, rather than using string concatenation.

## Question 25

you are currently working on creating a spark stream process to read and write in for a one-time micro batch, and also rewrite the existing target table, fill in the blanks to complete the below command successfully.

spark.table("source\_table")

.writeStream

.option("\_\_\_\_", “dbfs:/location/silver")

.outputMode("\_\_\_\_")

.trigger(Once=\_\_\_\_)

.table("target\_table")

* **checkpointlocation, complete, True**
* **targetlocation, overwrite, True**
* **checkpointlocation, True, overwrite**
* **checkpointlocation, True, complete**
* **checkpointlocation, overwrite, True**

## Question 26

You were asked to write python code to stop all running streams, which of the following command can be used to get a list of all active streams currently running so we can stop them, fill in the blank.

1. for s in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:
2. s.stop()

* **Spark.getActiveStreams()**
* **spark.streams.active**
* **activeStreams()**
* **getActiveStreams()**
* **spark.streams.getActive**

## Question 27

At the end of the inventory process a file gets uploaded to the cloud object storage, you are asked to build a process to ingest data which of the following method can be used to ingest the data incrementally, schema of the file is expected to change overtime ingestion process should be able to handle these changes automatically. Below is the auto loader to command to load the data, fill in the blanks for successful execution of below code.

spark.readStream

.format("cloudfiles")

.option("\_\_\_\_\_\_\_",”csv)

.option("\_\_\_\_\_\_\_", ‘dbfs:/location/checkpoint/’)

.load(data\_source)

.writeStream

.option("\_\_\_\_\_\_\_",’ dbfs:/location/checkpoint/’)

.option("\_\_\_\_\_\_\_", "true")

.table(table\_name))

* **format, checkpointlocation, schemalocation, overwrite**
* **cloudfiles.format, checkpointlocation, cloudfiles.schemalocation, overwrite**
* **cloudfiles.format, cloudfiles.schemalocation, checkpointlocation, mergeSchema**
* **cloudfiles.format, cloudfiles.schemalocation, checkpointlocation, append**
* **cloudfiles.format, cloudfiles.schemalocation, checkpointlocation, overwrite**

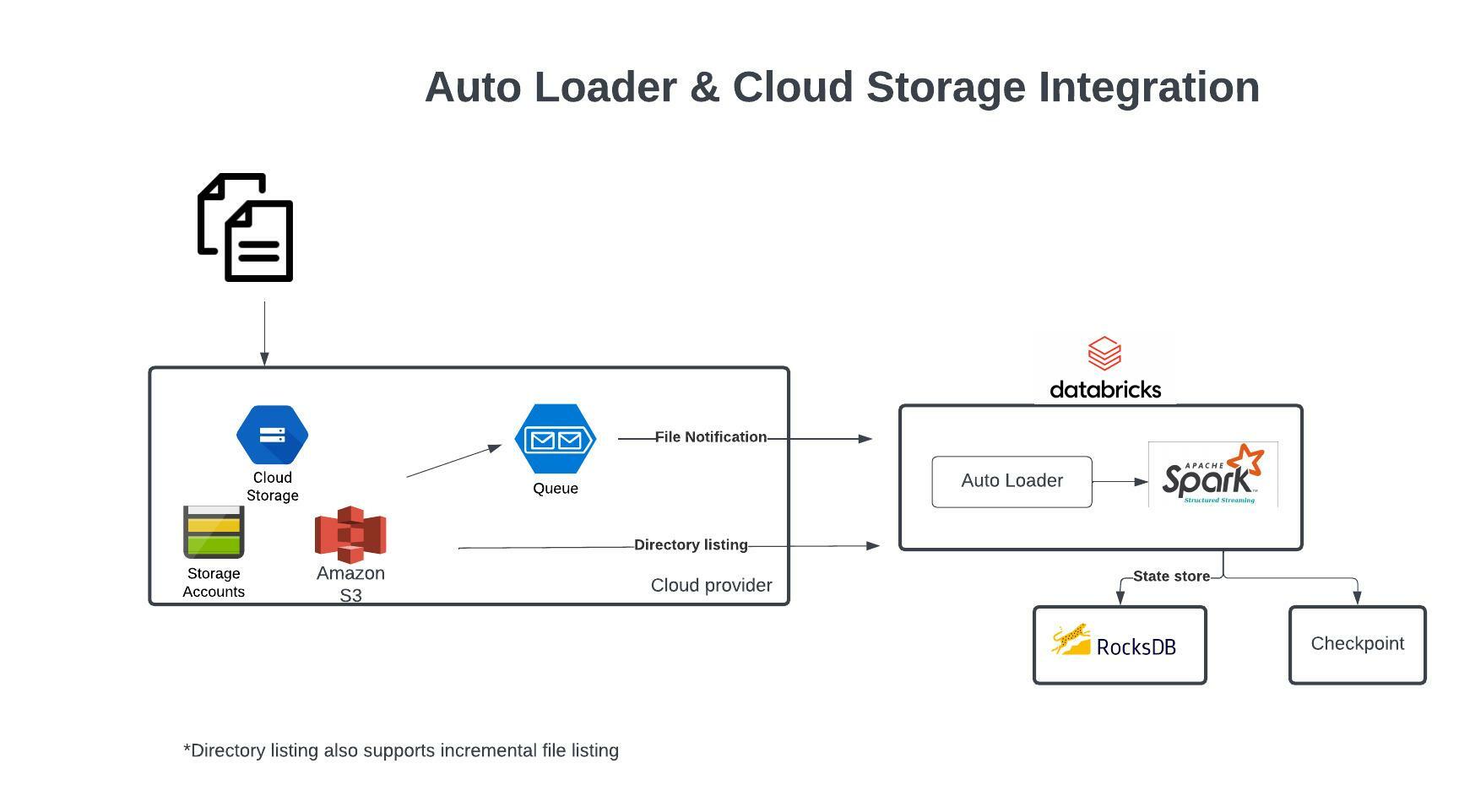
## Question 28

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

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

**Explanation**

The answer is, Efficiently process new data incrementally from cloud object storage, AUTO LOADER only supports ingesting files stored in a cloud object storage. Auto Loader cannot process streaming data sources like Kafka or Delta streams, use Structured streaming for these data sources.



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.

