



- Expert Verified, Online, **Free**.



## **CERTIFICATION TEST**

- [CertificationTest.net](https://CertificationTest.net) - Cheap & Quality Resources With Best Support



An upstream system has been configured to pass the date for a given batch of data to the Databricks Jobs API as a parameter. The notebook to be scheduled will use this parameter to load data with the following code: `df = spark.read.format("parquet").load(f"/mnt/source/{date}")`  
Which code block should be used to create the date Python variable used in the above code block?

- A. `date = spark.conf.get("date")`
- B. `input_dict = input()`  
`date= input_dict["date"]`
- C. `import sys`  
`date = sys.argv[1]`
- D. `date = dbutils.notebooks.getParam("date")`
- E. `dbutils.widgets.text("date", "null")`  
`date = dbutils.widgets.get("date")`

**Suggested Answer: E**

Community vote distribution

E (100%)

 **hal2401me**  9 months, 1 week ago

**Selected Answer: E**

`dbutils.widget.`

Just passed the exam with score >80%.

examtopics covers about 90% of questions. there were 5 questions I didn't see here in examtopics.

But friends, you need to look at the discussions, and do test yourself.

many answers provided here, even most voted answer, does NOT exists anymore in the exam - not the question, but the answer.

Wish you all good luck, friends!

upvoted 12 times

 **Gowtham02**  2 weeks, 3 days ago

**Selected Answer: E**

widget is correct

upvoted 1 times

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

Surely E

upvoted 2 times

 **prasioso** 1 week, 3 days ago

Agree with E. Did you clear the exam? Are the questions still relevant?

upvoted 1 times

 **ultimomassimo** 3 months ago

**Selected Answer: E**

E 100%

There is no such thing as `dbutils.notebooks.getParam`, so no idea why some halfwits suggest D...

upvoted 1 times

 **shoaibmohammed3** 4 months ago

**Selected Answer: E**

`dbutils.widgets` is where we store all the param and use `.get` to fetch the params

upvoted 1 times

 **johnserafim** 4 months, 1 week ago

**Selected Answer: D**

D is correct!

The question states that the upstream system passes the date as a parameter to the Databricks Jobs API. In Databricks, when a parameter is passed to a notebook via the Jobs API, it can be retrieved using the `dbutils.notebooks.getParam()` method.

Option D directly retrieves the parameter value using this method, which is the correct approach for this scenario.

upvoted 1 times

🗳️ 👤 **ElkeV** 4 months, 3 weeks ago

**Selected Answer: E**

It is the way to fill in a parameter in a notebook

upvoted 1 times

🗳️ 👤 **HairyTorso** 5 months, 4 weeks ago

**Selected Answer: E**

Around question #130 they just repeat themselves. So it's not 226 but around 130... Shame

upvoted 2 times

🗳️ 👤 **akashdesarda** 9 months ago

**Selected Answer: E**

Jobs API allows to sending parameters via jobs parameter. This parameter must have the same notebook params. Eventually, it can be read using `dbutils.widgets.get`

upvoted 1 times

🗳️ 👤 **HorskKos** 10 months, 3 weeks ago

E is correct because:

A - gets configuration of a spark session

B - gets a value from a manual input - non relevant for the job run

C - `sys.argv` - gets parameters, which were used to run a Python script from CMD - completely non-related

D - haven't found this function on the web at all, assume that it doesn't exist

Therefore E is correct, though it's a bad practice to type a date as a parameter, it's better to get it with datetime library and then use it in the code

upvoted 4 times

🗳️ 👤 **Shailly** 11 months, 3 weeks ago

Answer is E.

Even though the value is passed from an upstream system, you can create parameters using widgets inside notebook and use the value as an input from the databricks jobs API.

upvoted 1 times

🗳️ 👤 **Isio05** 1 year ago

**Selected Answer: E**

Widgets are used to create parameters in notebook that can be then utilized by e.g. jobs

upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

E.

E. `dbutils.widgets.text("date", "null")`

`date = dbutils.widgets.get("date")`

upvoted 1 times

🗳️ 👤 **AziLa** 1 year, 2 months ago

correct ans is E

upvoted 1 times

🗳️ 👤 **Sosicha** 1 year, 2 months ago

Are you reading the question? It asks about an upstream system that has been configured to pass the date for a given batch of data to the Databricks Jobs API as a parameter. Upstream system usually don't use widgets. Widgets they are made for humans. Only C and D are correct but D is better so D.

upvoted 1 times

🗳️ 👤 **hal2401me** 1 year, 4 months ago

**Selected Answer: E**

vote for E

`dbutils.widget`

upvoted 1 times

🗳️ 👤 **AziLa** 1 year, 5 months ago

Correct Ans is E  
upvoted 1 times

The Databricks workspace administrator has configured interactive clusters for each of the data engineering groups. To control costs, clusters are set to terminate after 30 minutes of inactivity. Each user should be able to execute workloads against their assigned clusters at any time of the day.

Assuming users have been added to a workspace but not granted any permissions, which of the following describes the minimal permissions a user would need to start and attach to an already configured cluster.

- A. "Can Manage" privileges on the required cluster
- B. Workspace Admin privileges, cluster creation allowed, "Can Attach To" privileges on the required cluster
- C. Cluster creation allowed, "Can Attach To" privileges on the required cluster
- D. "Can Restart" privileges on the required cluster
- E. Cluster creation allowed, "Can Restart" privileges on the required cluster

**Suggested Answer: A**

Community vote distribution

D (100%)

🗳️ 👤 **dkhodyriev** 6 days ago

**Selected Answer: D**

The answer is D. "Can Restart" privileges on the required cluster.

This is the minimal permission needed because:

It allows starting terminated clusters (necessary due to 30-minute auto-termination)

It includes the ability to attach to and execute workloads on clusters

It doesn't require cluster creation permissions since clusters already exist

It's less privileged than "Can Manage" which would include unnecessary configuration editing rights

upvoted 1 times

🗳️ 👤 **SP\_07** 1 month, 3 weeks ago

**Selected Answer: D**

The minimal permission a user needs to start (i.e., restart a terminated cluster) and attach to an already configured Databricks cluster is "Can Restart" privileges on the required cluster. This allows users to restart and attach to clusters but not manage or modify them.

upvoted 1 times

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: D**

suretly D

upvoted 1 times

🗳️ 👤 **codebender** 2 months, 3 weeks ago

**Selected Answer: D**

Restart is necessary to start the cluster

upvoted 2 times

🗳️ 👤 **Ashok\_Choudhary\_CT** 3 months ago

**Selected Answer: A**

There are three types of permissions:

1. Can Restart
2. Can Attach To
3. Can Manage

Can manage provides both the permissions. Can restart doesn't provide "Can Attach To" permission, So the correct answer is "A"

upvoted 2 times

🗳️ 👤 **ultimomassimo** 3 months ago

**Selected Answer: D**

100% D

User has to be able to start the cluster as it shuts down after 30 min of inactivity.

Everything is explain clearly in the table here - Compute ACLs: <https://docs.databricks.com/aws/en/security/auth/access-control/#clusters>

upvoted 3 times

🗨️ 👤 **Tedet** 4 months ago

**Selected Answer: C**

To execute workloads against an already configured cluster in Databricks, users need the following minimum permissions:

1. Cluster Creation Allowed: Users need the ability to create clusters, which ensures they can start a cluster if it's not already running or if the cluster has been terminated after the inactivity period.
2. "Can Attach To" Privileges on the Required Cluster: This permission allows users to attach their notebooks or jobs to the existing cluster. The "Can Attach To" permission is the key to allowing users to interact with the cluster for running jobs or notebooks.

upvoted 1 times

🗨️ 👤 **johnserafim** 4 months, 1 week ago

**Selected Answer: C**

I should choose the C answer because the "Can Attach To" permission is the minimal requirement for a user to attach to an interactive cluster and execute workloads.

upvoted 1 times

🗨️ 👤 **shaswat1404** 4 months, 3 weeks ago

**Selected Answer: D**

can manage all permission

workspace admin irrelevant

can attach to cannot start the cluster

can restart correct can do both

cluster creation + restart (cluster is already created, dont need to give permission)

upvoted 1 times

🗨️ 👤 **EelkeV** 4 months, 3 weeks ago

**Selected Answer: C**

You do not need all the permissions, just the lowest

upvoted 1 times

🗨️ 👤 **Dhusanth** 5 months ago

**Selected Answer: D**

<https://docs.azure.cn/en-us/databricks/security/auth-authz/access-control/cluster-acl#cluster-permissions>

upvoted 4 times

🗨️ 👤 **nadegetiedjo** 5 months, 2 weeks ago

**Selected Answer: D**

if added to a workspace, it has the default settings of the users in that workspace

upvoted 1 times

🗨️ 👤 **sakis213** 5 months, 2 weeks ago

**Selected Answer: C**

Can Restart only allows restarting the cluster and does not grant permission to attach workloads.

upvoted 2 times

🗨️ 👤 **yeyi97** 5 months, 3 weeks ago

**Selected Answer: C**

Option C makes more sense since the question says also attach. Not just restart.

upvoted 1 times

🗨️ 👤 **rockreid** 6 months, 2 weeks ago

**Selected Answer: C**

To execute workloads, users need to be able to attach their notebooks or jobs to the cluster. The "Can Attach To" privilege specifically allows users to attach to and use the cluster, which is essential for running their workloads.



upvoted 1 times

🗨️ 👤 **akashdesarda** 9 months ago

**Selected Answer: D**

Questions is users need to start & use so it will be Can restrat . Can attach cannot start compute

upvoted 1 times

  **Robbyisok** 9 months, 3 weeks ago

D is the correct answer. Focus on this line "user would need to start and attach to an already configured cluster."

upvoted 2 times

When scheduling Structured Streaming jobs for production, which configuration automatically recovers from query failures and keeps costs low?

- A. Cluster: New Job Cluster;  
Retries: Unlimited;  
Maximum Concurrent Runs: Unlimited
- B. Cluster: New Job Cluster;  
Retries: None;  
Maximum Concurrent Runs: 1
- C. Cluster: Existing All-Purpose Cluster;  
Retries: Unlimited;  
Maximum Concurrent Runs: 1
- D. Cluster: New Job Cluster;  
Retries: Unlimited;  
Maximum Concurrent Runs: 1
- E. Cluster: Existing All-Purpose Cluster;  
Retries: None;  
Maximum Concurrent Runs: 1

**Suggested Answer: D**

Community vote distribution

D (100%)

🗳️ 👤 **8605246** Highly Voted 1 year, 10 months ago

the answer given is correct:

Maximum concurrent runs: Set to 1. There must be only one instance of each query concurrently active.

Retries: Set to Unlimited.

<https://docs.databricks.com/en/structured-streaming/query-recovery.html>

upvoted 11 times

🗳️ 👤 **KadELbied** Most Recent 1 month, 3 weeks ago

Selected Answer: D

Surely d

upvoted 1 times

🗳️ 👤 **codebender** 2 months, 3 weeks ago

Selected Answer: D

Cant be all purpose general compute

upvoted 1 times

🗳️ 👤 **EelkeV** 4 months, 3 weeks ago

Selected Answer: D

Job cluster autoterminates, and you want retries for recover

upvoted 1 times

🗳️ 👤 **akashdesarda** 9 months ago

Selected Answer: D

Use databricks jobs as it as native integration with Streaming use case. See the example Job here <https://docs.databricks.com/en/structured-streaming/query-recovery.html#configure-structured-streaming-jobs-to-restart-streaming-queries-on-failure>

upvoted 2 times

🗳️ 👤 **imatheushenrique** 1 year ago

D. Cluster: New Job Cluster;

Retries: Unlimited;

Maximum Concurrent Runs: 1

upvoted 1 times



🗨️ 👤 **imatheushenrique** 1 year ago

D. Cluster: New Job Cluster;  
Retries: Unlimited;  
Maximum Concurrent Runs: 1  
upvoted 1 times

🗨️ 👤 **juliom6** 1 year, 2 months ago

D is correct  
<https://docs.databricks.com/en/structured-streaming/query-recovery.html>  
upvoted 1 times

🗨️ 👤 **AziLa** 1 year, 5 months ago

Correct Ans is D  
upvoted 1 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: D**

D is correct  
upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 6 months ago

**Selected Answer: D**

D is correct  
upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: D**

D is correct  
upvoted 1 times

The data engineering team has configured a Databricks SQL query and alert to monitor the values in a Delta Lake table. The recent\_sensor\_recordings table contains an identifying sensor\_id alongside the timestamp and temperature for the most recent 5 minutes of recordings.

The below query is used to create the alert:

```
SELECT MEAN(temperature), MAX(temperature), MIN(temperature)
FROM recent_sensor_recordings
GROUP BY sensor_id
```

The query is set to refresh each minute and always completes in less than 10 seconds. The alert is set to trigger when mean (temperature) > 120. Notifications are triggered to be sent at most every 1 minute.

If this alert raises notifications for 3 consecutive minutes and then stops, which statement must be true?

- A. The total average temperature across all sensors exceeded 120 on three consecutive executions of the query
- B. The recent\_sensor\_recordings table was unresponsive for three consecutive runs of the query
- C. The source query failed to update properly for three consecutive minutes and then restarted
- D. The maximum temperature recording for at least one sensor exceeded 120 on three consecutive executions of the query
- E. The average temperature recordings for at least one sensor exceeded 120 on three consecutive executions of the query

**Suggested Answer: E**

Community vote distribution

E (100%)

🗳️ **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

🗳️ **EelkeV** 4 months, 3 weeks ago

**Selected Answer: E**

Because the mean is calculated for each sensor, and on that the alert is raised. It happened for three times, unknown for which sensor. Could be any

upvoted 3 times

🗳️ **AndreFR** 10 months, 2 weeks ago

A excluded because there is a group by clause

B & C excluded table needs to be updated to mean value to change

D excluded, because alert is set on average not max temperature

Correct answer is E by elimination

upvoted 2 times

🗳️ **panya** 1 year ago

Correct

upvoted 1 times

🗳️ **imatheushenrique** 1 year ago

E. The average temperature recordings for at least one sensor exceeded 120 on three consecutive executions of the query

upvoted 1 times

🗳️ **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: E**

E is correct

upvoted 1 times

🗳️ **sturcu** 1 year, 8 months ago

**Selected Answer: E**

correct

upvoted 2 times

🗳️ **saikot** 1 year, 9 months ago

The correct answer is E

<https://www.myexamcollection.com/databricks-certified-professional-data-engineer-databricks-certified-professional-data-engineer-exam-question-answers.htm>

upvoted 1 times

A junior developer complains that the code in their notebook isn't producing the correct results in the development environment. A shared screenshot reveals that while they're using a notebook versioned with Databricks Repos, they're using a personal branch that contains old logic. The desired branch named dev-2.3.9 is not available from the branch selection dropdown. Which approach will allow this developer to review the current logic for this notebook?

- A. Use Repos to make a pull request use the Databricks REST API to update the current branch to dev-2.3.9
- B. Use Repos to pull changes from the remote Git repository and select the dev-2.3.9 branch.
- C. Use Repos to checkout the dev-2.3.9 branch and auto-resolve conflicts with the current branch
- D. Merge all changes back to the main branch in the remote Git repository and clone the repo again
- E. Use Repos to merge the current branch and the dev-2.3.9 branch, then make a pull request to sync with the remote repository

**Suggested Answer: B**

Community vote distribution

B (100%)

🗳️ 👤 **Gowtham02** 2 weeks, 3 days ago

**Selected Answer: B**

B is correct  
upvoted 1 times

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

suretly B  
upvoted 1 times

🗳️ 👤 **codebender** 2 months, 3 weeks ago

**Selected Answer: B**

First step is to pull from the latest commits  
upvoted 2 times

🗳️ 👤 **benni\_ale** 8 months, 3 weeks ago

**Selected Answer: B**

I would also say B but could anyone explain how to pick that branch if it is not available from dropdown?  
upvoted 1 times

🗳️ 👤 **NBurman** 7 months, 3 weeks ago

The first step is to perform a pull. This fetches all the changes from the remote branch and only after that will you see the dev branch.  
upvoted 4 times

🗳️ 👤 **benni\_ale** 9 months, 1 week ago

**Selected Answer: B**

I would say B  
upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

B. Use Repos to pull changes from the remote Git repository and select the dev-2.3.9 branch.  
upvoted 2 times

🗳️ 👤 **AziLa** 1 year, 5 months ago

correct ans is B  
upvoted 1 times

🗳️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: B**

vote for B also  
upvoted 2 times

🗳️ 👤 **sturcu** 1 year, 8 months ago

Selected Answer: B

B is correct

upvoted 1 times

The security team is exploring whether or not the Databricks secrets module can be leveraged for connecting to an external database. After testing the code with all Python variables being defined with strings, they upload the password to the secrets module and configure the correct permissions for the currently active user. They then modify their code to the following (leaving all other variables unchanged).

```
password = dbutils.secrets.get(scope="db_creds", key="jdbc_password")

print(password)

df = (spark
      .read
      .format("jdbc")
      .option("url", connection)
      .option("dbtable", tablename)
      .option("user", username)
      .option("password", password)
      )
```

Which statement describes what will happen when the above code is executed?