## Question 29

You are asked to setup an AUTO LOADER to process the incoming data, this data arrives in JSON format and get dropped into cloud object storage and you are required to process the data as soon as it arrives in cloud storage, which of the following statements is correct

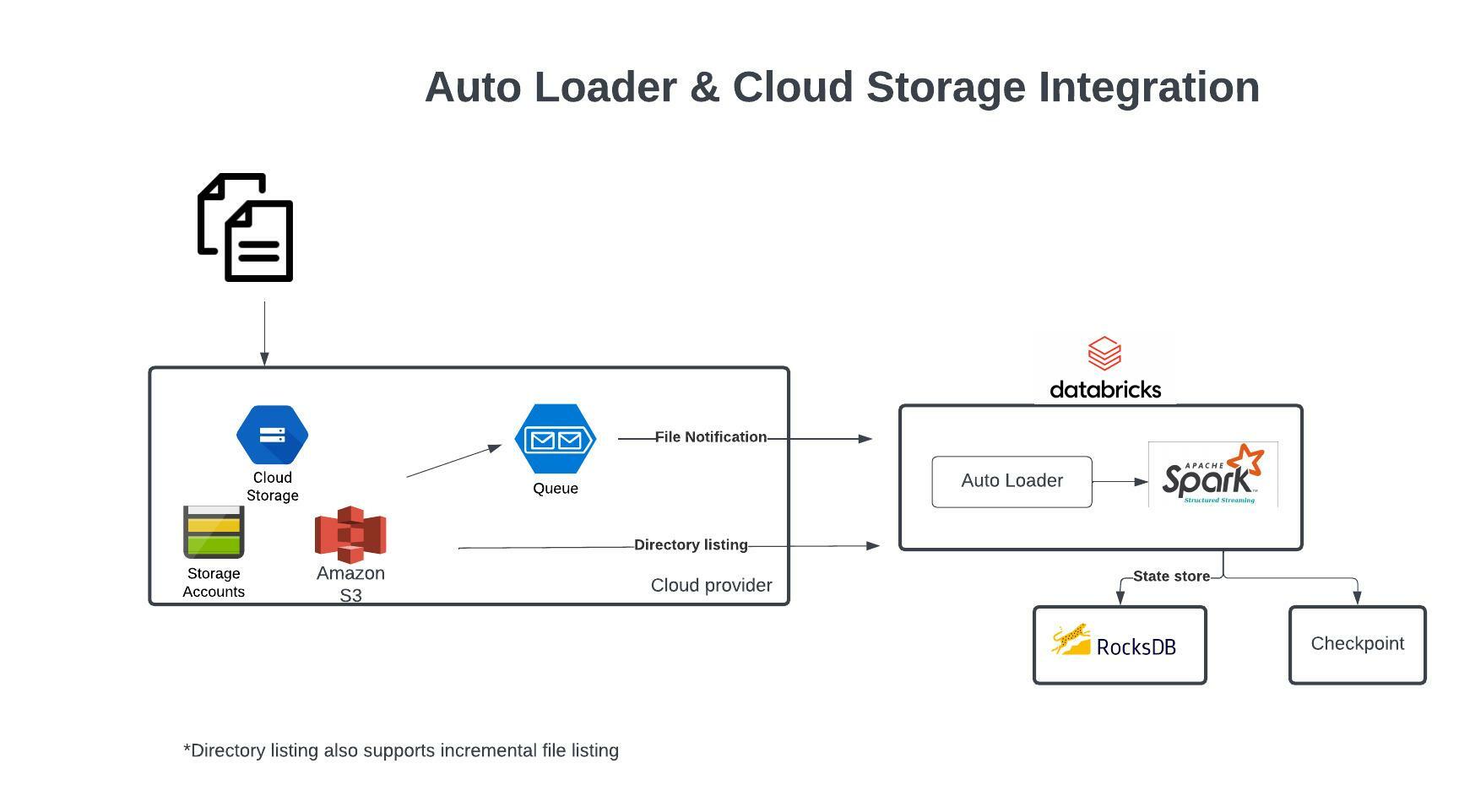
* **AUTO LOADER is native to DELTA lake it cannot support external cloud object storage**
* **AUTO LOADER has to be triggered from an external process when the file arrives in the cloud storage**
* **AUTO LOADER needs to be converted to a Structured stream process**
* **AUTO LOADER can only process continuous data when stored in DELTA lake**
* **AUTO LOADER can support file notification method so it can process data as it arrives**

**Explanation**

Auto Loader supports two modes when ingesting new files from cloud object storage

**Directory listing**: Auto Loader identifies new files by listing the input directory, and uses a directory polling approach.

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



File notification is more efficient and can be used to process the data in real-time as data arrives in cloud object storage.

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

Bottom of Form

Top of Form

Question 30

What is the main difference between the bronze layer and silver layer in a medallion architecture?

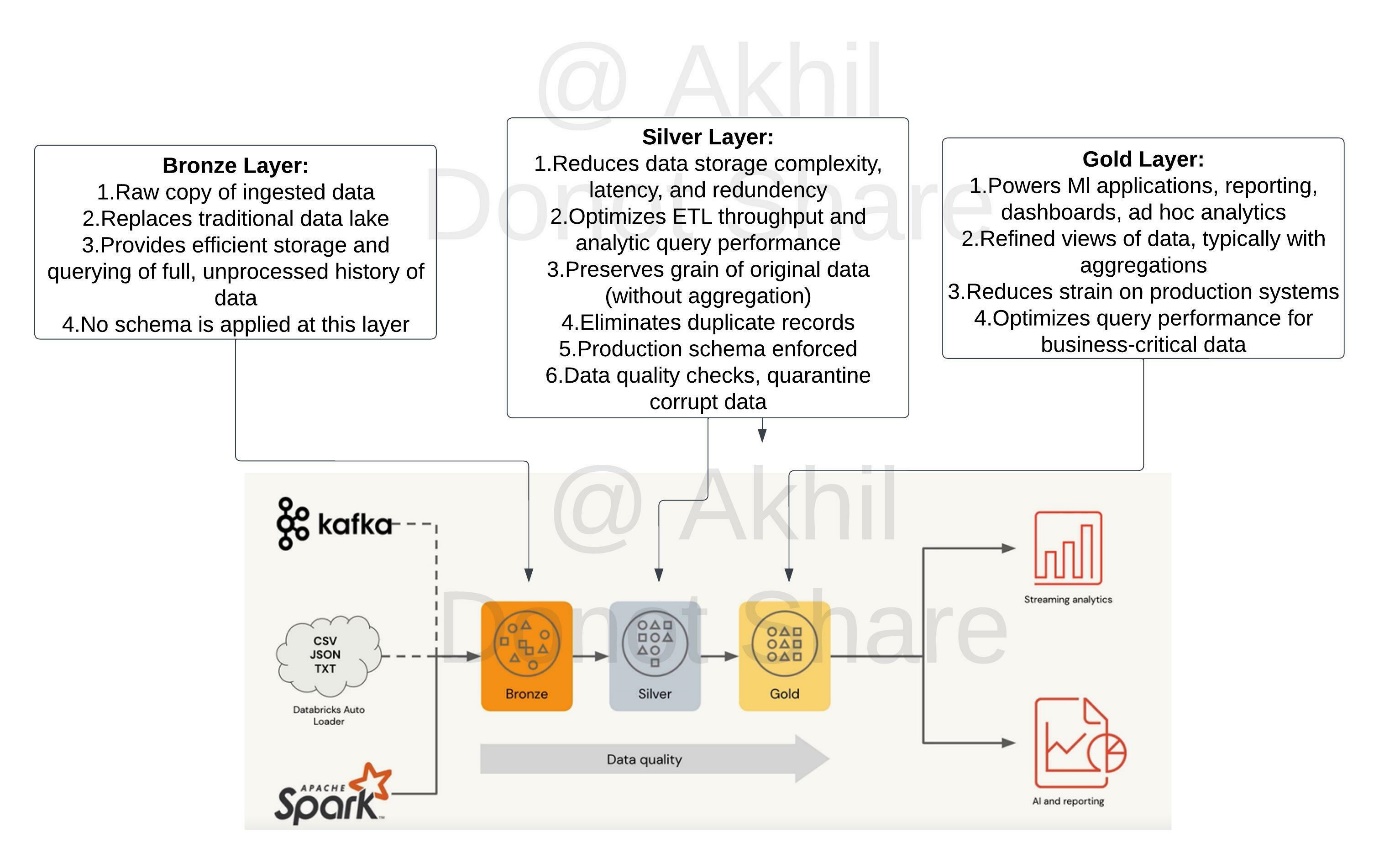
* **Duplicates are removed in bronze, schema is applied in silver**
* **Silver may contain aggregated data**
* **Bronze is raw copy of ingested data, silver contains data with production schema and optimized for ELT/ETL throughput**
* **Bad data is filtered in Bronze, silver is a copy of bronze data**

**Explanation**

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

***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 31

What is the main difference between the silver layer and the gold layer in medalion architecture?