- A. The connection to the external table will fail; the string "REDACTED" will be printed.
- B. An interactive input box will appear in the notebook; if the right password is provided, the connection will succeed and the encoded password will be saved to DBFS.
- C. An interactive input box will appear in the notebook; if the right password is provided, the connection will succeed and the password will be printed in plain text.
- D. The connection to the external table will succeed; the string value of password will be printed in plain text.
- E. The connection to the external table will succeed; the string "REDACTED" will be printed.

**Suggested Answer: E**

Community vote distribution


E (100%)

 **akashdesarda** Highly Voted 9 months ago

**Selected Answer: E**

Whatever we read using dbutils.secret module is always printed as '[REDACTED]', but when consumed in code, underlying vales are passed.


upvoted 5 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

Suretly E

upvoted 2 times

 **benni\_ale** 8 months, 3 weeks ago

**Selected Answer: E**

Shave like a bomber

upvoted 1 times

 **imatheushenrique** 1 year ago


E. The connection to the external table will succeed; the string "REDACTED" will be printed.

upvoted 1 times

 **PrashantTiwari** 1 year, 4 months ago

E is correct

upvoted 1 times

 **AziLa** 1 year, 5 months ago

correct ans is E

upvoted 1 times

 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: E**

E is correct

upvoted 2 times

🗲️ 👤 **ATLTennis** 1 year, 5 months ago

**Selected Answer: E**

E is correct

upvoted 2 times

🗲️ 👤 **kz\_data** 1 year, 6 months ago

**Selected Answer: E**

Correct answer E

upvoted 2 times

🗲️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: E**

Correct: <https://docs.databricks.com/en/external-data/jdbc.html>

upvoted 4 times

🗲️ 👤 **Brian9** 1 year, 10 months ago

<https://learn.microsoft.com/en-us/azure/databricks/security/secrets/redaction>

Option E - which is selected answer seems correct.

upvoted 3 times

🗲️ 👤 **8605246** 1 year, 10 months ago

This option is correct, although the password won't be printed out, the connection will still succeed.

upvoted 1 times

The data science team has created and logged a production model using MLflow. The following code correctly imports and applies the production model to output the predictions as a new DataFrame named preds with the schema "customer\_id LONG, predictions DOUBLE, date DATE".

```
from pyspark.sql.functions import current_date

model = mlflow.pyfunc.spark_udf(spark, model_uri="models:/churn/prod")
df = spark.table("customers")
columns = ["account_age", "time_since_last_seen", "app_rating"]
preds = (df.select(
    "customer_id",
    model(*columns).alias("predictions"),
    current_date().alias("date")
))
```

The data science team would like predictions saved to a Delta Lake table with the ability to compare all predictions across time. Churn predictions will be made at most once per day.

Which code block accomplishes this task while minimizing potential compute costs?

A. `preds.write.mode("append").saveAsTable("churn_preds")`

B. `preds.write.format("delta").save("/preds/churn_preds")`

C. `(preds.writeStream
 .outputMode("overwrite")
 .option("checkpointPath", "_checkpoints/churn_preds")
 .start("/preds/churn_preds")
)`

D. `(preds.write
 .format("delta")
 .mode("overwrite")
 .saveAsTable("churn_preds")
)`

E. `(preds.writeStream
 .outputMode("append")
 .option("checkpointPath", "_checkpoints/churn_preds")
 .table("churn_preds")
)`

**Suggested Answer: A**

Community vote distribution

A (100%)

 **thxsgod** Highly Voted 1 year, 9 months ago

**Selected Answer: A**

You need:

- Batch operation since it is at most once a day
- Append, since you need to keep track of past predictions

A is the correct answer. You don't need to specify "format" when you use `saveAsTable`.


upvoted 13 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

 **benni\_ale** 8 months, 3 weeks ago

**Selected Answer: A**

Batch, Append

upvoted 1 times

 **coercion** 1 year, 1 month ago

**Selected Answer: A**



default table format is delta so no need to specify the format.

As per the requirement, "append" mode is required to maintain the history. Default mode is "ErrorIfExists"

upvoted 1 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: A**

A is correct

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 6 months ago

**Selected Answer: A**

A is correct

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: A**

Correct

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

Correct

upvoted 1 times

🗨️ 👤 **Eertyy** 1 year, 9 months ago

answer is B

upvoted 2 times

🗨️ 👤 **Starvosxant** 1 year, 8 months ago

First: the default node Databricks saves tables IS Delta Format. So no reason why you say it wouldn't benefit from Lakehouse features.

Second: the default write mode is Error, means that if you try to write to a location and that already exists there, it will prone a Error. And the question specify that you gonna write once a day.

You better revisit basic topics before continue to the professional level certification, or buy the dump entirely.

upvoted 4 times

🗨️ 👤 **Eertyy** 1 year, 9 months ago

Here's why:

A. saves the data as a managed table, which may not be efficient for large-scale data or frequent updates. It doesn't utilize Delta Lake capabilities.

C. is used for streaming operations, not batch processing. Also, using "overwrite" as output mode will replace the existing data each time, which is not suitable for keeping historical predictions.

D. is similar to option A but with "overwrite" mode. It will replace the entire table each time, which is not suitable for maintaining a historical record of predictions.

E. is also for streaming operations and not for batch processing. Additionally, it uses the "table" method, which is not typically used for writing batch data into Delta Lake tables.

Option B is suitable for batch processing, writes data in Delta Lake format, and allows you to efficiently maintain a historical record of predictions while minimizing compute costs.

upvoted 3 times

🗨️ 👤 **pradyumn9999** 1 year, 8 months ago

Its also said they want to compare past values as well, so mode needs to be append. By default is error mode.

upvoted 4 times

🗨️ 👤 **buggumaster** 1 year, 10 months ago

Selected answer is wrong, not write Format is specified in A.

upvoted 1 times

🗨️ 👤 **buggumaster** 1 year, 10 months ago

Selected answer is wrong, not writeMode is specified in A.

upvoted 1 times

An upstream source writes Parquet data as hourly batches to directories named with the current date. A nightly batch job runs the following code to ingest all data from the previous day as indicated by the date variable:

```
(spark.read
  .format("parquet")
  .load(f"/mnt/raw_orders/{date}")
  .dropDuplicates(["customer_id", "order_id"])
  .write
  .mode("append")
  .saveAsTable("orders")
)
```

Assume that the fields `customer_id` and `order_id` serve as a composite key to uniquely identify each order.

If the upstream system is known to occasionally produce duplicate entries for a single order hours apart, which statement is correct?

- A. Each write to the orders table will only contain unique records, and only those records without duplicates in the target table will be written.
- B. Each write to the orders table will only contain unique records, but newly written records may have duplicates already present in the target table.
- C. Each write to the orders table will only contain unique records; if existing records with the same key are present in the target table, these records will be overwritten.
- D. Each write to the orders table will only contain unique records; if existing records with the same key are present in the target table, the operation will fail.
- E. Each write to the orders table will run deduplication over the union of new and existing records, ensuring no duplicate records are present.

**Suggested Answer: B**

Community vote distribution

B (100%)

  **Eertyy**  1 year, 9 months ago

B. Each write to the orders table will only contain unique records, but newly written records may have duplicates already present in the target table.



Explanation:

In the provided code, the `.dropDuplicates(["customer_id", "order_id"])` operation is performed on the data loaded from the Parquet files. This operation ensures that only unique records, based on the composite key of "customer\_id" and "order\_id," are retained in the DataFrame before writing to the "orders" table.

However, this operation does not consider duplicates that may already exist in the "orders" table. It only filters duplicates from the current batch of data. If there are duplicates in the "orders" table from previous batches, they will remain in the table.



So, newly written records will not have duplicates within the batch being written, but duplicates from previous batches may still exist in the target table.

upvoted 21 times

  **meatpoof** 5 months, 1 week ago

The question doesn't say orders already exists. Arekm's answer is more correct

upvoted 1 times

  **KadELbied**  1 month, 3 weeks ago

**Selected Answer: B**

suretly B

upvoted 1 times

  **arekm** 6 months ago

**Selected Answer: B**

No duplicates in the current batch - that is obvious. The duplicates may happen since the source occasionally produces duplicates hours apart. This means that one record can be generated by the source and processed on day 1, the duplicate on day 2. Since there is no logic checking if the corresponding record exists in the target - you get the duplicates there given we use append mode.

upvoted 2 times

🗲️ 👤 **Anithec0der** 6 months, 3 weeks ago

**Selected Answer: B**

yeah B is the correct answer cause in the current code it will look for duplicates in the currentDF based on composite keys and not for the duplicates which are already in the target table. if we want to insert for the rows which are not there in target table then we can make use of Merge Into statement of databricks.

upvoted 1 times

🗲️ 👤 **benni\_ale** 9 months, 1 week ago

**Selected Answer: B**

Append method does not take in consideration any key in the target table, it simply add all rows of the input table to the target table.

upvoted 1 times

🗲️ 👤 **panya** 1 year ago

Yes it should be B

upvoted 1 times

🗲️ 👤 **imatheushenrique** 1 year ago

B. Each write to the orders table will only contain unique records, but newly written records may have duplicates already present in the target table.

Using merge this problem would not happen

upvoted 1 times

🗲️ 👤 **DavidRou** 1 year, 3 months ago

**Selected Answer: B**

B is the right answer. The above code only remove duplicates from the batch that is processed, no logic is applied to already saved records.

upvoted 1 times

🗲️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: B**

B is correct

upvoted 1 times

🗲️ 👤 **5ffcd04** 1 year, 6 months ago

**Selected Answer: B**

Answer B

upvoted 1 times

🗲️ 👤 **kz\_data** 1 year, 6 months ago

**Selected Answer: B**

B is correct

upvoted 1 times

🗲️ 👤 **vivekla** 1 year, 7 months ago

correct B

upvoted 1 times

🗲️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: B**

Correct

upvoted 1 times

🗲️ 👤 **Starvosxant** 1 year, 8 months ago

Correct. B

upvoted 1 times

🗲️ 👤 **thxsgod** 1 year, 9 months ago

**Selected Answer: B**

Correct

upvoted 2 times

A junior member of the data engineering team is exploring the language interoperability of Databricks notebooks. The intended outcome of the below code is to register a view of all sales that occurred in countries on the continent of Africa that appear in the geo\_lookup table. Before executing the code, running SHOW TABLES on the current database indicates the database contains only two tables: geo\_lookup and sales.

**Cmd 1**

```
%python
countries_af = [x[0] for x in
spark.table("geo_lookup").filter("continent='AF']").select("country").collect()]
```

**Cmd 2**

```
%sql
CREATE VIEW sales_af AS
SELECT *
FROM sales
WHERE city IN countries_af
AND CONTINENT = "AF"
```

Which statement correctly describes the outcome of executing these command cells in order in an interactive notebook?


- A. Both commands will succeed. Executing show tables will show that countries\_af and sales\_af have been registered as views.
- B. Cmd 1 will succeed. Cmd 2 will search all accessible databases for a table or view named countries\_af: if this entity exists, Cmd 2 will succeed.
- C. Cmd 1 will succeed and Cmd 2 will fail. countries\_af will be a Python variable representing a PySpark DataFrame.
- D. Both commands will fail. No new variables, tables, or views will be created.
- E. Cmd 1 will succeed and Cmd 2 will fail. countries\_af will be a Python variable containing a list of strings.

**Suggested Answer: E**

Community vote distribution

E (88%)

12%

 **aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: E**

Cmd 1 is a PySpark command that collects the list of countries from the 'geo\_lookup' table where the continent is Africa ('AF'). This command will execute successfully, resulting in countries\_af being a list of country names (strings) in Python's local memory.

Cmd 2 is an SQL command intended to create a view named 'sales\_af' from the 'sales' table, filtered by the cities in the countries\_af list. However, this will fail because the countries\_af variable exists in the Python environment and is not recognized in the SQL context. SQL does not have access to Python variables directly; they are two separate execution contexts within a Databricks notebook. There is no table or view named countries\_af that SQL can reference; it is merely a Python list variable.

The other options are incorrect because they either assume cross-contextual operation between Python and SQL within a Databricks notebook (which is not possible in the way described in the commands), or they do not correctly interpret the outcome of running the commands.

upvoted 12 times

 **freely** 6 months, 2 weeks ago

I mean without specifying the catalog and the schema in a unity catalog context ? this will only succeed if the table is in the default catalog and schema

upvoted 1 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

 **benni\_ale** 9 months ago



**Selected Answer: E**

E , the collect method outputs strings so the python variable will be a list of string which should not be called as a spark table as in cmd 2

upvoted 1 times

  **imatheushenrique** 1 year ago

E. Cmd 1 will succeed and Cmd 2 will fail. countries\_af will be a Python variable containing a list of strings.  
upvoted 1 times

  **juliom6** 1 year, 2 months ago

**Selected Answer: E**

E is correct.

```
%sql
```

```
create table geo_lookup (continent varchar(2), country varchar(15));
```

```
insert into geo_lookup (continent, country) values
```

```
('AF','Nigeria'),
```

```
('AF','Kenya');
```

```
create table sales (city varchar(15), continent varchar(2));
```

```
insert into sales (city, continent) values
```

```
('Nigeria','AF'),
```

```
('Kenya','AF');
```

```
%python
```

```
countries_af = [x[0] for x in spark.table('geo_lookup').filter("continent='AF'").select('country').collect()]
```

```
%sql
```

```
create view sales_af as
```

```
select *
```

```
from sales
```


```
where city in countries_af
```

```
and continent = "AF";
```

ParseException: [PARSE\_SYNTAX\_ERROR] Syntax error at or near 'in'.(line 4, pos 11)

i.e. countries\_af is a python list of strings and can't be used inside a sql statement

upvoted 3 times

  **AndreFR** 10 months, 2 weeks ago

```
%python
```

```
print(countries_af)
```

```
type(countries_af)
```

upvoted 1 times

  **leopedroso1** 1 year, 4 months ago

By simulating this code in databricks we can see an error being thrown in the SQL statement

ParseException:

[PARSE\_SYNTAX\_ERROR] Syntax error at or near 'IN'.(line 1, pos 38)

```
== SQL ==
```

```
SELECT * FROM backup.sales WHERE CITY IN countries_af AND CONTINENT = "AF"
```

upvoted 1 times

  **RiktRikt007** 1 year, 4 months ago

**Selected Answer: B**

B shows the actual flow of spark sql, where E shows the question context, i mean from databricks point of view E never looked, it's true that question state that database has no other tables, so ?? that mean databricks will not check for that particular table ? it will right ? i also confused by "database has no other database statement" and E and B both are right, but again B state "if countries table exists then command 2 will run" here "if" used, but question want to describe the language interoperability, so most of us selected E

upvoted 1 times

  **benni\_ale** 9 months, 1 week ago

how could it succed if all people tested sql parse syntax error?

upvoted 1 times

🗨️ 👤 **PrashantTiwari** 1 year, 4 months ago

E is correct

upvoted 2 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: E**

vote for E

upvoted 2 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

E is correct answer

upvoted 1 times

🗨️ 👤 **ismoshkov** 1 year, 7 months ago

**Selected Answer: B**

<https://docs.databricks.com/en/notebooks/notebooks-code.html#mix-languages>

Variables defined in one language (and hence in the REPL for that language) are not available in the REPL of another language

upvoted 1 times

🗨️ 👤 **Naveenkm** 1 year, 7 months ago

It is mentioned there exists only 2 objects in database. so B is not an option

upvoted 1 times

🗨️ 👤 **Karen1232123** 1 year, 7 months ago

even if it exists, a table or a view won't work in cmd 2

upvoted 2 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: E**

correct

upvoted 1 times

🗨️ 👤 **lucasasterio** 1 year, 9 months ago

**Selected Answer: E**

correct

upvoted 2 times

🗨️ 👤 **Eertyy** 1 year, 10 months ago

E is right nswer

upvoted 2 times

A Delta table of weather records is partitioned by date and has the below schema: date DATE, device\_id INT, temp FLOAT, latitude FLOAT, longitude FLOAT

To find all the records from within the Arctic Circle, you execute a query with the below filter: latitude > 66.3

Which statement describes how the Delta engine identifies which files to load?

- A. All records are cached to an operational database and then the filter is applied
- B. The Parquet file footers are scanned for min and max statistics for the latitude column
- C. All records are cached to attached storage and then the filter is applied
- D. The Delta log is scanned for min and max statistics for the latitude column
- E. The Hive metastore is scanned for min and max statistics for the latitude column

**Suggested Answer: B**

Community vote distribution

D (100%)

🗳️ 👤 **taif12340** Highly Voted 1 year, 10 months ago

Answer D:

In the Transaction log, Delta Lake captures statistics for each data file of the table. These statistics indicate per file:

- Total number of records
- Minimum value in each column of the first 32 columns of the table
- Maximum value in each column of the first 32 columns of the table
- Null value counts for in each column of the first 32 columns of the table

When a query with a selective filter is executed against the table, the query optimizer uses these statistics to generate the query result. it leverages them to identify data files that may contain records matching the conditional filter.

For the SELECT query in the question, The transaction log is scanned for min and max statistics for the price column

upvoted 22 times

🗳️ 👤 **7afe201** Most Recent 1 week, 4 days ago

Selected Answer: D

Since the table is not partitioned on Latitude Delta log is used. If latitude was a partition column then Parquet footers would come into play.

upvoted 1 times

🗳️ 👤 **Gaurav207** 1 month, 2 weeks ago

Selected Answer: D

In the Transaction log, Delta Lake captures statistics for each data file of the table. These statistics indicate per file:

- Total number of records
- Minimum value in each column of the first 32 columns of the table
- Maximum value in each column of the first 32 columns of the table
- Null value counts for in each column of the first 32 columns of the table

upvoted 1 times

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

Selected Answer: D

suretly D

upvoted 1 times

🗳️ 👤 **JoG1221** 2 months, 1 week ago

Selected Answer: D

elta extracts stats from the Parquet footers at write time and stores them in \_delta\_log.

At query time, Delta reads the stats from the log instead of scanning file footers = faster performance. This is what enables efficient data skipping and query optimization.

upvoted 1 times

🗳️ 👤 **johnserafim** 3 months, 3 weeks ago

**Selected Answer: B**

B is correct!

Delta Lake stores min/max statistics for each column in the Parquet file footers. The engine scans these footers to determine if a file contains any data that satisfies the latitude > 66.3 condition. If the minimum latitude in a file is greater than 66.3, the file is loaded. If the maximum latitude is less than or equal to 66.3, the file is skipped.

upvoted 3 times

🗳️ 👤 **akashdesarda** 9 months ago

**Selected Answer: D**

Above mentioned points are correct. If the table was just parquet table then parquet file footer have been used. But since this is Delta table, then delta log is used to scan & skip files. It uses stats written in in transaction log.

upvoted 3 times

🗳️ 👤 **AndreFR** 10 months, 2 weeks ago

Answer D :

Delta data skipping automatically collects the stats (min, max, etc.) for the first 32 columns for each underlying Parquet file when you write data into a Delta table. Databricks takes advantage of this information (minimum and maximum values) at query time to skip unnecessary files in order to speed up the queries.

<https://www.databricks.com/discover/pages/optimize-data-workloads-guide#delta-data>

upvoted 2 times

🗳️ 👤 **saravanan289** 10 months, 2 weeks ago

**Selected Answer: D**

Delta table stores file statistics in transaction log

upvoted 2 times

🗳️ 👤 **03355a2** 1 year ago

**Selected Answer: D**

No explanation needed, this is where the information is stored.

upvoted 2 times

🗳️ 👤 **imatheushenrique** 1 year ago

D. The Delta log is scanned for min and max statistics for the latitude column

upvoted 1 times

🗳️ 👤 **coercion** 1 year, 1 month ago

**Selected Answer: D**

Delta log collects statistics like min value, max value, no of records, no of files for each transaction that happens on the table for the first 32 columns (default value)

upvoted 1 times

🗳️ 👤 **Tayari** 1 year, 2 months ago

**Selected Answer: D**

D is the answer

upvoted 1 times

🗳️ 👤 **arik90** 1 year, 3 months ago

**Selected Answer: D**

Based on Docu is D I don't know why here is showing B

upvoted 1 times

🗳️ 👤 **alexvno** 1 year, 3 months ago

**Selected Answer: D**

Delta log first

upvoted 1 times



🗳️ 👤 **DavidRou** 1 year, 3 months ago

**Selected Answer: D**

Statistics on first 32 columns of a table are computed and written in the Delta Log by default.

upvoted 1 times



  **vikram12apr** 1 year, 4 months ago

**Selected Answer: D**

D is the right answer

upvoted 1 times

The data engineering team has configured a job to process customer requests to be forgotten (have their data deleted). All user data that needs to be deleted is stored in Delta Lake tables using default table settings.

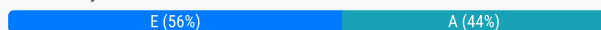
The team has decided to process all deletions from the previous week as a batch job at 1am each Sunday. The total duration of this job is less than one hour. Every Monday at 3am, a batch job executes a series of VACUUM commands on all Delta Lake tables throughout the organization. The compliance officer has recently learned about Delta Lake's time travel functionality. They are concerned that this might allow continued access to deleted data.

Assuming all delete logic is correctly implemented, which statement correctly addresses this concern?

- A. Because the VACUUM command permanently deletes all files containing deleted records, deleted records may be accessible with time travel for around 24 hours.
- B. Because the default data retention threshold is 24 hours, data files containing deleted records will be retained until the VACUUM job is run the following day.
- C. Because Delta Lake time travel provides full access to the entire history of a table, deleted records can always be recreated by users with full admin privileges.
- D. Because Delta Lake's delete statements have ACID guarantees, deleted records will be permanently purged from all storage systems as soon as a delete job completes.
- E. Because the default data retention threshold is 7 days, data files containing deleted records will be retained until the VACUUM job is run 8 days later.

**Suggested Answer: A**

Community vote distribution



**asmayassineg** Highly Voted 1 year, 11 months ago

Answer is E, default retention period is 7 days <https://learn.microsoft.com/en-us/azure/databricks/delta/vacuum>  
upvoted 21 times

**mardigras** Highly Voted 1 year, 4 months ago

**Selected Answer: A**

The answer has to be A.

The deletion is done on Sunday 1am and then the next day Monday 3am, VACUUM was initiated, so one can only time travel for about 24 hours.  
upvoted 12 times

**csrazdan** 9 months, 3 weeks ago

The default retention threshold for time travel is 7 days. VACUUM which is executed on Monday 3 am will remove history for changes where time travel has expired for previous 7 days.  
upvoted 9 times

**benni\_ale** 8 months, 2 weeks ago

Exactly!

upvoted 2 times

**paluskapter** Most Recent 3 weeks, 3 days ago

**Selected Answer: E**

Because of the retention period of 7days, vacuum won't delete it the next day.  
upvoted 1 times

**KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

**Deep92** 2 months, 3 weeks ago

**Selected Answer: E**

Team configured weekly deletion and vacuum will delete weekly data. So this is correct for me.  
upvoted 1 times

🗨️ 👤 **Tedet** 4 months ago

Selected Answer: E

Read question carefully: "all deletions from the previous week", "Every Monday at 3am, a batch job executes a series of VACUUM commands"

Logically 8th day.

Apply basics: Default retention threshold for VACUUM is 7 days beyond which on running VACUUM data will be purged.

upvoted 1 times

🗨️ 👤 **fabiospont** 4 months, 3 weeks ago

Selected Answer: E

E is correct, because the VACUUM retention is of 168h or 7 days, after statments of deletions.

upvoted 1 times

🗨️ 👤 **arekm** 6 months ago

Selected Answer: E

Saturday delete puts the deleted records in the transaction log. The retention clock starts ticking. Since the default for keeping the history is 168 hours (7 days), by no means the following Monday vacuum removes since the clock did not reach 168 hours (7 days) - it is at hour 26.

upvoted 2 times

🗨️ 👤 **Anithec0der** 6 months, 3 weeks ago

Selected Answer: E

I was also thinking in the same way that data will be deleted immediate after the vaccum command is run but it actually logically deletes the data and not physically till the 7 day from the ask of vaccum command. so E is perfect.

upvoted 1 times

🗨️ 👤 **AlejandroU** 6 months, 3 weeks ago

Selected Answer: E

Answer E. The retention period for time travel queries in Delta Lake is controlled by a 7-day default, not 24 hours. Hence, the statement (Option A) that deleted records may be accessible for around 24 hours is incorrect in the context of Delta Lake's default retention period.

upvoted 1 times

🗨️ 👤 **benni\_ale** 8 months, 2 weeks ago

Selected Answer: E

Default retention period is 7 days so the vacuum command won't delete the files corresponding to deleted rows at Sunday 1 am but the ones of the previous week instead.

upvoted 1 times

🗨️ 👤 **tangerine141** 9 months ago

Selected Answer: E

Delta Lake's default retention threshold for old data files (which allows time travel) is 7 days. This means that even after records are deleted, the files that previously contained those records are kept for 7 days before they are eligible for permanent deletion by the VACUUM command.

The VACUUM command is responsible for permanently deleting the old data files after the retention period. Since the job runs every Monday, this means that data deleted during the previous week will not be fully purged until after the retention period has passed (which would be 8 days after the deletion, considering the weekly processing).

upvoted 2 times

🗨️ 👤 **akashdesarda** 9 months ago

Selected Answer: E

Delete job is running as batch job for all requests made current week on Sunday & Vacuum is ran next day . Since there is no mention of change is retention period then it is 7 days. Vacuum will delete data older than 7 days, i.e. it will delete data of previous week & not current week. Current weeks data will be removed in next week's vacuum job.

upvoted 1 times

🗨️ 👤 **fe3b2fc** 10 months, 2 weeks ago

Selected Answer: E

From the documentation.

"The default retention threshold for data files after running VACUUM is 7 days."

It doesn't matter if VACUUM is ran the following day, the retention period on a default setup is still 7 days after they do the VACUUM on Monday.

upvoted 3 times

🗨️ 👤 **03355a2** 1 year ago

Selected Answer: A

They expect the deleted records for the previous week to be deleted Sunday from 1am to 2am. Then the next day(Monday) at 3am approx 24hrs later, the vacuum command is ran. This means the records from the previous week are only around for 24ish hours before they are removed with the

vacuum command. They aren't waiting 8 days to run the command, there fore E is wrong.

upvoted 3 times

🗨️ 👤 **akashdesarda** 9 months ago

This week's vacuum will remove data of the previous week's delete command since default retention has not changed.

upvoted 2 times

🗨️ 👤 **imatheushenrique** 1 year ago

E. Because the default data retention threshold is 7 days, data files containing deleted records will be retained until the VACUUM job is run 8 days later.

upvoted 1 times

🗨️ 👤 **coercion** 1 year, 1 month ago

**Selected Answer: E**

Default retention period is 7 days so newly deleted data on Sunday will be available for next 7 days (even if vacuum was run on Monday as it will delete 7 days old data and not the data that was loaded yesterday "Sunday" )

upvoted 1 times

A junior data engineer has configured a workload that posts the following JSON to the Databricks REST API endpoint 2.0/jobs/create.

```
{
  "name": "Ingest new data",
  "existing_cluster_id": "6015-954420-peace720",
  "notebook_task": {
    "notebook_path": "/Prod/ingest.py"
  }
}
```


Assuming that all configurations and referenced resources are available, which statement describes the result of executing this workload three times?

- A. Three new jobs named "Ingest new data" will be defined in the workspace, and they will each run once daily.
- B. The logic defined in the referenced notebook will be executed three times on new clusters with the configurations of the provided cluster ID.
- C. Three new jobs named "Ingest new data" will be defined in the workspace, but no jobs will be executed.
- D. One new job named "Ingest new data" will be defined in the workspace, but it will not be executed.
- E. The logic defined in the referenced notebook will be executed three times on the referenced existing all purpose cluster.

**Suggested Answer: C**


Community vote distribution

C (100%)

 **arekm** Highly Voted 6 months ago

**Selected Answer: C**

You can totally create 3 jobs with the same name using the UI. REST API is no different. Since no schedule information is in the json, it will not run.  
upvoted 5 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: C**

suretly C  
upvoted 1 times

 **AlejandroU** 6 months, 3 weeks ago

**Selected Answer: C**

Answer C. 3 new jobs with the same name will be created with each API call, but they won't be executed unless further configuration is made for scheduling or triggering.  
upvoted 1 times

 **akashdesarda** 9 months ago

**Selected Answer: C**

Databricks will keep on creating new jobs if you keep running create rept api. Each will have the same name but a different ID. Also no trigger/schedule is mentioned so they wont run.  
upvoted 1 times

 **imatheushenrique** 1 year ago

C. Three new jobs named "Ingest new data" will be defined in the workspace, but no jobs will be executed.  
upvoted 1 times

 **coercion** 1 year, 1 month ago

**Selected Answer: C**

Learnt new thing : DBX can have duplicate job names (Job ID will be different). So three jobs will be created with three job ids but it will not run as no schedule is mentioned.  
upvoted 3 times

 **franciscodm** 1 year, 3 months ago

C for sure  
upvoted 1 times

 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: C**

Correct answer is C

upvoted 1 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: C**

Correct answer is C

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: C**

databricks jobs create will create a new job with the same name each time it is run.

In order to overwrite the extsting job you need to run databricks jobs reset

upvoted 3 times

🗨️ 👤 **bob\_** 1 year, 9 months ago

Answer is correct. The 3 API calls create 3 jobs with the same name but different job ids. There is no schedule defined so will not execute.

upvoted 3 times

🗨️ 👤 **Eertyy** 1 year, 9 months ago

correct answer is A, because an api can create can create same job with same name if executed thrice

upvoted 2 times

🗨️ 👤 **Eertyy** 1 year, 9 months ago

to add more: when you execute this workload three times, it will define three new jobs in the workspace, each with the name "Ingest new data."

These jobs can be scheduled to run daily or at a specified frequency, depending on how they are configured.

upvoted 2 times

🗨️ 👤 **Starvosxant** 1 year, 8 months ago

Ok. So Tell me the schedule these Jobs will run?

Dont know? Why? Maybe because it is not specified, or even configured. So the answer is correct. Create 3 Jobs but none will be executed.

upvoted 2 times

🗨️ 👤 **mwyopme** 1 year, 9 months ago

therefore answer: D

upvoted 2 times

🗨️ 👤 **mwyopme** 1 year, 9 months ago

only one job, why 3? - because there 3 lines of JSON??

answer should be:

One new job named "Ingest new data" will be defined in the workspace, but it will not be executed.

upvoted 1 times

🗨️ 👤 **vsydoriak99** 1 year, 9 months ago

Because the create command was run 3 times. Databricks can have several jobs with the same name

upvoted 3 times

An upstream system is emitting change data capture (CDC) logs that are being written to a cloud object storage directory. Each record in the log indicates the change type (insert, update, or delete) and the values for each field after the change. The source table has a primary key identified by the field `pk_id`.

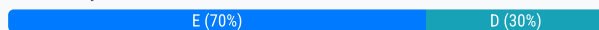
For auditing purposes, the data governance team wishes to maintain a full record of all values that have ever been valid in the source system. For analytical purposes, only the most recent value for each record needs to be recorded. The Databricks job to ingest these records occurs once per hour, but each individual record may have changed multiple times over the course of an hour.

Which solution meets these requirements?

- A. Create a separate history table for each `pk_id` resolve the current state of the table by running a union all filtering the history tables for the most recent state.
- B. Use `MERGE INTO` to insert, update, or delete the most recent entry for each `pk_id` into a bronze table, then propagate all changes throughout the system.
- C. Iterate through an ordered set of changes to the table, applying each in turn; rely on Delta Lake's versioning ability to create an audit log.
- D. Use Delta Lake's change data feed to automatically process CDC data from an external system, propagating all changes to all dependent tables in the Lakehouse.
- E. Ingest all log information into a bronze table; use `MERGE INTO` to insert, update, or delete the most recent entry for each `pk_id` into a silver table to recreate the current table state.