**Silver may contain aggregated data**

**Gold may contain aggregated data**

**Data quality checks are applied in gold**

**Silver is a copy of bronze data**

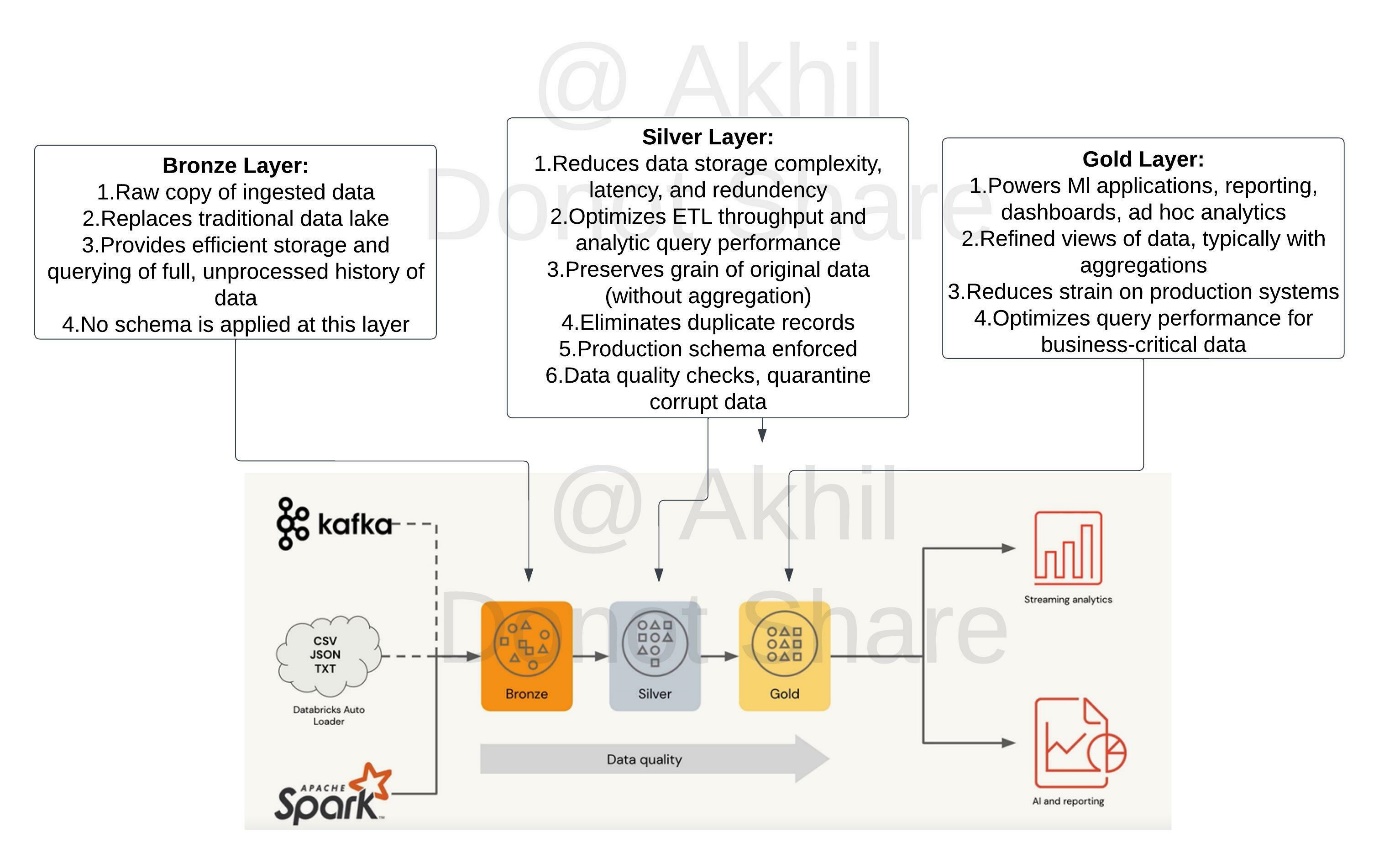
**God is a copy of silver data**

**Explanation**

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

***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 32

What is the main difference between the silver layer and gold layer in medallion architecture?

* **Silver optimized to perform ETL, Gold is optimized query performance**
* **Gold is optimized go perform ETL, Silver is optimized for query performance**
* **Silver is copy of Bronze, Gold is a copy of Silver**
* **Silver is stored in Delta Lake, Gold is stored in memory**
* **Silver may contain aggregated data, gold may preserve the granularity of original data**

**Explanation**

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

Gold Layer:

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

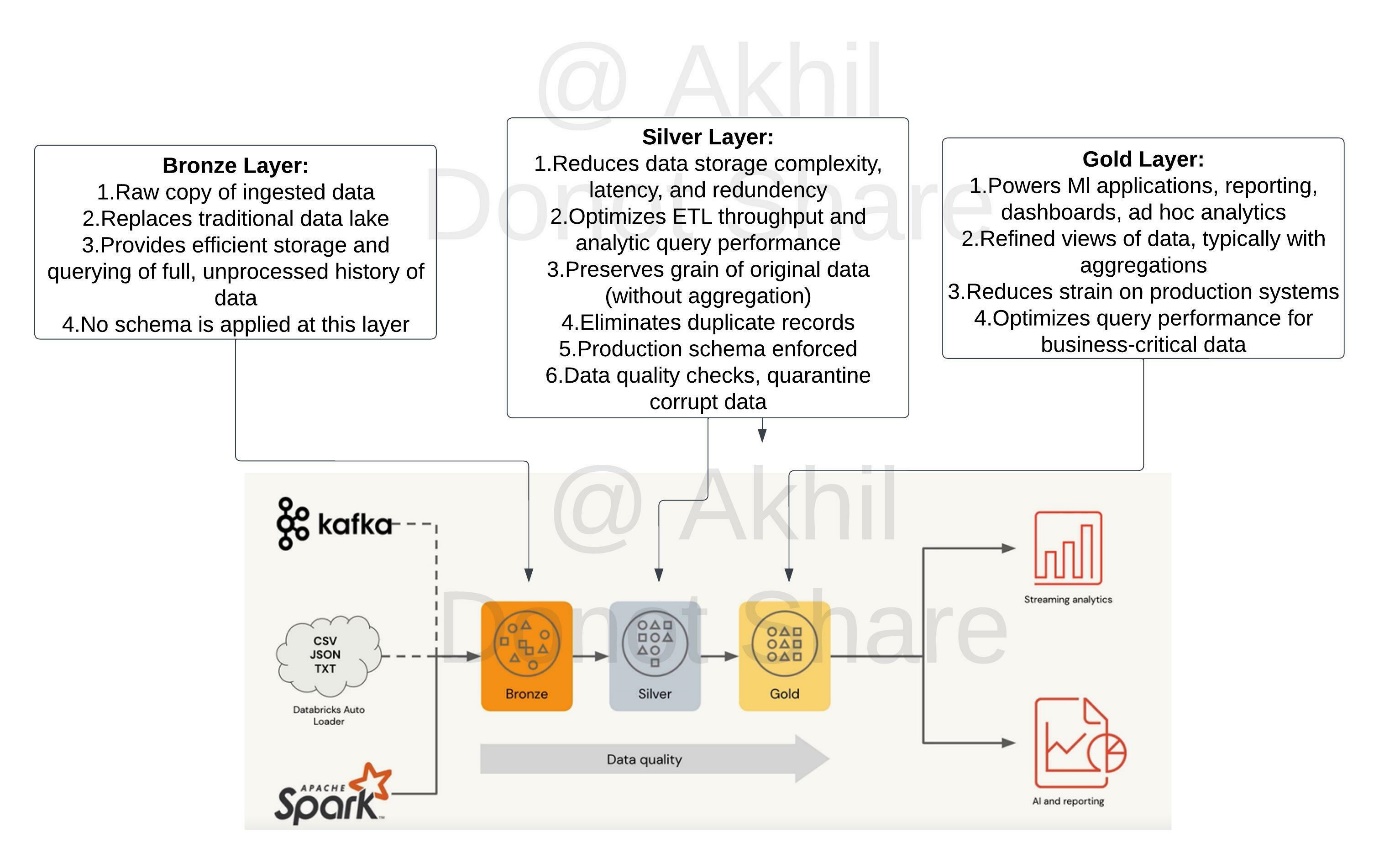
2. Refined views of data, typically with aggregations