**Suggested Answer: E**

Community vote distribution



**Eertyy** Highly Voted 1 year, 10 months ago

The answer given is correct  
upvoted 10 times

**fabiospont** 4 months, 3 weeks ago

Option D  
upvoted 2 times

**Eertyy** 1 year, 9 months ago

I want to correct my response. It seems the right answer Option D, it leverages Delta Lake's built-in capabilities for handling CDC data. It is designed to efficiently capture, process, and propagate changes, making it a more robust and scalable solution, particularly for large-scale data scenarios with frequent updates and auditing requirements.  
upvoted 3 times

**arekm** 6 months ago

The question names 2 requirements to keep the data

- archival with all records
- querying with only the currently valid values

CDF is not designed as a permanent storage for archival purposes. It keeps the data to propagate it to downstream applications / workloads. CDF is also purged with the vacuum command, so this would make a very unreliable archival.

Medallion architecture that Databricks promotes seems to be a clear winner.  
upvoted 1 times

**Sriramiyer92** 6 months ago

Actually you were correct in the first go. If you are dealing with small amount of data to changes, CDC is it way to go. But not in this case. (Keywords: For auditing purposes, the data governance team wishes to maintain a full record of all values that have ever been valid in the source system.) Answer is E in that case.  
upvoted 1 times

**Starvosxant** 1 year, 8 months ago

Databricks is NOT able to process CDC alone. It needs a intermediare Tool to make it on an object storage and then ingest it.  
So how can be D?

upvoted 5 times

  **imatheushenrique** Highly Voted 1 year ago

E.

This is the correct answer because it meets the requirements of maintaining a full record of all values that have ever been valid in the source system and recreating the current table state with only the most recent value for each record. The code ingests all log information into a bronze table, which preserves the raw CDC data as it is. Then, it uses merge into to perform an upsert operation on a silver table, which means it will insert new records or update or delete existing records based on the change type and the pk\_id columns. This way, the silver table will always reflect the current state of the source table, while the bronze table will keep the history of all changes.

upvoted 6 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

Selected Answer: E

Surely E

upvoted 1 times

  **Tedet** 4 months ago



Selected Answer: E

Bronze Table (Raw Ingest): You start by ingesting all the change data capture (CDC) records into a bronze table.

Silver Table (Processed State): The silver table represents the most recent state of the data. You would use the MERGE INTO command to process the changes from the bronze table and update the silver table accordingly.

Audit Trail: Since you're ingesting all the data into the bronze table, you maintain a full history of changes that have occurred over time, which satisfies the auditing requirement.

upvoted 2 times

  **fabiospont** 4 months, 3 weeks ago

D is the option more indicated. CDC is the most efficiently.

upvoted 1 times

  **mwynn** 5 months, 3 weeks ago

Selected Answer: E

If getting External CDC Data (Kafka, etc) no need for CDF! Just ingest to Bronze with (pipelines.reset.allowed = false)

upvoted 1 times

  **arekm** 6 months ago

Selected Answer: E

The question names 2 requirements to keep the data

- archival with all records

- querying with only the currently valid values

CDF is not designed as a permanent storage for archival purposes. It keeps the data to propagate it to downstream applications / workloads. CDF is also purged with the vacuum command, so this would make a very unreliable archival.

Medallion architecture that Databricks promotes seems to be a clear winner.

upvoted 1 times

  **arekm** 6 months ago

On top of that - CDC with CDF is not automatic. You still need SQL or Python to read the changes and put them somewhere.

upvoted 1 times

  **benni\_ale** 7 months, 2 weeks ago

Selected Answer: E


You can only read the change data feed for enabled tables. You must explicitly enable the change data feed option using one of the following methods: TBLPROPERTIES (delta.enableChangeDataFeed = true) . this means it is a delta feature or in other words it is a feature supported by delta tables. the data to process in the question is external so it is not a delta table => can't be B... Hopefully I am correct.

upvoted 1 times

  **benni\_ale** 7 months, 2 weeks ago

i meant can't be D

upvoted 1 times

  **benni\_ale** 8 months, 2 weeks ago

Selected Answer: E

E . databricks cdc is not set to process external cdc. if u have external cdc u could send to bronze for auditing purposes and use bronze to get silver where u have only valid records



upvoted 1 times

🗨️ 👤 **databrick\_work** 9 months, 2 weeks ago

E is correct

upvoted 1 times

🗨️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: E**

The answer is E

upvoted 2 times

🗨️ 👤 **RafaelCFC** 1 year, 5 months ago

**Selected Answer: E**

Complimenting kz\_data's response, be aware that the data that is being consumed is not a Databrick's CDC data feed object, but rather, CDC coming from somewhere else, that is, just regular data. So, indeed, it can't be processed without another tool.

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

Answer E is correct, as the CDC captured from the external database may contain duplicates for the same pk\_id (key) due to multiple updates within the processed hour, we need to take the most recent update for the pk\_id, and then MERGE into a silver table.

upvoted 2 times

🗨️ 👤 **a560fe1** 1 year, 5 months ago

CDF captures changes only from a Delta table and is only forward-looking once enabled. The CDC logs are writing to object storage. So you would need to ingest those and merge into downstream tables, hence the answer is E

upvoted 2 times

🗨️ 👤 **hamzaKhribi** 1 year, 7 months ago

**Selected Answer: D**

For me the answer is D, the question states that CDC logs are emitted on an external storage meaning it can be ingested into the bronze layer on a table with CDF enabled. In this case we let databricks handle the complexity of following changes and only worry about data quality. meaning with CDF enabled databricks will already work the audit data for us with the table\_changes of the pre-image and post-image and also give us the last updated value for our use case.

here is a similar example: <https://www.databricks.com/blog/2021/06/09/how-to-simplify-cdc-with-delta-lakes-change-data-feed.html>

upvoted 3 times

🗨️ 👤 **spaceexplorer** 1 year, 5 months ago

This article shows exactly why D is not right. Since "CDF captures changes only from a Delta table and is only forward-looking once enabled."

upvoted 4 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: E**

E is correct

upvoted 3 times

An hourly batch job is configured to ingest data files from a cloud object storage container where each batch represent all records produced by the source system in a given hour. The batch job to process these records into the Lakehouse is sufficiently delayed to ensure no late-arriving data is missed. The user\_id field represents a unique key for the data, which has the following schema: user\_id BIGINT, username STRING, user\_utc STRING, user\_region STRING, last\_login BIGINT, auto\_pay BOOLEAN, last\_updated BIGINT

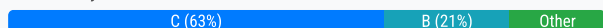
New records are all ingested into a table named account\_history which maintains a full record of all data in the same schema as the source. The next table in the system is named account\_current and is implemented as a Type 1 table representing the most recent value for each unique user\_id.

Assuming there are millions of user accounts and tens of thousands of records processed hourly, which implementation can be used to efficiently update the described account\_current table as part of each hourly batch job?

- A. Use Auto Loader to subscribe to new files in the account\_history directory; configure a Structured Streaming trigger once job to batch update newly detected files into the account\_current table.
- B. Overwrite the account\_current table with each batch using the results of a query against the account\_history table grouping by user\_id and filtering for the max value of last\_updated.
- C. Filter records in account\_history using the last\_updated field and the most recent hour processed, as well as the max last\_login by user\_id write a merge statement to update or insert the most recent value for each user\_id.
- D. Use Delta Lake version history to get the difference between the latest version of account\_history and one version prior, then write these records to account\_current.
- E. Filter records in account\_history using the last\_updated field and the most recent hour processed, making sure to deduplicate on username; write a merge statement to update or insert the most recent value for each username.

**Suggested Answer: C**

Community vote distribution



🗳️ 👤 **RafaelCFC** Highly Voted 1 year, 5 months ago

**Selected Answer: C**

My reasoning is thus:

The application is based on batch processes, so A is wrong.

Overwriting the table would destroy the Type 1 SCD behavior, so B is wrong.

Comparing versions of account\_history would not be efficient, as the whole data would be scanned, so D is wrong. 'username' is not a key column, so we have no guarantee that it's unique, thus de-duplicating by it can yield wrongly grouped sets of rows, so E is not a safe bet, with the information we know.

C is the best option.

upvoted 11 times

🗳️ 👤 **terrku** Highly Voted 1 year, 3 months ago

**Selected Answer: B**

Type 1 table means the behavior is overwriting.

upvoted 9 times

🗳️ 👤 **thotwielder** Most Recent 2 weeks, 4 days ago

**Selected Answer: B**

c: wrong. it is using last\_login, but the key is user\_id, no reason to group by last\_login.

b: correct. group by key (user\_id), and get the latest record. overwrite is very efficient.

upvoted 1 times

🗳️ 👤 **thotwielder** 2 weeks, 4 days ago

update. to clarify c is wrong not because group by last\_login. c is wrong because merge is not as efficient as overwrite because there are millions of user which means the account\_current table is huge and merge needs to compare source and target table which is time consuming compared with simple overwrite.

upvoted 1 times

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

suretly C

upvoted 1 times

🗨️ 👤 **JoG1221** 2 months, 1 week ago

**Selected Answer: C**

This approach is:

Efficient: Only processes incremental data, not the full table

Accurate: Ensures latest record per user\_id

Scalable: Works well with millions of users and hourly ingestion

upvoted 1 times

🗨️ 👤 **Tedet** 4 months ago

**Selected Answer: C**

This option proposes filtering the records in the account\_history table for the most recent records based on the last\_updated field, which is exactly what you want to do to get the most recent value.

The use of a MERGE statement ensures that only the most recent records are inserted or updated in the account\_current table.

This method avoids full overwrites of the account\_current table and only updates records that have actually changed, which is efficient for large datasets.

Conclusion: This is the most efficient approach because it ensures only the most recent data is merged, and it avoids unnecessary full table rebuilds.

upvoted 1 times

🗨️ 👤 **Ananth4Sap** 8 months, 1 week ago

C is correct because

B is wrong as it says filtering the max value of last updated + overwriting we will miss some valid records.

two valid scenarios: 1-Filtering the max value of last updated +merging (Option-c)

2.use window function on last update, filter and then overwrite (no options)

upvoted 3 times

🗨️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: C**

A. NO. Batch job required so AutoLoader and StructuredStreaming unnecessarily complex solutions.

B. NO. A full overwrite of the table is not efficient.

C. YES. Seems it is filtering and merging on the id by using as less data as reasonable in the merge statement, why not?

D. NO. Difference operation is very ineffecient for this purpose

E. NO. Username is not key

upvoted 3 times

🗨️ 👤 **Dhusanth** 10 months, 4 weeks ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗨️ 👤 **faraaz132** 11 months ago

C is correct because:

A record might have multiple changes and we need to select the most recent change that happened on that record. For that we will use max Log in date and rank it using window function, then we filter on rank=1 and use it for UPSERT operation.

upvoted 1 times

🗨️ 👤 **Karunakaran\_R** 1 year ago

I think B ,Type 1 table must overwrite the data

upvoted 1 times

🗨️ 👤 **Freyr** 1 year, 1 month ago

C is correct.

A Type 1 table means that it performs an "upsert" operation without maintaining history, based on the merge condition. This means that new records are inserted, and existing records are updated. As a result, the merge process does not retain historical records.

Therefore, the correct answer is C.

upvoted 3 times

🗨️ 👤 **PrashantTiwari** 1 year, 4 months ago

C is correct

upvoted 1 times

🗨️ 👤 **DAN\_H** 1 year, 5 months ago

**Selected Answer: C**

answer is C

upvoted 2 times

🗨️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: C**

answer is C

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: C**

Correct answer is C

upvoted 1 times

🗨️ 👤 **ATLTennis** 1 year, 5 months ago

**Selected Answer: D**

D is the most optimal way to identify the changes in the last data refresh

upvoted 1 times

🗨️ 👤 **AndreFR** 10 months, 2 weeks ago

doesn't work because it doesn't take into account requirement on user\_id

upvoted 2 times

A table in the Lakehouse named `customer_churn_params` is used in churn prediction by the machine learning team. The table contains information about customers derived from a number of upstream sources. Currently, the data engineering team populates this table nightly by overwriting the table with the current valid values derived from upstream data sources.

The churn prediction model used by the ML team is fairly stable in production. The team is only interested in making predictions on records that have changed in the past 24 hours.

Which approach would simplify the identification of these changed records?

- A. Apply the churn model to all rows in the `customer_churn_params` table, but implement logic to perform an upsert into the predictions table that ignores rows where predictions have not changed.
- B. Convert the batch job to a Structured Streaming job using the complete output mode; configure a Structured Streaming job to read from the `customer_churn_params` table and incrementally predict against the churn model.
- C. Calculate the difference between the previous model predictions and the current `customer_churn_params` on a key identifying unique customers before making new predictions; only make predictions on those customers not in the previous predictions.
- D. Modify the overwrite logic to include a field populated by calling `spark.sql.functions.current_timestamp()` as data are being written; use this field to identify records written on a particular date.
- E. Replace the current overwrite logic with a merge statement to modify only those records that have changed; write logic to make predictions on the changed records identified by the change data feed.



**Suggested Answer: E**

Community vote distribution



E (100%)

  **Eertyy** Highly Voted 1 year, 10 months ago

E is right answer  
upvoted 6 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**  
suretly E  
upvoted 1 times

  **JoG1221** 2 months, 1 week ago

**Selected Answer: E**  
Option E aligns with Delta Lake best practices:  
Efficient updates using MERGE  
Precise tracking with Change Data Feed  
Streamlined ML inference on only updated records  
upvoted 1 times

  **AlHerd** 3 months ago

**Selected Answer: E**  
E.

While both D and E look right D only adds a timestamp but doesn't track whether the record content actually changed, leading to false positives.  
upvoted 1 times

  **Tedet** 4 months ago

**Selected Answer: D**  
Evaluation:

Adding a `current_timestamp()` field to each record during the overwrite allows you to track when each record was written.

This makes it easy to identify records that have been updated or inserted recently by filtering on this timestamp field (e.g., filtering for records written in the past 24 hours).

This approach simplifies identifying recently changed records because you can easily filter for the most recent data and then run churn predictions only on those records.

Conclusion: This is a simple and efficient solution. It allows you to track changes by using a timestamp, making it easy to filter and predict only on changed records without complex logic.

upvoted 1 times

🗨️ 👤 **arekm** 6 months ago

**Selected Answer: E**

A, B, and C don't make sense. Adding a timestamp with an overwrite logic that overwrites everything does not make sense - all records would have a timestamp from the last night. That would be not helpful in identifying what changed.

E is correct. Only write changes, use CDF to identify the changes and apply the model.

upvoted 2 times

🗨️ 👤 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: E**

While both E and D are correct.

E is more accurate, given the scenario

upvoted 1 times

🗨️ 👤 **janeZ** 6 months, 3 weeks ago

**Selected Answer: D**

D is the right answer

upvoted 1 times

🗨️ 👤 **Melik3** 10 months, 4 weeks ago

I don't understand why E is correct. With E we are updating only data needed but we are then doing prediction on the whole table which means that we are doing again predictions on not changing records which is not efficient

upvoted 1 times

🗨️ 👤 **Tedet** 4 months ago

You are 100pc correct Melik3. Reason being consequences of E are below.

A merge statement ensures that only the records that have changed are updated, but it doesn't directly address how to identify which records have changed within the last 24 hours.

Using a change data feed can help track changes, but it may not be the most efficient method unless the infrastructure is set up for real-time change tracking.

The complexity of managing and using the change data feed for just 24-hour changes might introduce unnecessary overhead.

Conclusion: This is a good option, but it could be more complex to implement than simply adding a `current_timestamp()` field.

upvoted 1 times

🗨️ 👤 **benni\_ale** 8 months, 1 week ago

"write logic to make predictions on the CHANGED records identified by the change data feed". the only thing partially wrong about E is that it has never been stated that the table has a change data feed enables.

upvoted 1 times

🗨️ 👤 **leopedroso1** 1 year, 4 months ago

E is the correct one. By removing overwrite with merge, this will lead to an UPSERT causing updating only the data needed ( When Matched Update + When not mached insert clauses). Then, with the CDC the capability of identifying is also satisfied.

upvoted 2 times

🗨️ 👤 **AziLa** 1 year, 5 months ago

correct ans is E

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

E is correct

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: E**

E is Correct

upvoted 3 times

A table is registered with the following code:

```
CREATE TABLE recent_orders AS (
  SELECT a.user_id, a.email, b.order_id, b.order_date
  FROM
    (SELECT user_id, email
     FROM users) a
  INNER JOIN
    (SELECT user_id, order_id, order_date
     FROM orders
     WHERE order_date >= (current_date() - 7)) b
  ON a.user_id = b.user_id
)
```

Both users and orders are Delta Lake tables. Which statement describes the results of querying recent\_orders?

- A. All logic will execute at query time and return the result of joining the valid versions of the source tables at the time the query finishes.
- B. All logic will execute when the table is defined and store the result of joining tables to the DBFS; this stored data will be returned when the table is queried.
- C. Results will be computed and cached when the table is defined; these cached results will incrementally update as new records are inserted into source tables.
- D. All logic will execute at query time and return the result of joining the valid versions of the source tables at the time the query began.
- E. The versions of each source table will be stored in the table transaction log; query results will be saved to DBFS with each query.

**Suggested Answer: D**

Community vote distribution

B (81%)

D (19%)

**asmayassineg** Highly Voted 1 year, 11 months ago

Correct answer is B. table is created and data of join will be stored on DBFS and it will be returned on query time  
upvoted 19 times

**practitioner** Highly Voted 10 months, 2 weeks ago

**Selected Answer: D**

The question says "Which statement describes the results of querying recent\_orders?"

The question doesn't ask about the code snippet itself. This question is about the logic of "select \* from recent\_orders" after the creation of recent\_orders.

answer is D

D is the right answer  
upvoted 19 times

**fe3b2fc** 10 months, 1 week ago

Not sure how so many people misunderstood the actual question. It already says at the top that the table is registered as the code given, they're not executing the code again.  
upvoted 3 times

**Onobhas01** 10 months ago

Why are people so fixed on how the table was created, question says what happens when a query is run against the table.  
upvoted 2 times

**HairyTorso** 6 months ago

As Benni\_ale said - CTAS can be used to create a snapshot of a table. The data will be stored and not updated further, unless table is explicitly updated.  
upvoted 1 times

**benni\_ale** 8 months, 1 week ago

option D mentions "return the result of joining the valid versions of the source tables" but that's not true. when u interrogate the table resulting from a join of two tables u are not re-performing joins operations at query time anymore an the version is the one from the last time the CTAS statement was executed

upvoted 5 times

🗨️ **⚡ Vitality** Most Recent 3 weeks, 1 day ago

**Selected Answer: D**

Definitely D, c'mon guys.

upvoted 1 times

🗨️ **👤 c315d10** 4 weeks ago

**Selected Answer: D**

d is correct

upvoted 1 times

🗨️ **👤 KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

suretly B

upvoted 1 times

🗨️ **👤 JoG1221** 2 months, 1 week ago

**Selected Answer: B**

It materializes the result and saves the data into a Delta Lake table stored on DBFS (or other compatible storage).

upvoted 1 times

🗨️ **👤 lakime** 3 months, 1 week ago

**Selected Answer: D**

It's CTAAS, nothing is being stored in DBFS...

upvoted 1 times

🗨️ **👤 Tedet** 4 months ago

**Selected Answer: B**

It's CTAS. So snapshot at the time of creation will be returned.

upvoted 1 times

🗨️ **👤 ptty** 4 months, 1 week ago

**Selected Answer: B**

It's B because this is different from creating a view (which would use CREATE VIEW instead), where the query logic would be executed each time the view is accessed.

upvoted 3 times

🗨️ **👤 arekm** 6 months ago

**Selected Answer: B**

B - it is a table.

Query time answers assume we are talking about a view, which we aren't. Table is not automatically updated whenever the tables used in CTAS change - it is a standalone entity.

upvoted 2 times

🗨️ **👤 Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: B**

The answer is B.

Pls note it is CTAS statement and not a subquery.

upvoted 1 times

🗨️ **👤 benni\_ale** 8 months, 1 week ago

**Selected Answer: B**

B is correct

upvoted 1 times

🗨️ **👤 nedlo** 8 months, 1 week ago

**Selected Answer: B**

Only logic there is inside create statemetn and it will execute once while executing "create table" statement. Further select queries will only select any data that was inserted during create table statement , data wont by updated automatically. So B

upvoted 1 times



🗨️ 👤 **benni\_ale** 8 months, 2 weeks ago

i think B

upvoted 1 times

🗨️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: B**

i picked b

upvoted 1 times

🗨️ 👤 **Isio05** 1 year ago

**Selected Answer: B**

CTAS statements persist it results, so B

upvoted 2 times

🗨️ 👤 **imatheushenrique** 1 year ago

B. All logic will execute when the table is defined and store the result of joining tables to the DBFS; this stored data will be returned when the table is queried.

upvoted 2 times

A production workload incrementally applies updates from an external Change Data Capture feed to a Delta Lake table as an always-on Structured Stream job. When data was initially migrated for this table, OPTIMIZE was executed and most data files were resized to 1 GB. Auto Optimize and Auto Compaction were both turned on for the streaming production job. Recent review of data files shows that most data files are under 64 MB, although each partition in the table contains at least 1 GB of data and the total table size is over 10 TB.



Which of the following likely explains these smaller file sizes?

- A. Databricks has autotuned to a smaller target file size to reduce duration of MERGE operations
- B. Z-order indices calculated on the table are preventing file compaction
- C. Bloom filter indices calculated on the table are preventing file compaction
- D. Databricks has autotuned to a smaller target file size based on the overall size of data in the table
- E. Databricks has autotuned to a smaller target file size based on the amount of data in each partition

**Suggested Answer: A**

Community vote distribution

A (100%)

  **cotardo2077** Highly Voted 1 year, 9 months ago

**Selected Answer: A**

<https://docs.databricks.com/en/delta/tune-file-size.html#autotune-table 'Autotune file size based on workload'>

upvoted 13 times

  **Vitality** 3 weeks, 1 day ago



MERGE operations overall benefit from fewer, larger files, so reducing file size to improve MERGE performance would be counterintuitive. E can be the only correct answer.

upvoted 1 times

  **meatpoof** 5 months ago

Your source doesn't support your answer. It doesn't mention anything about autotuning to increase the speed of merges

upvoted 2 times

  **Vitality** Most Recent 3 weeks, 1 day ago

**Selected Answer: E**

Definitely E. Auto Compaction is tuning file sizes based on the data volume per partition. This helps optimize performance for streaming workloads, where smaller files can reduce latency and improve the efficiency of incremental updates.



upvoted 1 times

  **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

Surely A



upvoted 1 times

  **JoG1221** 2 months, 1 week ago

**Selected Answer: E**

Databricks has autotuned to a smaller target file size based on the amount of data in each partition

upvoted 1 times

  **JoG1221** 2 months, 1 week ago

**Selected Answer: D**

File size tuning is not based on total table size or merge ops, but on partition-level dynamics.

upvoted 1 times

  **JoG1221** 2 months, 1 week ago

Answer is E

upvoted 1 times

  **kishanu** 2 months, 3 weeks ago

**Selected Answer: E**

Databricks Auto Optimize and Auto Compaction features are designed to optimize file sizes dynamically for better performance and efficiency in Delta Lake. These features do not use a fixed target file size like 1 GB, but instead autotune file sizes based on partition-level characteristics.

In this case:

Each partition has at least 1 GB of data, and the overall table is large (10+ TB), but...

You see many small files <64 MB, which seems suboptimal at first.

However, Databricks may intentionally use smaller file sizes within partitions when:

The data change rate is high (as in a streaming CDC feed).

Smaller file sizes help with faster read times, reduced shuffle, and quicker MERGE operations during structured streaming.

The amount of new data added per batch or microbatch is small, leading to many smaller files, especially when auto compaction determines this improves job performance at runtime.

This makes option E the most accurate description of what's happening.

upvoted 2 times

  **AIHerd** 3 months ago

**Selected Answer: A**

An always-on Structured Streaming job that applies updates from a Change Data Capture (CDC) feed uses frequent MERGE operations to apply changes (inserts, updates, deletes) to the Delta table.

Because these MERGE operations are constant and high-frequency, Databricks may autotune to a smaller target file size to reduce the duration and overhead of each merge. This behaviour is described explicitly in the documentation.

So, with this in view, the correct answer is A

upvoted 1 times

  **EZZALDIN** 3 months ago

**Selected Answer: E**

The primary goal of Auto Optimize and Auto Compaction in a streaming job isn't specifically to reduce MERGE duration. Instead, these features adjust file sizes based on the incremental volume of data being ingested in each micro-batch within a partition. Even though each partition contains around 1 GB of data (from the original OPTIMIZE), the streaming job writes small batches that are compacted into smaller files (often under 64 MB) because that's the amount of new data per batch.

So, Option E is more accurate: Databricks auto-tunes the target file size based on the amount of data in each partition (from each micro-batch), not specifically to speed up MERGE operations.

upvoted 1 times

  **Tedet** 4 months ago

**Selected Answer: E**

Option E is more accurate because Delta Lake's Auto Optimize and Auto Compaction are designed to adjust file sizes based on the streaming data partitioning, which inherently leads to smaller files over time. The system auto-tunes file sizes as new, incremental data is ingested and partitioned. Option A is plausible, but optimizing file sizes for MERGE operations is not the core focus of Auto Optimize in this case. The system's auto-tuning mechanism is more about managing file sizes based on the streaming data's partition size and maintaining efficient reads/writes, rather than directly optimizing for MERGE performance.

upvoted 3 times

  **Tedet** 4 months ago

**Selected Answer: A**

Options Behavior

auto (recommended) Tunes target file size while respecting other autotuning functionality. Requires Databricks Runtime 10.4 LTS or above.

legacy Alias for true. Requires Databricks Runtime 10.4 LTS or above.

true Use 128 MB as the target file size. No dynamic sizing.

false Turns off auto compaction. Can be set at the session level to override auto compaction for all Delta tables modified in the workload.

upvoted 1 times

  **rollno1** 4 months, 2 weeks ago

**Selected Answer: E**

MERGE operations are not the main update mechanism in this scenario—it's an incremental stream update, not batch MERGE. Larger partitions often result in smaller file sizes because:

Frequent incremental writes cause small batch updates.

Compaction happens at the partition level, not globally.

upvoted 1 times

🗨️ 👤 **Melik3** 10 months, 4 weeks ago

**Selected Answer: A**

It is important here to understand the difference between the partition size and the data files. the partition size is 1GB which is caused by OPTIMIZE and also expected. In each partition are data files. Databricks did an attuning to these datafile and resized them to a small size to be able to do MERGE statements efficiently that's why A is the correct answer

upvoted 4 times

🗨️ 👤 **imatheushenrique** 1 year ago

One of the purposes of a optimize execution is the gain in merge oprations, so:

A. Databricks has autotuned to a smaller target file size to reduce duration of MERGE operations

upvoted 1 times

🗨️ 👤 **RiktRikt007** 1 year, 4 months ago

how A is correct ? While Databricks does have autotuning capabilities, it primarily considers the table size. In this case, the table is over 10 TB, which would typically lead to a target file size of 1 GB, not under 64 MB.

upvoted 2 times

🗨️ 👤 **PrashantTiwari** 1 year, 4 months ago

The target file size is based on the current size of the Delta table. For tables smaller than 2.56 TB, the autotuned target file size is 256 MB. For tables with a size between 2.56 TB and 10 TB, the target size will grow linearly from 256 MB to 1 GB. For tables larger than 10 TB, the target file size is 1 GB. Correct answer is A

upvoted 2 times

🗨️ 👤 **AziLa** 1 year, 5 months ago

correct ans is A

upvoted 1 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: A**

A is correct

upvoted 2 times

Which statement regarding stream-static joins and static Delta tables is correct?

- A. Each microbatch of a stream-static join will use the most recent version of the static Delta table as of each microbatch.
- B. Each microbatch of a stream-static join will use the most recent version of the static Delta table as of the job's initialization.
- C. The checkpoint directory will be used to track state information for the unique keys present in the join.
- D. Stream-static joins cannot use static Delta tables because of consistency issues.
- E. The checkpoint directory will be used to track updates to the static Delta table.

**Suggested Answer: A**

Community vote distribution

A (100%)

  **Eertyy**  1 year, 9 months ago

B is the right answer as Option B is more typical for stream-static joins, as it provides a consistent static DataFrame snapshot for the entire job's duration. Option A might be suitable in specialized cases where you need real-time updates of the static DataFrame for each microbatch.

upvoted 12 times

  **arekm** 6 months ago

The explanation suggests the author would like the stream-static join to work in this way. However, it works as it does - see the first sentence in here: <https://learn.microsoft.com/en-us/azure/databricks/transform/join#stream-static>



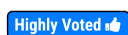
upvoted 3 times

  **hamzaKhribi** 1 year, 7 months ago

Answer is A, When Azure Databricks processes a micro-batch of data in a stream-static join, the latest valid version of data from the static Delta table joins with the records present in the current micro-batch

from <https://learn.microsoft.com/en-us/azure/databricks/structured-streaming/delta-lake>

upvoted 13 times

  **BrianNguyen95**  1 year, 10 months ago

correct answer is A



upvoted 6 times

  **KadELbied**  1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

  **JoG1221** 2 months, 1 week ago

**Selected Answer: B**

In a stream-static join, Spark treats the static Delta table as a constant snapshot at the time the streaming query starts. This means:

The static table is loaded once when the stream starts.

All micro-batches of the stream will join with that same version of the static table.

Any updates made to the static table after the stream starts will not be reflected in the join, unless the stream is restarted.



upvoted 1 times

  **arekm** 6 months ago

**Selected Answer: A**

A is correct, see: <https://learn.microsoft.com/en-us/azure/databricks/transform/join#stream-static>

upvoted 3 times

  **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: B**

A stream-static join joins the latest valid version of a Delta table (the static data) to a data stream using a stateless join.

When Databricks processes a micro-batch of data in a stream-static join, the latest valid version of data from the static Delta table joins with the records present in the current micro-batch. Because the join is stateless, you do not need to configure watermarking and can process results with low latency. The data in the static Delta table used in the join should be slowly-changing.

upvoted 1 times

🗨️ 👤 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: A**

<https://docs.databricks.com/en/transform/join.html#stream-static>

upvoted 1 times

🗨️ 👤 **akashdesarda** 9 months ago

**Selected Answer: A**

This is straight from docs, "A stream-static join joins the latest valid version of a Delta table (the static data) to a data stream using a stateless join.

When Azure Databricks processes a micro-batch of data in a stream-static join, the latest valid version of data from the static Delta table joins with the records present in the current micro-batch. Because the join is stateless, you do not need to configure watermarking and can process results with low latency. The data in the static Delta table used in the join should be slowly-changing."

<https://learn.microsoft.com/en-us/azure/databricks/transform/join#stream-static>

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: A**

correct answer is A

upvoted 1 times

🗨️ 👤 **hamzaKhribi** 1 year, 7 months ago

**Selected Answer: A**

Correct Answer A

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: A**

A is correct.

When Databricks processes a micro-batch of data in a stream-static join, the latest valid version of data from the static

upvoted 1 times

🗨️ 👤 **sagar21692** 1 year, 9 months ago

Correct answer is A. <https://docs.databricks.com/en/structured-streaming/delta-lake.html>

upvoted 1 times

A junior data engineer has been asked to develop a streaming data pipeline with a grouped aggregation using DataFrame df. The pipeline needs to calculate the average humidity and average temperature for each non-overlapping five-minute interval. Events are recorded once per minute per device.

Streaming DataFrame df has the following schema:

"device\_id INT, event\_time TIMESTAMP, temp FLOAT, humidity FLOAT"

Code block:

```
df.withWatermark("event_time", "10 minutes")
  .groupBy(
    _____
    "device_id"
  )
  .agg(
    avg("temp").alias("avg_temp"),
    avg("humidity").alias("avg_humidity")
  )
  .writeStream
  .format("delta")
  .saveAsTable("sensor_avg")
```

Choose the response that correctly fills in the blank within the code block to complete this task.