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 33

A dataset has been defined using Delta Live Tables and includes an expectations clause: CONSTRAINT valid timestamp EXPECT (timestamp > '2020-01-01')

What is the expected behaviour when a batch of data containing data that violates these constraints is processed?

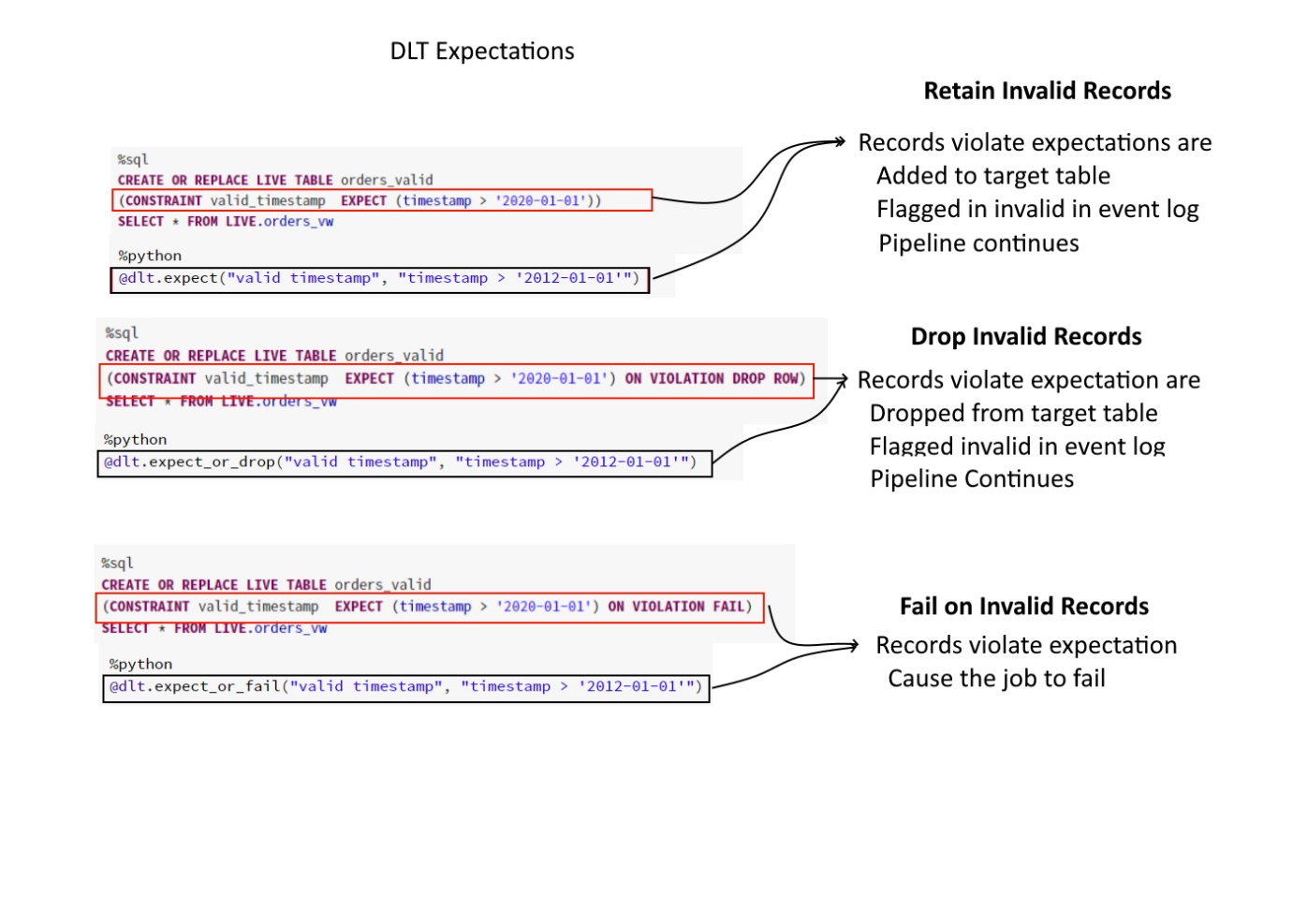
* **Records that violate the expectation are added to the target dataset and recorded as invalid in the event log.**
* **Records that violate the expectation are dropped from the target dataset and recorded as invalid in the event log.**
* **Records that violate the expectation cause the job to fail.**
* **Records that violate the expectation are added to the target dataset and flagged as invalid in a field added to the target dataset.**
* **Records that violate the expectation are dropped from the target dataset and loaded into a quarantine table.**

**Explanation**

The answer is, Records that violate the expectation are added to the target dataset and recorded as invalid in the event log.

Delta live tables support three types of expectations to fix bad data in DLT pipelines

Review below example code to examine these expectations,



## Question 34

A dataset has been defined using Delta Live Tables and includes an expectations clause: CONSTRAINT valid\_timestamp EXPECT (timestamp > '2020-01-01') ON VIOLATION DROP ROW

What is the expected behavior when a batch of data containing data that violates these constraints is processed?