- A. to\_interval("event\_time", "5 minutes").alias("time")
- B. window("event\_time", "5 minutes").alias("time")
- C. "event\_time"
- D. window("event\_time", "10 minutes").alias("time")
- E. lag("event\_time", "10 minutes").alias("time")

**Suggested Answer: B**

Community vote distribution

B (100%)

🗳️ 👤 **42f87fd** 1 week, 4 days ago

**Selected Answer: B**

non-overlapping 5 minute interval --> so use window("event\_time", "5 minutes")

upvoted 1 times

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

Suretly B

upvoted 1 times

🗳️ 👤 **imatheushenrique** 7 months ago

B. window("event\_time", "5 minutes").alias("time")

In Structured Streaming, expressing such windows on event-time is simply performing a special grouping using the window() function. For example, counts over 5 minute tumbling (non-overlapping) windows on the eventTime column in the event is as following.

upvoted 4 times

🗳️ 👤 **Jay\_98\_11** 11 months, 3 weeks ago

**Selected Answer: B**

correct B

upvoted 2 times

🗳️ 👤 **kz\_data** 11 months, 3 weeks ago

**Selected Answer: B**

B is correct



upvoted 1 times

🗳️ 👤 **BIKRAM063** 1 year, 1 month ago

**Selected Answer: B**

Window of 5 mins

upvoted 2 times



  **sturcu** 1 year, 2 months ago

**Selected Answer: B**

B is correct:


<https://www.databricks.com/blog/2017/05/08/event-time-aggregation-watermarking-apache-sparks-structured-streaming.html>

upvoted 2 times

  **Eertyy** 1 year, 3 months ago

answer is B

upvoted 2 times

  **thxsgod** 1 year, 3 months ago

**Selected Answer: B**

Correct, B.

upvoted 4 times



A data architect has designed a system in which two Structured Streaming jobs will concurrently write to a single bronze Delta table. Each job is subscribing to a different topic from an Apache Kafka source, but they will write data with the same schema. To keep the directory structure simple, a data engineer has decided to nest a checkpoint directory to be shared by both streams.

The proposed directory structure is displayed below:

```
./bronze
├── __checkpoint
├── __delta_log
├── year_week=2020_01
├── year_week=2020_02
└── ...
```

Which statement describes whether this checkpoint directory structure is valid for the given scenario and why?

- A. No; Delta Lake manages streaming checkpoints in the transaction log.
- B. Yes; both of the streams can share a single checkpoint directory.
- C. No; only one stream can write to a Delta Lake table.
- D. Yes; Delta Lake supports infinite concurrent writers.
- E. No; each of the streams needs to have its own checkpoint directory.

**Suggested Answer: E**

Community vote distribution

E (90%)

10%

 **thxgod** Highly Voted 1 year, 9 months ago


**Selected Answer: E**

Correct, E.

Source:

[https://docs.databricks.com/en/optimizations/isolation-](https://docs.databricks.com/en/optimizations/isolation-level.html#:~:text=If%20a%20streaming%20query%20using%20the%20same%20checkpoint%20location%20is%20started%20multiple%20times%20concurrent)


[level.html#:~:text=If%20a%20streaming%20query%20using%20the%20same%20checkpoint%20location%20is%20started%20multiple%20times%20concurrent](https://docs.databricks.com/en/optimizations/isolation-level.html#:~:text=If%20a%20streaming%20query%20using%20the%20same%20checkpoint%20location%20is%20started%20multiple%20times%20concurrent)  
upvoted 11 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

Surely E

upvoted 1 times

 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: E**

E is the correct


upvoted 1 times

 **imatheushenrique** 1 year ago

E. No; each of the streams needs to have its own checkpoint directory.

The checkpoint directory is 1 to 1

upvoted 2 times

 **svik** 1 year, 1 month ago

**Selected Answer: B**

It is not clear from the question that year\_week=2020\_01 and year\_week=2020\_02 are used by stream 1 and stream 2 respectively. If they use the common parent checkpoint directory with individual sub folders for checkpointing, that should work fine. In that case the answer should be B

upvoted 2 times

 **Kill9** 1 year ago

That are table partitions. They are not used to build checkpoint adress. The adress finish at /bronze

upvoted 1 times

 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: E**

correct E



upvoted 1 times

  **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

E is correct



upvoted 1 times

  **sturcu** 1 year, 8 months ago

E is correct.

If user wants 1 checkpoint directory then he needs to unions streams before writing.

upvoted 2 times

  **Eertyy** 1 year, 10 months ago

answer is correct

upvoted 3 times


A Structured Streaming job deployed to production has been experiencing delays during peak hours of the day. At present, during normal execution, each microbatch of data is processed in less than 3 seconds. During peak hours of the day, execution time for each microbatch becomes very inconsistent, sometimes exceeding 30 seconds. The streaming write is currently configured with a trigger interval of 10 seconds. Holding all other variables constant and assuming records need to be processed in less than 10 seconds, which adjustment will meet the requirement?

- A. Decrease the trigger interval to 5 seconds; triggering batches more frequently allows idle executors to begin processing the next batch while longer running tasks from previous batches finish.
- B. Increase the trigger interval to 30 seconds; setting the trigger interval near the maximum execution time observed for each batch is always best practice to ensure no records are dropped.
- C. The trigger interval cannot be modified without modifying the checkpoint directory; to maintain the current stream state, increase the number of shuffle partitions to maximize parallelism.
- D. Use the trigger once option and configure a Databricks job to execute the query every 10 seconds; this ensures all backlogged records are processed with each batch.
- E. Decrease the trigger interval to 5 seconds; triggering batches more frequently may prevent records from backing up and large batches from causing spill.

**Suggested Answer: D**

Community vote distribution

E (100%)

  **RafaelCFC** Highly Voted 1 year, 5 months ago

**Selected Answer: E**

I believe this is a case of the least bad option, not exactly the best option possible.

- A is wrong because in Streaming you very rarely have any executors idle, as all cores are engaged in processing the window of data;
- B is wrong because triggering every 30s will not meet the 10s target processing interval;
- C is wrong in two manners: increasing shuffle partitions to any number above the number of available cores in the cluster will worsen performance in streaming; also, the checkpoint folder has no connection with trigger time.
- D is wrong because, keeping all other things the same as described by the problem, keeping the trigger time as 10s will not change the underlying conditions of the delay (i.e.: too much data to be processed in a timely manner).

E is the only option that might improve processing time.

upvoted 9 times

  **arekm** 6 months ago

With one addition to A explanation - micro-batches are sequential by design.

upvoted 1 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

  **arekm** 6 months ago

**Selected Answer: E**

Answer E, see explanation by RafaelCFC.

upvoted 1 times

  **ASRCA** 6 months ago

**Selected Answer: A**

Option A emphasizes utilizing idle executors to begin processing the next batch while longer-running tasks from previous batches finish. This approach can help maintain a steady flow of data processing and reduce the likelihood of bottlenecks.

upvoted 1 times

🗨️ **arekm** 6 months ago

Structured streaming processes batches in sequence. It does so since it guarantees exactly once processing, see:

<https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html>

upvoted 1 times

🗨️ **Thameur01** 6 months, 4 weeks ago

**Selected Answer: B**

If microbatch execution occasionally exceeds 30 seconds, a trigger interval of 5 seconds would cause multiple batches to queue up while the previous batch is still running. This would exacerbate the delays and potentially lead to backpressure and failure.

B is the best option in this case.

If we assume for sure that execution time should be less than 10s, then in that case a 5s interval will make more sense and E should be the best answer.

upvoted 1 times

🗨️ **wdeleersnyder** 10 months, 4 weeks ago

In Databricks Runtime 11.3 LTS and above, the Trigger.Once setting is deprecated. Databricks recommends you use Trigger.AvailableNow for all incremental batch processing workloads.

<https://docs.databricks.com/en/structured-streaming/triggers.html>

Doesn't seem like E is a valid and recommended option given that it is deprecated.

upvoted 2 times

🗨️ **wdeleersnyder** 10 months, 4 weeks ago

Ooops, I mean, D.

upvoted 2 times

🗨️ **imatheushenrique** 1 year ago

Considering the best option for performance gain is:

E. Decrease the trigger interval to 5 seconds; triggering batches more frequently may prevent records from backing up and large batches from causing spill.

upvoted 2 times

🗨️ **ojudz08** 1 year, 4 months ago

**Selected Answer: E**

E is the answer.

Enable the settings uses the 128 MB as the target file size

<https://learn.microsoft.com/en-us/azure/databricks/delta/tune-file-size>

upvoted 2 times

🗨️ **DAN\_H** 1 year, 5 months ago

**Selected Answer: E**

E is correct as A is wrong because in Streaming you very rarely have any executors idle

upvoted 2 times

🗨️ **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

I think is E is correct

upvoted 1 times

🗨️ **ervinshang** 1 year, 6 months ago

**Selected Answer: E**

correct answer is E

upvoted 1 times

🗨️ **ofed** 1 year, 7 months ago

Only C. Even if you trigger more frequently you decrease both load and time for this load. E doesn't change anything.

upvoted 1 times

🗨️ **sturcu** 1 year, 8 months ago

**Selected Answer: E**

Changing trigger interval to "one" will cause this to be a "batch" and will not execute in microbranches. This will not help at all

upvoted 4 times

🗨️ **Eertyy** 1 year, 9 months ago

correct answer is E

upvoted 1 times

  **azurearch** 1 year, 9 months ago

sorry, the caveat is holding all other variables constant.. that means we are not allowed to change trigger intervals. is C the answer then

upvoted 1 times

  **azurearch** 1 year, 9 months ago

what if in between those 5 seconds trigger interval if there are more records, that would still increase the time it takes to process.. i doubt E is correct. I will go with answer D. it is not to execute all queries within 10 secs. it is to execute trigger now batch every 10 seconds.

upvoted 1 times

  **azurearch** 1 year, 9 months ago

A option also is about setting trigger interval to 5 seconds, just to understand.. why its not the answer

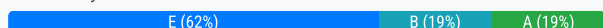
upvoted 1 times

Which statement describes Delta Lake Auto Compaction?

- A. An asynchronous job runs after the write completes to detect if files could be further compacted; if yes, an OPTIMIZE job is executed toward a default of 1 GB.
- B. Before a Jobs cluster terminates, OPTIMIZE is executed on all tables modified during the most recent job.
- C. Optimized writes use logical partitions instead of directory partitions; because partition boundaries are only represented in metadata, fewer small files are written.
- D. Data is queued in a messaging bus instead of committing data directly to memory; all data is committed from the messaging bus in one batch once the job is complete.
- E. An asynchronous job runs after the write completes to detect if files could be further compacted; if yes, an OPTIMIZE job is executed toward a default of 128 MB.

**Suggested Answer: A**

Community vote distribution



**aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: A**

Delta Lake's Auto Compaction feature is designed to improve the efficiency of data storage by reducing the number of small files in a Delta table. After data is written to a Delta table, an asynchronous job can be triggered to evaluate the file sizes. If it determines that there are a significant number of small files, it will automatically run the OPTIMIZE command, which coalesces these small files into larger ones, typically aiming for files around 1 GB in size for optimal performance.

E is incorrect because the statement is similar to A but with an incorrect default file size target.  
upvoted 5 times

**Kill9** 1 year ago

Table property delta.autoOptimize.autoCompact target 128 mb. For table property delta.tuneFileSizesForRewrites, tables larger than 10 TB, the target file size is 1 GB.

<https://learn.microsoft.com/en-us/azure/databricks/delta/tune-file-size>

upvoted 3 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

**RandomForest** 5 months, 2 weeks ago

**Selected Answer: E**

Delta Lake Auto Compaction is a feature that automatically detects opportunities to optimize small files. When a write operation is completed, an asynchronous job assesses whether the resulting files can be compacted into larger files (the default target size is 128 MB). If compaction is needed, the system executes an OPTIMIZE job in the background to improve file size and query performance.

This feature reduces the overhead of managing small files manually and improves storage and query efficiency. It aligns with Delta Lake's goal of simplifying and optimizing data lake performance.

upvoted 2 times

**mwynn** 5 months, 3 weeks ago

**Selected Answer: E**

I think it is E because they are just asking us to generally describe the feature - here's some info I gleaned from a DB Academy video:

- Compact small files on write with auto-optimize (tries to achieve file size of 128 MB)
- Auto-Compact launches a new job after execution of first Spark job (i.e. async), where it will try to compress files closer to 128 MB

upvoted 4 times

**pallazoj** 4 months, 2 weeks ago

This is true. I just heard the same statement in Databricks Academy video. Advanced Data Engineering with Databricks/Section5/Lesson1:Designing the foundation from 4:00 into the video!

upvoted 1 times

🗨️ 👤 **Nicks\_name** 6 months, 3 weeks ago

**Selected Answer: E**

typo in databricks documentation about sync job, but default size is explicitly mentioned as 128

upvoted 1 times

🗨️ 👤 **carah** 6 months, 3 weeks ago

**Selected Answer: B**

Table property: delta.autoOptimize.autoCompact

B. correct, although <https://docs.databricks.com/en/delta/tune-file-size.html#auto-compaction-for-delta-lake-on-databricks> does not mention OPTIMIZE, it is best option

A., E. wrong, auto compaction runs synchronously

C. wrong, it describes Table setting: delta.autoOptimize.optimizeWrite

D. wrong, not related to file compaction

upvoted 3 times

🗨️ 👤 **arekm** 6 months ago

The problem I have with B is that it says - on all tables. That depends on whether we use spark settings or table settings.

However, I still believe the asynchronous in A and E was meant to be synchronous (it is a typo). If it was not, then you are right :)

upvoted 1 times

🗨️ 👤 **vish9** 7 months, 4 weeks ago

There appears to be a typo in databricks documentation

upvoted 3 times

🗨️ 👤 **rrprofessional** 8 months ago

Enable auto compaction. By default will use 128 MB as the target file size.

upvoted 1 times

🗨️ 👤 **akashdesarda** 9 months ago

**Selected Answer: B**

If you go through this docs - then one thing is clear that it is not async job, so we have to eliminate A & C. D is wrong. It has no special job wrt the partition. Also file size of 128 MB is legacy config, latest one is dynamic. So we are left with B

upvoted 3 times

🗨️ 👤 **mouthwash** 5 months, 3 weeks ago

This. Don't be fooled by the typo answers, typo is inserted for a reason. It makes the answer wrong.

upvoted 1 times

🗨️ 👤 **pk07** 9 months ago

**Selected Answer: E**

<https://docs.databricks.com/en/delta/tune-file-size.html>

upvoted 2 times

🗨️ 👤 **partha1022** 10 months, 2 weeks ago

**Selected Answer: B**

Auto compaction is synchronous job.

upvoted 2 times

🗨️ 👤 **Shailly** 11 months, 2 weeks ago

**Selected Answer: B**

A and E are wrong because auto compaction is synchronous operation!

I vote for B

As per documentation - "Auto compaction occurs after a write to a table has succeeded and runs synchronously on the cluster that has performed the write. Auto compaction only compacts files that haven't been compacted previously."

<https://docs.delta.io/latest/optimizations-oss.html>

upvoted 4 times

🗨️ 👤 **imatheushenrique** 1 year ago

E. An asynchronous job runs after the write completes to detect if files could be further compacted; if yes, an OPTIMIZE job is executed toward a default of 128 MB.

<https://community.databricks.com/t5/data-engineering/what-is-the-difference-between-optimize-and-auto-optimize/td-p/21189>

upvoted 1 times

🗨️ 👤 **ojudz08** 1 year, 4 months ago

**Selected Answer: E**

E is the answer.

Enable the settings uses the 128 MB as the target file size

<https://learn.microsoft.com/en-us/azure/databricks/delta/tune-file-size>

upvoted 2 times

🗨️ 👤 **DAN\_H** 1 year, 5 months ago

**Selected Answer: E**

default file size is 128MB in auto compaction

upvoted 1 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

E is correct as the default file size is 128MB in auto compaction, not 1GB as normal OPTIMIZE statement.

upvoted 1 times

🗨️ 👤 **IWantCerts** 1 year, 5 months ago

**Selected Answer: E**

128MB is the default.

upvoted 1 times



Which statement characterizes the general programming model used by Spark Structured Streaming?

- A. Structured Streaming leverages the parallel processing of GPUs to achieve highly parallel data throughput.
- B. Structured Streaming is implemented as a messaging bus and is derived from Apache Kafka.
- C. Structured Streaming uses specialized hardware and I/O streams to achieve sub-second latency for data transfer.
- D. Structured Streaming models new data arriving in a data stream as new rows appended to an unbounded table.
- E. Structured Streaming relies on a distributed network of nodes that hold incremental state values for cached stages.

**Suggested Answer: D**

Community vote distribution

D (100%)

🗳️ 👤 **8605246** Highly Voted 1 year, 4 months ago

correct; The key idea in Structured Streaming is to treat a live data stream as a table that is being continuously appended. This leads to a new stream processing model that is very similar to a batch processing model. You will express your streaming computation as standard batch-like query as on a static table, and Spark runs it as an incremental query on the unbounded input table. Let's understand this model in more detail.

<https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html>

upvoted 5 times

🗳️ 👤 **KadELbied** Most Recent 1 month, 3 weeks ago

Selected Answer: D

Surely D

upvoted 1 times

🗳️ 👤 **arekm** 6 months ago

Selected Answer: D

D - see explanation of 8605246

upvoted 1 times

🗳️ 👤 **imatheushenrique** 7 months ago

D. Structured Streaming models new data arriving in a data stream as new rows appended to an unbounded table.

upvoted 2 times

🗳️ 👤 **mardigras** 10 months ago

Selected Answer: D

Yes. answer is D

upvoted 2 times

🗳️ 👤 **Jay\_98\_11** 11 months, 3 weeks ago

Selected Answer: D

vote for D

upvoted 1 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

Selected Answer: D

Correct.

Structured streaming needs to be considered as a table with append

upvoted 2 times

Which configuration parameter directly affects the size of a spark-partition upon ingestion of data into Spark?

- A. spark.sql.files.maxPartitionBytes
- B. spark.sql.autoBroadcastJoinThreshold
- C. spark.sql.files.openCostInBytes
- D. spark.sql.adaptive.coalescePartitions.minPartitionNum
- E. spark.sql.adaptive.advisoryPartitionSizeInBytes

**Suggested Answer: A**

Community vote distribution

A (100%)

🗳️ 👤 **8605246** Highly Voted 👍 1 year, 4 months ago

correct; The maximum number of bytes to pack into a single partition when reading files. This configuration is effective only when using file-based sources such as Parquet, JSON and ORC.

<https://spark.apache.org/docs/latest/sql-performance-tuning.html>

upvoted 5 times

🗳️ 👤 **KadELbied** Most Recent 🕒 1 month, 3 weeks ago

Selected Answer: A

suretly A

upvoted 1 times

🗳️ 👤 **Jay\_98\_11** 11 months, 3 weeks ago

Selected Answer: A

correct

upvoted 3 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

Selected Answer: A

from the provided list, this fits best.

In reality partition size/number can be influenced my many settings

upvoted 1 times

A Spark job is taking longer than expected. Using the Spark UI, a data engineer notes that the Min, Median, and Max Durations for tasks in a particular stage show the minimum and median time to complete a task as roughly the same, but the max duration for a task to be roughly 100 times as long as the minimum.

Which situation is causing increased duration of the overall job?

- A. Task queueing resulting from improper thread pool assignment.
- B. Spill resulting from attached volume storage being too small.
- C. Network latency due to some cluster nodes being in different regions from the source data
- D. Skew caused by more data being assigned to a subset of spark-partitions.
- E. Credential validation errors while pulling data from an external system.

**Suggested Answer: D**

*Community vote distribution*

D (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: D**

Surely D

upvoted 1 times

🗳️ 👤 **arekm** 6 months ago

**Selected Answer: D**

D - other answers don't make sense. In particular C - all nodes of the cluster must be in the same region (at least on AWS and Azure; GCP - I don't know, but they have networks spanning regions, so maybe it is possible).

upvoted 2 times

🗳️ 👤 **benni\_ale** 7 months ago

**Selected Answer: D**

D is correct

upvoted 1 times

🗳️ 👤 **AndreFR** 10 months, 1 week ago

A excluded because task queueing does not increase the duration of a task

B excluded, spill is writing to storage when a memory is insufficient (not storage insufficient)

C excluded, region cannot have a 100 times impact on duration

E excluded, no errors mentioned in question

upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

D. Skew caused by more data being assigned to a subset of spark-partitions.

upvoted 1 times

🗳️ 👤 **vikram12apr** 1 year, 3 months ago

**Selected Answer: D**

because a particular executors are executing majority of data while rest are processing very less. The total execution time depends upon the slowest executors.

Answer is D.

upvoted 3 times

🗳️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: D**

correct



upvoted 1 times

🗳️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: D**

I think D is correct



upvoted 1 times

  **sturcu** 1 year, 8 months ago

**Selected Answer: D**

D is correct

upvoted 1 times

  **Eertyy** 1 year, 9 months ago

D is the correct answer

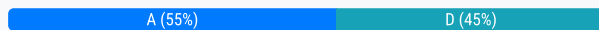
upvoted 3 times

Each configuration below is identical to the extent that each cluster has 400 GB total of RAM, 160 total cores and only one Executor per VM. Given a job with at least one wide transformation, which of the following cluster configurations will result in maximum performance?

- A. • Total VMs; 1
  - 400 GB per Executor
  - 160 Cores / Executor
- B. • Total VMs: 8
  - 50 GB per Executor
  - 20 Cores / Executor
- C. • Total VMs: 16
  - 25 GB per Executor
  - 10 Cores/Executor
- D. • Total VMs: 4
  - 100 GB per Executor
  - 40 Cores/Executor
- E. • Total VMs: 2
  - 200 GB per Executor
  - 80 Cores / Executor

**Suggested Answer: B**

Community vote distribution



**robson90** Highly Voted 1 year, 10 months ago

Option A, question is about maximum performance. Wide transformation will result in often expensive shuffle. With one executor this problem will be resolved. <https://docs.databricks.com/en/clusters/cluster-config-best-practices.html#complex-batch-etl>  
upvoted 43 times

**dp\_learner** 1 year, 7 months ago

source : <https://docs.databricks.com/en/clusters/cluster-config-best-practices.html>  
upvoted 3 times

**Ashok\_Choudhary\_CT** Highly Voted 2 months, 4 weeks ago

**Selected Answer: C**

How Option (C) Excels?

- ✓ More Executors (16 vs. 8 in Option B) → Faster parallel execution.
- ✓ Fewer Cores per Executor (10 vs. 20 in Option B) → Prevents CPU contention and scheduling delays.
- ✓ Better Memory Management (25GB vs. 50GB in Option B) → Reduces GC overhead.

Final Verdict

Option (C) is the "Best" configuration for handling a job with wide transformations.  
upvoted 5 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

**capt2101akash** 3 months ago


**Selected Answer: A**

The question talks about higher performance for one large wide transformation. This needs fewer large VMs/Executor. Therefore, one needs to choose the largest possible option.  
upvoted 1 times

**shaswat1404** 4 months, 3 weeks ago

**Selected Answer: C**

overly large executors are bad due to large Garbage Collection (GC) overhead and inefficient parallelism  
option C provides the best balance of parallelism, memory utilization and performance efficiency  
upvoted 1 times

  **fabiospont** 4 months, 3 weeks ago

**Selected Answer: A**

A is correct only one VM per Job.  
upvoted 1 times

  **HairyTorso** 6 months ago

**Selected Answer: B**

Number of workers

Choosing the right number of workers requires some trials and iterations to figure out the compute and memory needs of a Spark job. Here are some guidelines to help you start:

Never choose a single worker for a production job, as it will be the single point for failure

Start with 2-4 workers for small workloads (for example, a job with no wide transformations like joins and aggregations)

Start with 8-10 workers for medium to big workloads that involve wide transformations like joins and aggregations, then scale up if necessary

<https://www.databricks.com/discover/pages/optimize-data-workloads-guide#number-workers>

upvoted 4 times

  **arekm** 6 months ago

**Selected Answer: A**

Maximum performance - A guarantees no shuffles between nodes in the cluster. Only processes on one VM.  
upvoted 1 times

  **AlejandroU** 6 months, 3 weeks ago

**Selected Answer: B**

Answer B offers a good balance with 8 executors, providing a decent amount of memory and cores per executor, allowing for significant parallel processing.

Option C increases the number of executors further but at the cost of reduced memory and cores per executor, which might not be as effective for wide transformations.

upvoted 1 times

  **arekm** 6 months ago

The question is about maximum performance.

upvoted 1 times

  **janeZ** 6 months, 3 weeks ago

**Selected Answer: C**

for wide transformations, leveraging multiple executors typically results in better performance, resource utilization, and fault tolerance.

upvoted 2 times

  **Shakmak** 7 months ago

**Selected Answer: B**

B is a correct Answer based on

<https://www.databricks.com/discover/pages/optimize-data-workloads-guide#all-purpose>

upvoted 2 times

  **AndreFR** 7 months, 3 weeks ago

**Selected Answer: B**

Besides that A & E do not provide enough parallelism & fault tolerance, I can't explain why, but the correct answer is B. I got the same question during the exam and got 100% at tooling with answer B. (B is the answer provided by other sites similar to examtopics)

Choosing between B, C & D is tricky !

upvoted 3 times

  **Snakode** 7 months ago

Exactly, Also how can one node will resolve shuffle issue

upvoted 1 times

  **Nicks\_name** 6 months, 3 weeks ago

VM != node  
upvoted 1 times

  **kimberlysmith** 7 months, 3 weeks ago

**Selected Answer: B**

B

"Number of workers


Choosing the right number of workers requires some trials and iterations to figure out the compute and memory needs of a Spark job. Here are some guidelines to help you start:

Never choose a single worker for a production job, as it will be the single point for failure

Start with 2-4 workers for small workloads (for example, a job with no wide transformations like joins and aggregations)



Start with 8-10 workers for medium to big workloads that involve wide transformations like joins and aggregations, then scale up if necessary"

upvoted 3 times

  **benni\_ale** 7 months, 3 weeks ago

<https://www.databricks.com/discover/pages/optimize-data-workloads-guide>

upvoted 1 times

  **arik90** 1 year, 3 months ago

**Selected Answer: A**

Wide transformation falls under complex etl which means Option A is correct in the documentation didn't mention to do otherwise in this scenario.

upvoted 1 times

  **PrashantTiwari** 1 year, 4 months ago

A is correct

upvoted 1 times

  **vikrampatel5** 1 year, 5 months ago

**Selected Answer: A**

Option A:

<https://docs.databricks.com/en/clusters/cluster-config-best-practices.html#complex-batch-etl>

upvoted 3 times

  **RafaelCFC** 1 year, 5 months ago

**Selected Answer: A**

robson90's response explains it perfectly and has documentation to support it.

upvoted 1 times

A junior data engineer on your team has implemented the following code block.

```
MERGE INTO events
USING new_events
ON events.event_id = new_events.event_id
WHEN NOT MATCHED
  INSERT *
```

The view new\_events contains a batch of records with the same schema as the events Delta table. The event\_id field serves as a unique key for this table.

When this query is executed, what will happen with new records that have the same event\_id as an existing record?

- A. They are merged.
- B. They are ignored.
- C. They are updated.
- D. They are inserted.
- E. They are deleted.

**Suggested Answer:** B

Community vote distribution

B (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

suretly B

upvoted 1 times

🗳️ 👤 **Ashish7singh2020** 4 months, 3 weeks ago

**Selected Answer: B**

merge will work if no match

upvoted 1 times

🗳️ 👤 **arekm** 6 months ago

**Selected Answer: B**

No WHEN MATCHED section in MERGE, hence no action on those records, hence ignore - answer B.

upvoted 2 times

🗳️ 👤 **Shakmak** 7 months ago

**Selected Answer: B**

B is a correct Answer

upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

B. They are ignored.

Because there is not mention so there is no WHEN statement for this condition

upvoted 2 times

🗳️ 👤 **PrashantTiwari** 1 year, 4 months ago

B is correct

upvoted 2 times

🗳️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: B**

B is correct

upvoted 1 times



🗳️ 👤 **alexvno** 1 year, 8 months ago

**Selected Answer: B**

Ignored

upvoted 1 times



  **rairaix** 1 year, 9 months ago

**Selected Answer: B**

The answer is correct. "If none of the WHEN MATCHED conditions evaluate to true for a source and target row pair that matches the merge\_condition, then the row is inserted into the target table." <https://docs.databricks.com/en/sql/language-manual/delta-merge-into.html#:~:text=If%20none%20of%20the%20WHEN%20MATCHED%20conditions%20evaluate%20to%20true%20for%20a%20source%20and%20target%20row>  
upvoted 3 times

A junior data engineer seeks to leverage Delta Lake's Change Data Feed functionality to create a Type 1 table representing all of the values that have ever been valid for all rows in a bronze table created with the property `delta.enableChangeDataFeed = true`. They plan to execute the following code as a daily job:

```
from pyspark.sql.functions import col

(spark.read.format("delta")
 .option("readChangeFeed", "true")
 .option("startingVersion", 0)
 .table("bronze")
 .filter(col("_change_type").isin(["update_postimage", "insert"]))
 .write
 .mode("append")
 .table("bronze_history_type1")
 )
```

Which statement describes the execution and results of running the above query multiple times?

- A. Each time the job is executed, newly updated records will be merged into the target table, overwriting previous values with the same primary keys.
- B. Each time the job is executed, the entire available history of inserted or updated records will be appended to the target table, resulting in many duplicate entries.
- C. Each time the job is executed, the target table will be overwritten using the entire history of inserted or updated records, giving the desired result.
- D. Each time the job is executed, the differences between the original and current versions are calculated; this may result in duplicate entries for some records.
- E. Each time the job is executed, only those records that have been inserted or updated since the last execution will be appended to the target table, giving the desired result.

**Suggested Answer: B**

Community vote distribution

B (83%)

E (17%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

select B

upvoted 1 times

🗳️ 👤 **Chugs** 4 months ago

**Selected Answer: B**

As update type is insert and update so B is correct option.

upvoted 1 times

🗳️ 👤 **asdsadasdas** 4 months, 2 weeks ago

**Selected Answer: B**

B, Spark.read reads the entire table every time processed. If it was readstream then E would be answer

upvoted 3 times

🗳️ 👤 **Ashish7singh2020** 4 months, 3 weeks ago

**Selected Answer: B**

since start version is 0

upvoted 1 times

🗳️ 👤 **akashdesarda** 7 months, 4 weeks ago

**Selected Answer: B**

The starting version is 0, that means in every version entire data will be fetched. It is then append.

upvoted 1 times

🗨️ 👤 **faraaz132** 11 months ago

Correct Answer: B (not E)

Although it was pretty obvious to me, I still wrote the code to check and yes, it will append the entire change during every write since starting version is mentioned as 0.

If in doubt, code it yourselves

upvoted 2 times

🗨️ 👤 **imatheushenrique** 1 year ago

("startingVersion", 0) that means the entire history of table will be read so B.

upvoted 3 times

🗨️ 👤 **PrashantTiwari** 1 year, 4 months ago

B is correct

upvoted 2 times

🗨️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: B**

B is correct

upvoted 2 times

🗨️ 👤 **5ffcd04** 1 year, 6 months ago

**Selected Answer: B**

Correct B

upvoted 1 times

🗨️ 👤 **azurelearn2020** 1 year, 6 months ago

**Selected Answer: B**

correct answer is B.

upvoted 1 times

🗨️ 👤 **[Removed]** 1 year, 6 months ago

**Selected Answer: E**

Considering that we are talking about Change Data Feed and the code is filtering by [ "update\_postimage", "insert" ] the column "\_change\_type", I would go with the option E.

Reference:

[https://docs.delta.io/latest/delta-change-data-feed.html#:~:text=\\_change\\_type,update\\_preimage%20%2C%20update\\_postimage](https://docs.delta.io/latest/delta-change-data-feed.html#:~:text=_change_type,update_preimage%20%2C%20update_postimage)

upvoted 1 times

🗨️ 👤 **5ffcd04** 1 year, 6 months ago

Notice option ("startingVersion", 0), which will bring all changes from beginning. Hence Answer is B.

upvoted 6 times

🗨️ 👤 **jyothsna12496** 1 year, 8 months ago

why is it Not E. It gets newly inserted or updated records

upvoted 1 times

🗨️ 👤 **[Removed]** 1 year, 6 months ago

I'm with you, follow the reference:

[https://docs.delta.io/latest/delta-change-data-feed.html#:~:text=\\_change\\_type,update\\_preimage%20%2C%20update\\_postimage](https://docs.delta.io/latest/delta-change-data-feed.html#:~:text=_change_type,update_preimage%20%2C%20update_postimage)

upvoted 1 times

🗨️ 👤 **5ffcd04** 1 year, 6 months ago

Notice .option ("startingVersion", 0), which will bring all changes from beginning. Hence Answer is B.

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: B**

correct

upvoted 1 times

🗨️ 👤 **azurearch** 1 year, 9 months ago



B is the right answer, sorry.

upvoted 2 times

  **azurearch** 1 year, 9 months ago

answer is A, because there is a filter as asmayassineg said. Filter filters only existing records from change feed

upvoted 1 times

  **asmayassineg** 1 year, 11 months ago

sorry, answer is correct B.

upvoted 2 times

A new data engineer notices that a critical field was omitted from an application that writes its Kafka source to Delta Lake. This happened even though the critical field was in the Kafka source. That field was further missing from data written to dependent, long-term storage. The retention threshold on the Kafka service is seven days. The pipeline has been in production for three months.