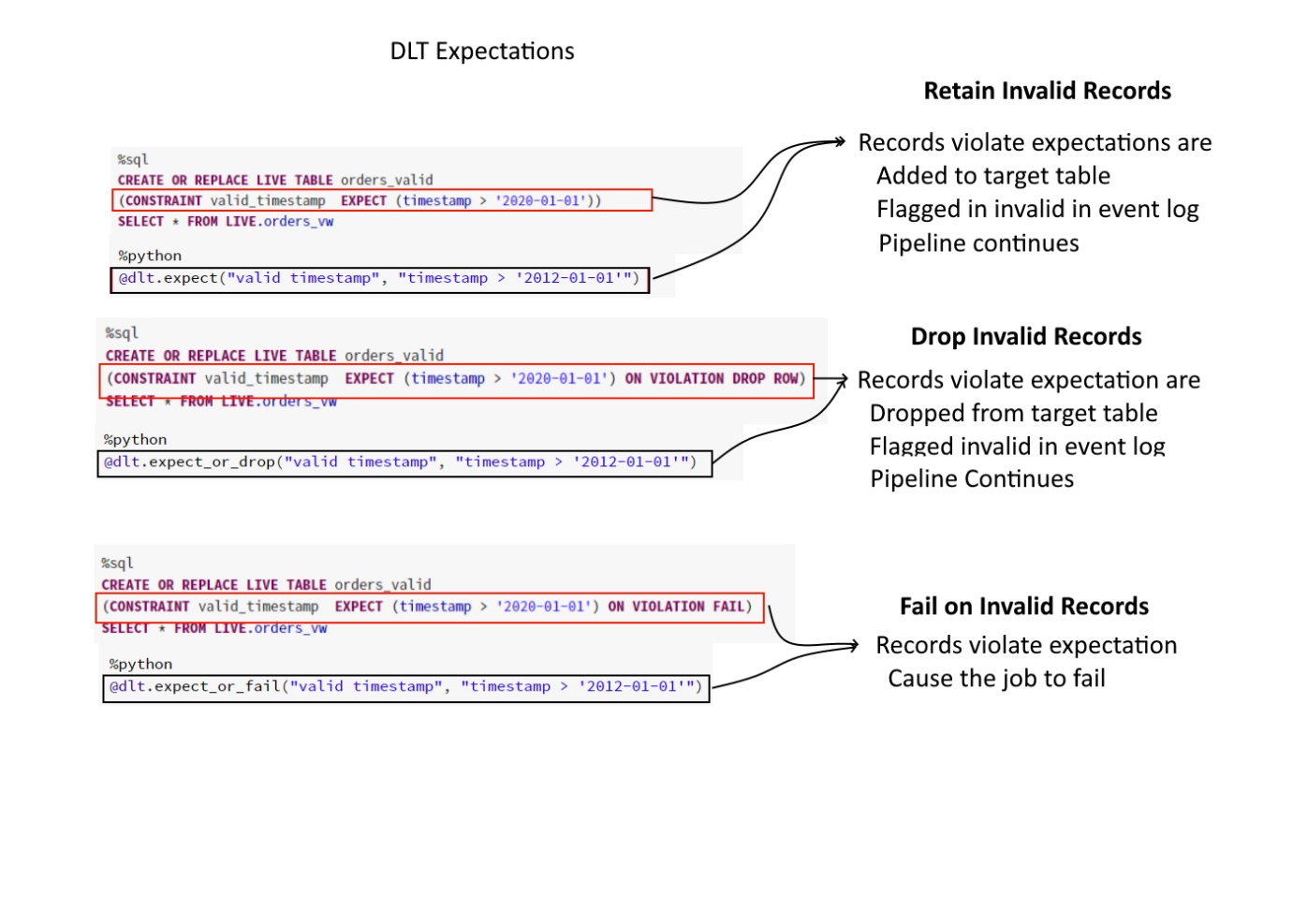
* **Records that violate the expectation are added to the target dataset and recorded as invalid in the event log.**
* **Records that violate the expectation are dropped from the target dataset and recorded as invalid in the event log.**
* **Records that violate the expectation cause the job to fail.**
* **Records that violate the expectation are added to the target dataset and flagged as invalid in a field added to the target dataset.**
* **Records that violate the expectation are dropped from the target dataset and loaded into a quarantine table.**

**Explanation**

The answer is Records that violate the expectation are dropped from the target dataset and recorded as invalid in the event log.

Delta live tables support three types of expectations to fix bad data in DLT pipelines

Review below example code to examine these expectations,



## Question 35

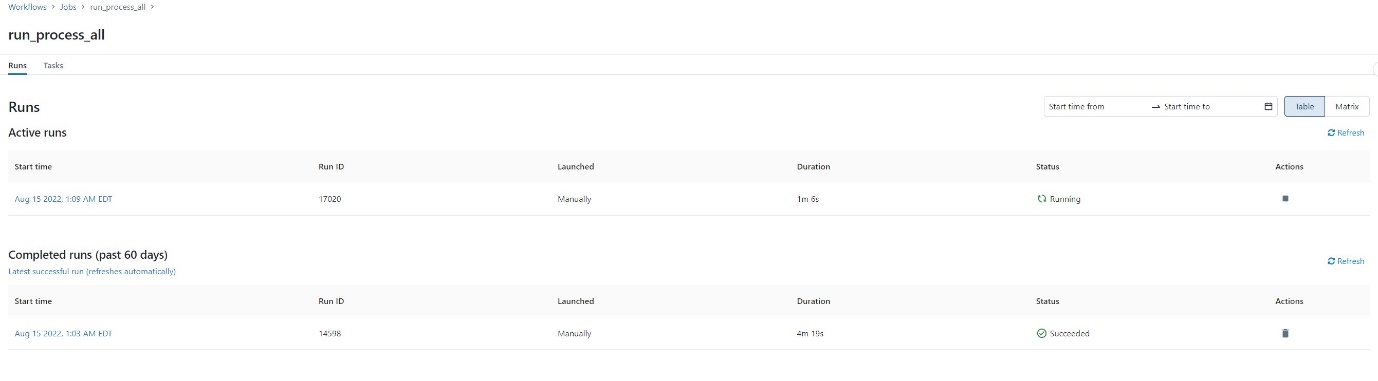
You are asked to debug a databricks job that is taking too long to run on Sunday’s, what are the steps you are going to take to identify the step that is taking longer to run?

* **A notebook activity of job run is only visible when using all-purpose cluster.**
* **Under Workflow UI and jobs select job you want to monitor and select the run, notebook activity can be viewed.**
* **Enable debug mode in the Jobs to see the output activity of a job, output should be available to view.**
* **Once a job is launched, you cannot access the job’s notebook activity.**
* **Use the compute’s spark UI to monitor the job activity.**

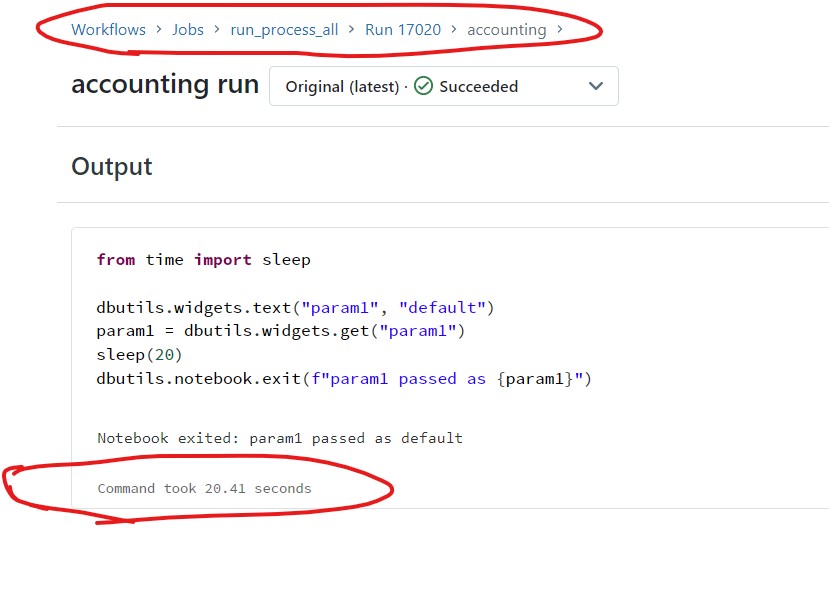
**Explanation**

The answer is, Under Workflow UI and jobs select job you want to monitor and select the run, notebook activity can be viewed.

You have the ability to view current active runs or completed runs, once you click the run you can see the



Click on the run to view the notebook output



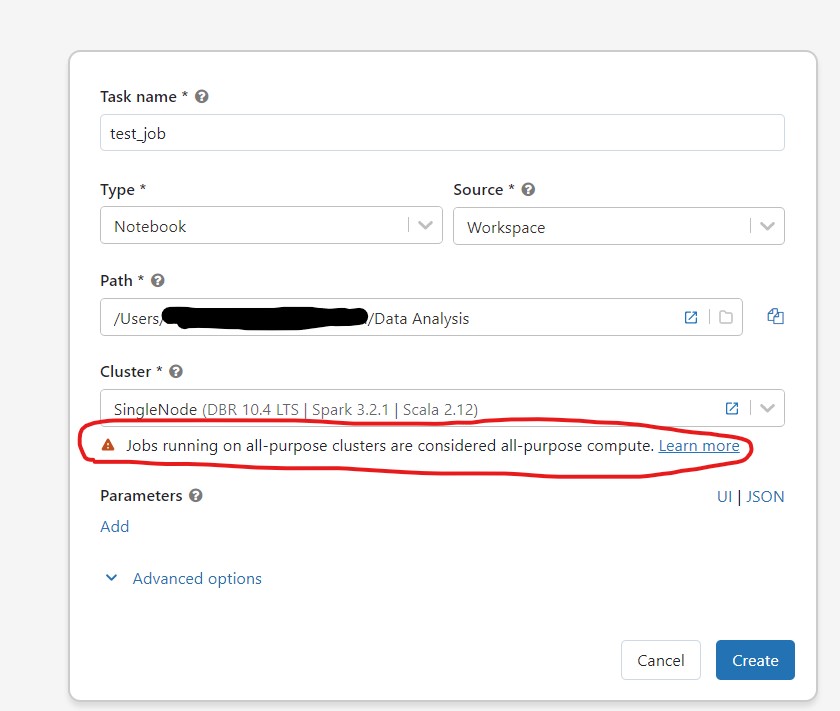
## Question 36

Your colleague was walking you through how a job was setup, but you noticed a warning message that said, “Jobs running on all-purpose cluster are considered all purpose compute", the colleague was not sure why he was getting the warning message, how do you best explain this warning message?

* **All-purpose clusters cannot be used for Job clusters, due to performance issues.**
* **All-purpose clusters take longer to start the cluster vs a job cluster**
* **All-purpose clusters are less expensive than the job clusters**
* **All-purpose clusters are more expensive than the job clusters**
* **All-purpose cluster provide interactive messages that can not be viewed in a job**

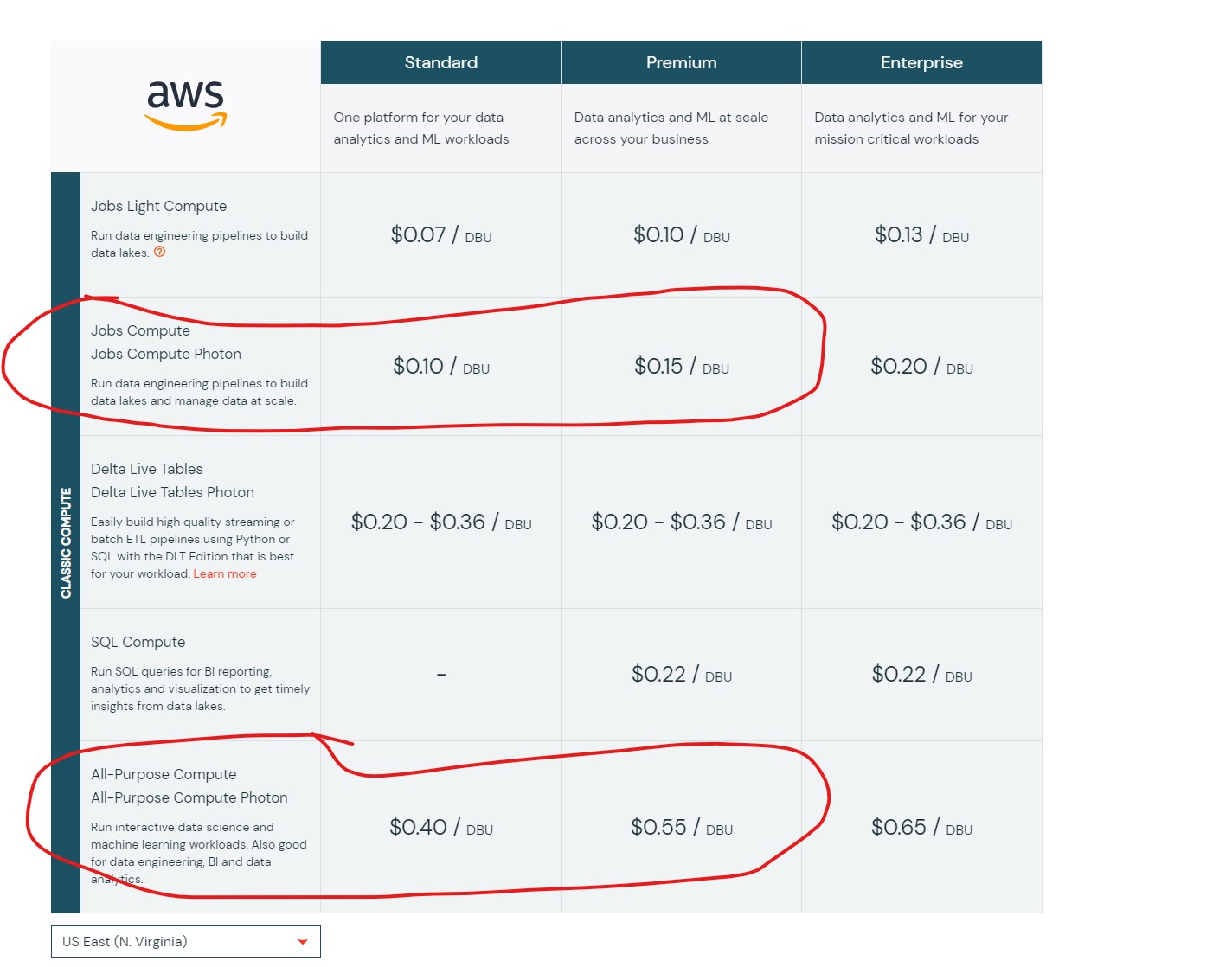
**Explanation**

Warning message:



Pricing for All-purpose clusters are more expensive than the job clusters

AWS pricing(Aug 15th 2022)



## Question 37

Your team has hundreds of jobs running but it is difficult to track cost of each job run, you are asked to provide a recommendation on how to monitor and track cost across various workloads

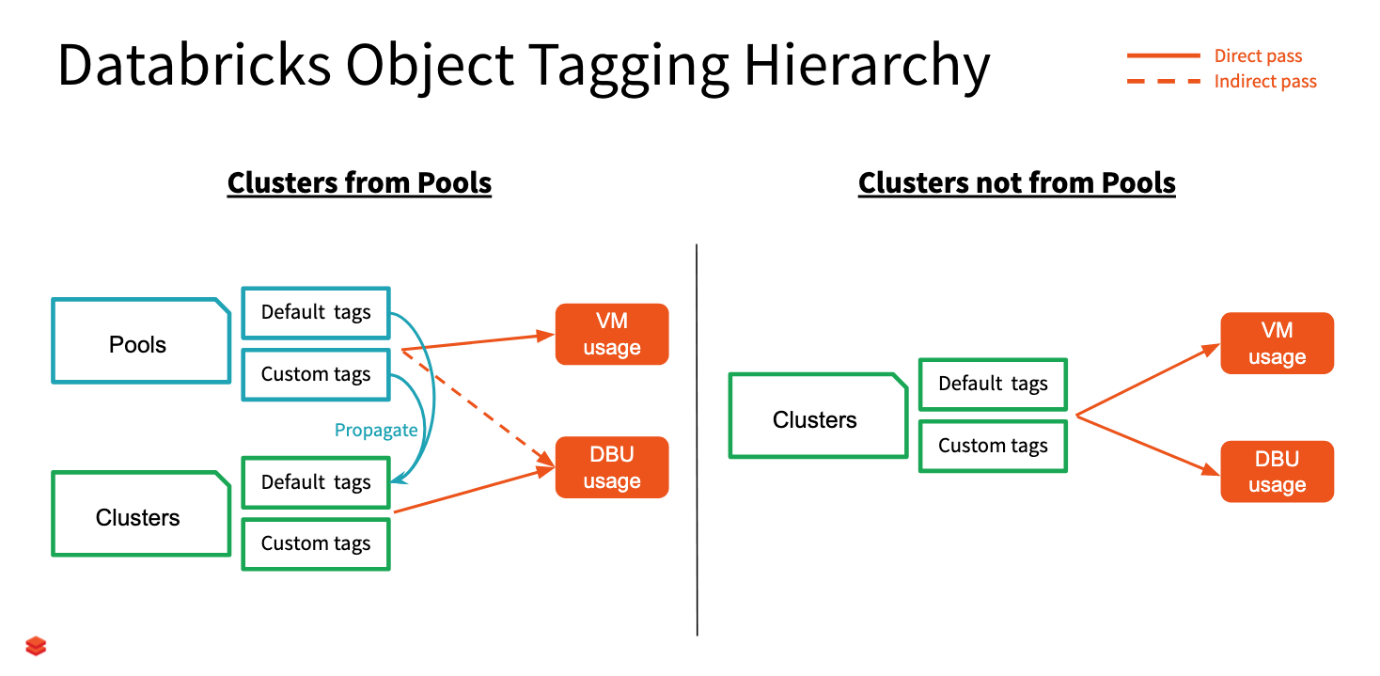
* **Create jobs in different workspaces, so we can track the cost easily**
* **Use Tags, during job creation so cost can be easily tracked**
* **Use job logs to monitor and track the costs**
* **Use workspace admin reporting**
* **Use a single cluster for all the jobs, so cost can be easily tracked**

**Explanation**

The answer is Use Tags, during job creation so cost can be easily tracked

Review below link for more details <https://docs.databricks.com/administration-guide/account-settings/usage-detail-tags-aws.html>

Here is a view how tags get propagated from pools to clusters and clusters without pools,



## Question 38

The sales team has asked the Data engineering team to develop a dashboard that shows sales performance for all stores, but the sales team would like to use the dashboard but would like to select individual store location, which of the following approaches Data Engineering team can use to build this functionality into the dashboard.

* **Use query Parameters which then allow user to choose any location**
* **Currently dashboards do not support parameters**
* **Use Databricks REST API to create a dashboard for each location**
* **Use SQL UDF function to filter the data based on the location**
* **Use Dynamic views to filter the data based on the location**

**Explanation**

The answer is

Databricks supports many types of parameters in the dashboard, a drop-down list can be created based on a query that has a unique list of store locations.

Here is a simple query that takes a parameter for

SELECT \*  FROM sales WHERE field IN ( {{ Multi Select Parameter }} )

Or

SELECT \*  FROM sales WHERE field =  {{ Single Select Parameter }}

**Query parameter types**

* [Text](https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/query-parameters#text)
* [Number](https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/query-parameters#number)
* [Dropdown List](https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/query-parameters#dropdown-list)
* [Query Based Dropdown List](https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/query-parameters#query-based-dropdown-list)
* [Date and Time](https://docs.microsoft.com/en-us/azure/databricks/sql/user/queries/query-parameters#date-and-time)

## Question 39

You are working on a dashboard that takes a long time to load in the browser, due to the fact that each visualization contains a lot of data to populate, which of the following approaches can be taken to address this issue?

* + **Increase size of the SQL endpoint cluster**
  + **Increase the scale of maximum range of SQL endpoint cluster**
  + **Use Databricks SQL Query filter to limit the amount of data in each visualization**
  + **Remove data from Delta Lake**
  + **Use Delta cache to store the intermediate results**

**Explanation**

Note\*: The question may sound misleading but these are types of questions the exam tries to ask.

A query filter lets you interactively reduce the amount of data shown in a visualization, similar to [query parameter](https://docs.databricks.com/sql/user/queries/query-parameters.html) but with a few key differences. A query filter limits data *after* it has been loaded into your browser. This makes filters ideal for smaller datasets and environments where query executions are time-consuming, rate-limited, or costly.

This query filter is different from than filter that needs to be applied at the data level, this filter is at the visualization level so you can toggle how much data you want to see.

1. SELECT action AS `action::filter`, COUNT(0) AS "actions count"
2. FROM events
3. GROUP BY action

When queries [have filters](https://docs.databricks.com/sql/user/queries/query-filters.html) you can also apply filters at the dashboard level. Select the **Use Dashboard Level Filters** checkbox to apply the filter to all queries.

[Dashboard filters](https://docs.databricks.com/sql/user/dashboards/index.html#dashboard-filters)

[Query filters | Databricks on AWS](https://docs.databricks.com/sql/user/queries/query-filters.html)

## Question 40

One of the queries in the Databricks SQL Dashboard takes a long time to refresh, which of the below steps can be taken to identify the root cause of this issue?

* **Restart the SQL endpoint**
* **Select the SQL endpoint cluster, spark UI, SQL tab to see the execution plan and time spent in each step**
* **Run optimize and Z ordering**
* **Change the Spot Instance Policy from “Cost optimized” to “Reliability Optimized.”**
* **Use Query History, to view queries and select query, and check query profile to time spent in each step**

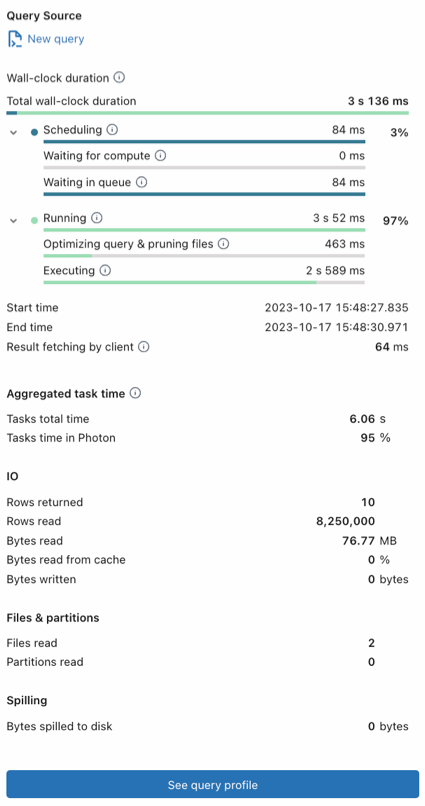
**Explanation**

The answer is, Use Query History, to view queries and select query, and check the query profile to see time spent in each step.

Here is the view of the query profile, for more info use the link

<https://docs.microsoft.com/en-us/azure/databricks/sql/admin/query-profile>

As you can see here Databricks SQL query profile is much different to Spark UI and provides much more clear information on how time is being spent on different queries and time it spent on each step.



## Question 41

A SQL Dashboard was built for the supply chain team to monitor the inventory and product orders, but all of the timestamps displayed on the dashboards are showing in UTC format, so they requested to change the time zone to the location of New York. How would you approach resolving this issue?

* **Move the workspace from Central US zone to East US Zone**
* **Change the timestamp on the delta tables to America/New\_York format**
* **Change the spark configuration of SQL endpoint to format the timestamp to America/New\_York**
* **Under SQL Admin Console, set the SQL configuration parameter time zone to America/New\_York**
* **Add SET Timezone = America/New\_York on every of the SQL queries in the dashboard.**

**Explanation**

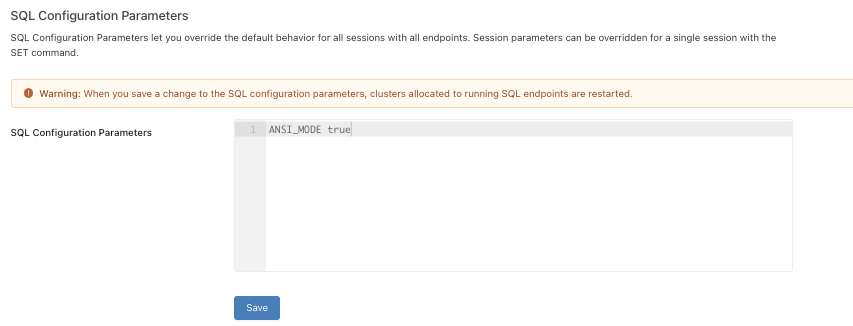
The answer is, Under SQL Admin Console, set the SQL configuration parameter time zone to America/New\_York

Here are steps you can take this to configure, so the entire dashboard is changed without changing individual queries

**Configure SQL parameters**

To configure all warehouses with SQL parameters:

1. Click  **Settings** at the bottom of the sidebar and select **SQL Admin Console**.
2. Click the **SQL Warehouse Settings** tab.
3. In the **SQL Configuration Parameters** textbox, specify one key-value pair per line. Separate the name of the parameter from its value using a space. For example, to enable ANSI\_MODE:



Similarly, we can add a line in the SQL Configuration parameters

**timezone America/New\_York**

[SQL configuration parameters | Databricks on AWS](https://docs.databricks.com/sql/admin/sql-configuration-parameters.html)

## Question 42

Which of the following technique can be used to implement fine-grained access control to rows and columns of the Delta table based on the user's access?

* + **Use Unity catalog to grant access to rows and columns**
  + **Row and column access control lists**
  + **Use dynamic view functions**
  + **Data access control lists**
  + **Dynamic Access control lists with Unity Catalog**

**Explanation**

The answer is, Use dynamic view functions.

Here is an example that limits access to rows based on the user being part managers group, in the below view if a user is not a part of the manager's group you can only see rows where the total amount is <= 1000000

Dynamic view function to filter rows

CREATE VIEW sales\_redacted AS

SELECT user\_id, country, product, total

FROM sales\_raw

WHERE CASE WHEN is\_member('managers') THEN TRUE ELSE total <= 1000000 END;

Dynamic view function to hide a column data based on user's access,

CREATE VIEW sales\_redacted AS

SELECT user\_id,

CASE WHEN is\_member('auditors') THEN email ELSE 'REDACTED' END AS email,

country,

product,

total

FROM sales\_raw

Please review below for more details

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

## Question 43

Unity catalog helps you manage the below resources in Databricks at account level

* **Tables**
* **ML Models**
* **Dashboards**
* **Meta Stores and Catalogs**
* **All of the above**

**Explanation**

The answer is all of the above.

Unity Catalog is a unified governance solution for all data and AI assets including files, tables, machine learning models, and dashboards in your lakehouse on any cloud.

## Question 44

John Smith is a newly joined team member in the Marketing team who currently has access read access to sales tables but does not have access to delete rows from the table, which of the following commands help you accomplish this?

* + **GRANT USAGE ON TABLE table\_name TO john.smith@marketing.com**
  + **GRANT DELETE ON TABLE table\_name TO john.smith@marketing.com**
  + **GRANT DELETE TO TABLE table\_name ON john.smith@marketing.com**
  + **GRANT MODIFY TO TABLE table\_name ON john.smith@marketing.com**
  + **GRANT MODIFY ON TABLE table\_name TO john.smith@marketing.com**

**Explanation**

The answer is GRANT MODIFY ON TABLE table\_name TO john.smith@marketing.com , please note INSERT, UPDATE, and DELETE are combined into one role called MODIFY.

Below are the list of privileges that can be granted to a user or a group,

SELECT: gives read access to an object.

CREATE: gives the ability to create an object (for example, a table in a schema).

**MODIFY: gives the ability to add, delete, and modify data to or from an object.**

USAGE: does not give any abilities, but is an additional requirement to perform any action on a schema object.

READ\_METADATA: gives the ability to view an object and its metadata.

CREATE\_NAMED\_FUNCTION: gives the ability to create a named UDF in an existing catalog or schema.

MODIFY\_CLASSPATH: gives the ability to add files to the Spark classpath.

ALL PRIVILEGES: gives all privileges (is translated into all the above privileges

## Question 45

Kevin is the owner of both the sales table and regional\_sales\_vw view which uses the sales table as the underlying source for the data, and Kevin is looking to grant select privilege on the view regional\_sales\_vw to one of newly joined team members Steven. Which of the following is a true statement?

* **Kevin can not grant access to Steven since he does not have security admin privilege**
* **Kevin although is the owner but does not have ALL PRIVILEGES permission**
* **Kevin can grant access to the view, because he is the owner of the view and the underlying table**
* **Kevin can not grant access to Steven since he does have workspace admin privilege**
* **Steve will also require SELECT access on the underlying table**

**Explanation**

The answer is, Kevin can grant access to the view, because he is the owner of the view and the underlying table,

Ownership determines whether or not you can grant privileges on derived objects to other users, a user who creates a schema, table, view, or function becomes its owner. The owner is granted all privileges and can grant privileges to other users

Bottom of Form