Which describes how Delta Lake can help to avoid data loss of this nature in the future?

- A. The Delta log and Structured Streaming checkpoints record the full history of the Kafka producer.
- B. Delta Lake schema evolution can retroactively calculate the correct value for newly added fields, as long as the data was in the original source.
- C. Delta Lake automatically checks that all fields present in the source data are included in the ingestion layer.
- D. Data can never be permanently dropped or deleted from Delta Lake, so data loss is not possible under any circumstance.
- E. Ingesting all raw data and metadata from Kafka to a bronze Delta table creates a permanent, replayable history of the data state.

**Suggested Answer:** E

Community vote distribution

E (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer:** E

select E

upvoted 1 times

🗳️ 👤 **kishanu** 2 months, 3 weeks ago

**Selected Answer:** E

E is the right answer, as the table in bronze can be replayed again when required.

upvoted 1 times

🗳️ 👤 **Tedet** 4 months ago

**Selected Answer:** A

Considering the Databricks documentation on change feed and your need to process new records that have not been processed yet, Option A might actually be a better fit since you're looking for a streaming solution that can continuously monitor new records. The change feed (Option D) works for batch processing changes from a specific version, which isn't ideal for real-time streaming.

upvoted 1 times

🗳️ 👤 **HairyTorso** 6 months ago

**Selected Answer:** E

E lgtm

upvoted 1 times

🗳️ 👤 **Anithec0der** 6 months, 3 weeks ago

**Selected Answer:** E

When we design pipeline, we will have to make sure data from source will be present there in the raw layer/bronze layer and the transformation we make should be done in refine and enterprise layer so by this way we can tackle this kind of situation where the necessary column was not replicated in previous runs of pipeline and we can create new column based on raw data we have.

upvoted 3 times

🗳️ 👤 **imatheushenrique** 1 year ago

Medallion Architecture is named in E. (Ingesting all raw data and metadata from Kafka to a bronze Delta table creates a permanent, replayable history of the data state.)

upvoted 3 times

🗳️ 👤 **ojudz08** 1 year, 4 months ago

**Selected Answer:** E

E is correct


upvoted 2 times

🗳️ 👤 **DAN\_H** 1 year, 5 months ago

**Selected Answer:** E

I think E is correct

upvoted 1 times

  **kz\_data** 1 year, 5 months ago

**Selected Answer: E**

I think E is correct

upvoted 1 times

  **alexvno** 1 year, 7 months ago

**Selected Answer: E**

Looks good - E

upvoted 2 times

A nightly job ingests data into a Delta Lake table using the following code:

```
from pyspark.sql.functions import current_timestamp, input_file_name, col
from pyspark.sql.column import Column

def ingest_daily_batch(time_col: Column, year:int, month:int, day:int):
    (spark.read
     .format("parquet")
     .load(f"/mnt/daily_batch/{year}/{month}/{day}")
     .select("*,
            time_col.alias("ingest_time"),
            input_file_name().alias("source_file")
            )
     .write
     .mode("append")
     .saveAsTable("bronze")
    )
```

The next step in the pipeline requires a function that returns an object that can be used to manipulate new records that have not yet been processed to the next table in the pipeline.

Which code snippet completes this function definition?

def new\_records():

A. return spark.readStream.table("bronze")

B. return spark.readStream.load("bronze")

C. 

```
return (spark.read
        .table("bronze")
        .filter(col("ingest_time") == current_timestamp())
        )
```

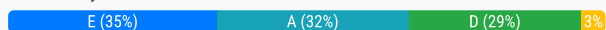
D. return spark.read.option("readChangeFeed", "true").table("bronze")

E. 

```
return (spark.read
        .table("bronze")
        .filter(col("source_file") == f"/mnt/daily_batch/{year}/{month}/{day}")
        )
```

#### Suggested Answer: D

Community vote distribution



**AzureDE2522** Highly Voted 1 year, 7 months ago

**Selected Answer: D**

# not providing a starting version/timestamp will result in the latest snapshot being fetched first

```
spark.readStream.format("delta") \
.option("readChangeFeed", "true") \
.table("myDeltaTable")
```

Please refer:

<https://docs.databricks.com/en/delta/delta-change-data-feed.html>

upvoted 13 times

**arekm** 6 months ago

Answer D would require specifying the start and (optionally) the end version for reading data from CDF. So D does not seem to be correct.

upvoted 2 times

**shaojunni** 8 months, 3 weeks ago

readChangeFeed is disabled by default.

upvoted 2 times

**t\_d\_v** 10 months, 2 weeks ago


There is no stream in option D

upvoted 3 times

**GHill1982** 10 months, 1 week ago

You can read Delta Lake Change Data Feed without using a stream. You can use batch queries to read the change data feed by setting the readChangeFeed option to true.

upvoted 2 times

  **arekm** 6 months ago

CDF without a stream requires a starting version at the minimum.

upvoted 1 times

  **Laraujo2022**  1 year, 7 months ago

In my opinion E is not correct because we do not see parameters pass within to the function (year, month and day)... the function is def new\_records():


upvoted 9 times

  **ConquerorAlpha**  4 days, 2 hours ago

**Selected Answer: E**

The question is clearly saying that the table is being written in batch format, then how you can read is using streaming code, hence option E is correct there with the proper filter to read the current date data. Please think logically.

upvoted 1 times

  **BryOs** 4 weeks ago

**Selected Answer: A**

It's Option A because it will allow you to get the latest changes even if CDF isn't enabled on the table. Option D would fail if CDF isn't enabled on the table. The statement doesn't indicate if CDF is active.


upvoted 1 times

  **KadELbied** 1 month, 2 weeks ago

**Selected Answer: A**

Suretly A


upvoted 1 times

  **AlHerd** 3 months ago

**Selected Answer: A**

Option A is best because it creates a streaming source that reads only new appended data from the "bronze" table incrementally. Even if ingestion is done in batch, using spark.readStream.table("bronze") lets downstream processing treat the table as a live data stream.

upvoted 2 times

  **Tedet** 4 months ago

**Selected Answer: D**

Explanation: This is the best option for Delta Lake, as it uses the readChangeFeed option. This option is specifically designed to read only the new changes (insertions, updates, or deletions) since the last read, which is exactly what is needed when you want to handle new records that have not yet been processed. This ensures that only records that are new or changed since the last read are returned.

Conclusion: This is the correct choice, as it ensures that only new records are read.



upvoted 1 times

  **asdsadasdas** 4 months, 2 weeks ago

**Selected Answer: A**

"manipulate new records that have \*not yet been processed\* to the next table " readstream can incrementally pick data yet to be processed. with D the issue is spark.read it will read the entire table

upvoted 1 times

  **asdsadasdas** 4 months, 2 weeks ago

Batch (read) Reads all available CDF history starting from the earliest retained version May load too much data or fail if old versions are deleted  
Streaming (readStream) Starts from the latest version unless a checkpoint exists

upvoted 1 times

  **shaswat1404** 4 months, 3 weeks ago

**Selected Answer: E**

in option A and B assume steaming ingestiopn but ingestion is in batch mode

in option C current\_timestamp is used which is dynamic and changes every time the query is executed therefore it wont correctly filter records injected in the last batch

in option D it only works if delta.enableChangeDataFeed = true was set on the table before the ingestion (its disabled by default and given query does not set this option as true) therefore this option is in valid

option E is correct as it correctly filters from the most recent batch as it uses file path to retrieve only data from the latest ingestion column source\_file was created specifically for this purpose ensuring the function returns onle new records..



upvoted 2 times

🗨️ 👤 **arekm** 6 months ago

**Selected Answer: A**

You can read data from the delta table using structured streaming. You have 2 options:

- without CDF - only process new rows (without updates and deletes)
- with CDF - all changes to the data, i.e. insert, update, delete.

Answer A uses the first option. However, in the question they talk about "new records". So using streaming for new records is OK. Answer A is correct.

upvoted 2 times

🗨️ 👤 **arekm** 6 months ago

At first I thought of answer D. However, after checking in the docs I learned that starting version is a must while reading from CDF using batch pattern.

upvoted 1 times

🗨️ 👤 **sgerin** 6 months, 2 weeks ago

**Selected Answer: E**

New records will be filtered for D /

upvoted 1 times

🗨️ 👤 **temple1305** 6 months, 2 weeks ago

**Selected Answer: D**

New records will be filtered for D -

example <https://delta.io/blog/2023-07-14-delta-lake-change-data-feed-cdf/>

upvoted 1 times

🗨️ 👤 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: A**

Answer A. A better approach would involve streaming directly from the Delta table (Option A), possibly along with using metadata like ingest\_time to track new records more accurately.

It might be better to rely on the streaming process itself rather than trying to filter based on the file path (option E).

upvoted 1 times

🗨️ 👤 **Thameur01** 6 months, 4 weeks ago

**Selected Answer: E**

Using the source\_file metadata field allows you to filter new records ingested from specific files.

E is the most robust and reliable option for tracking and working with new records in this batch ingestion pipeline.

upvoted 1 times

🗨️ 👤 **benni\_ale** 7 months ago

**Selected Answer: E**

I tried myself but none really works

upvoted 1 times

🗨️ 👤 **cbj** 8 months, 1 week ago

**Selected Answer: A**

Others can't ensure data not being processed. e.g. if the code not run for one day and run next day, C or E will mis process one day's data.

upvoted 3 times

🗨️ 👤 **shaojunni** 8 months, 3 weeks ago

**Selected Answer: A**

since "bronze" table is a delta table, readStream() only returns new data.

upvoted 4 times

A junior data engineer is working to implement logic for a Lakehouse table named `silver_device_recordings`. The source data contains 100 unique fields in a highly nested JSON structure.

The `silver_device_recordings` table will be used downstream to power several production monitoring dashboards and a production model. At present, 45 of the 100 fields are being used in at least one of these applications.

The data engineer is trying to determine the best approach for dealing with schema declaration given the highly-nested structure of the data and the numerous fields.

Which of the following accurately presents information about Delta Lake and Databricks that may impact their decision-making process?

- A. The Tungsten encoding used by Databricks is optimized for storing string data; newly-added native support for querying JSON strings means that string types are always most efficient.
- B. Because Delta Lake uses Parquet for data storage, data types can be easily evolved by just modifying file footer information in place.
- C. Human labor in writing code is the largest cost associated with data engineering workloads; as such, automating table declaration logic should be a priority in all migration workloads.
- D. Because Databricks will infer schema using types that allow all observed data to be processed, setting types manually provides greater assurance of data quality enforcement.
- E. Schema inference and evolution on Databricks ensure that inferred types will always accurately match the data types used by downstream systems.

**Suggested Answer: D**

Community vote distribution

D (100%)

🗳️ **RafaelCFC** Highly Voted 12 months ago

**Selected Answer: D**

A is wrong, because Tungsten is a project around improving Spark's efficiency on memory and CPU usage;

B is wrong because Parquet does not support file editing, it only supports overwrite and create operations by itself;

C is wrong because completely automating schema declaration for tables will incur in reduced previsibility for data types and data quality;

E is false because unlucky sampling can yield bad inferences by Spark;

upvoted 14 times

🗳️ **hal2401me** Highly Voted 9 months, 2 weeks ago

from my exam today, both C & D are no longer available, so they can't be correct.

E & A are available. E states "always accurate" so I hesitate to choose it.

There is a new option stating like "delta lake indexes first 32column in delta log for Z order and optimization"(not sure I remember exactly, it looks statementfully correct). and I chosed this "new" option. Because, this should impact the schema decision by putting high-usage field in the first 32 columns.

upvoted 7 times

🗳️ **Tedet** Most Recent 4 months ago

**Selected Answer: D**

Explanation: Databricks can infer schema when reading data, but automatic schema inference doesn't always guarantee the accuracy of data types.

For complex or highly-nested structures, schema inference might not always align with the actual data quality or the needs of downstream

applications, and manual type definition ensures that the schema is more consistent and predictable. While automatic inference is useful for quick analysis or exploratory work, manual schema definition provides better data quality assurance in production workloads, especially when dealing with large, complex data structures.

Conclusion: This statement correctly emphasizes the importance of manual schema declaration to ensure data quality enforcement and consistency, especially when dealing with complex structures. Best option.

upvoted 1 times

🗳️ **guillesd** 10 months, 4 weeks ago

**Selected Answer: D**



Only answer that makes sense

upvoted 1 times

🗳️ **AziLa** 11 months, 1 week ago

Correct Ans is D



upvoted 1 times

  **sturcu** 1 year, 2 months ago

**Selected Answer: D**

correct

upvoted 2 times

  **hammer\_1234\_h** 1 year, 3 months ago

D is correct.

we can use `schema hint` to enforce the schema information that we know and expect on an inferred schema.

upvoted 2 times

The data engineering team maintains the following code:

```
accountDF = spark.table("accounts")
orderDF = spark.table("orders")
itemDF = spark.table("items")

orderWithItemDF = (orderDF.join(
    itemDF,
    orderDF.itemID == itemDF.itemID)
    .select(
        orderDF.accountID,
        orderDF.itemID,
        itemDF.itemName))

finalDF = (accountDF.join(
    orderWithItemDF,
    accountDF.accountID == orderWithItemDF.accountID)
    .select(
        orderWithItemDF["*"],
        accountDF.city))

(finalDF.write
    .mode("overwrite")
    .table("enriched_itemized_orders_by_account"))
```

Assuming that this code produces logically correct results and the data in the source tables has been de-duplicated and validated, which statement describes what will occur when this code is executed?

- A. A batch job will update the enriched\_itemized\_orders\_by\_account table, replacing only those rows that have different values than the current version of the table, using accountID as the primary key.
- B. The enriched\_itemized\_orders\_by\_account table will be overwritten using the current valid version of data in each of the three tables referenced in the join logic.
- C. An incremental job will leverage information in the state store to identify unjoined rows in the source tables and write these rows to the enriched\_itemized\_orders\_by\_account table.
- D. An incremental job will detect if new rows have been written to any of the source tables; if new rows are detected, all results will be recalculated and used to overwrite the enriched\_itemized\_orders\_by\_account table.
- E. No computation will occur until enriched\_itemized\_orders\_by\_account is queried; upon query materialization, results will be calculated using the current valid version of data in each of the three tables referenced in the join logic.

**Suggested Answer: B**

Community vote distribution

B (100%)

arekm 6 months ago

**Selected Answer: B**

B - it is a batch overwrite, which means: whatever was there is gone.

upvoted 1 times

nedlo 8 months, 1 week ago

**Selected Answer: B**

i agree. Cannot be E because write itself is action

upvoted 1 times

AndreFR 10 months, 1 week ago

**Selected Answer: B**

B because code has : .mode("Overwrite")

upvoted 1 times

imatheushenrique 1 year ago

B is correct

upvoted 1 times

🗨️ 👤 **AziLa** 1 year, 5 months ago

Correct Ans is B

upvoted 2 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: B**

correct

upvoted 2 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: B**

B is correct

upvoted 3 times

The data engineering team is migrating an enterprise system with thousands of tables and views into the Lakehouse. They plan to implement the target architecture using a series of bronze, silver, and gold tables. Bronze tables will almost exclusively be used by production data engineering workloads, while silver tables will be used to support both data engineering and machine learning workloads. Gold tables will largely serve business intelligence and reporting purposes. While personal identifying information (PII) exists in all tiers of data, pseudonymization and anonymization rules are in place for all data at the silver and gold levels.

The organization is interested in reducing security concerns while maximizing the ability to collaborate across diverse teams.

Which statement exemplifies best practices for implementing this system?

- A. Isolating tables in separate databases based on data quality tiers allows for easy permissions management through database ACLs and allows physical separation of default storage locations for managed tables.
- B. Because databases on Databricks are merely a logical construct, choices around database organization do not impact security or discoverability in the Lakehouse.
- C. Storing all production tables in a single database provides a unified view of all data assets available throughout the Lakehouse, simplifying discoverability by granting all users view privileges on this database.
- D. Working in the default Databricks database provides the greatest security when working with managed tables, as these will be created in the DBFS root.
- E. Because all tables must live in the same storage containers used for the database they're created in, organizations should be prepared to create between dozens and thousands of databases depending on their data isolation requirements.

**Suggested Answer: A**

Community vote distribution

A (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times

🗳️ 👤 **arekm** 6 months ago

**Selected Answer: A**

A - most logical

B - it is a logical construct, but under default settings tables are stored where the database is so there is a security component to it

C - never a good idea to store everything in one db since db allows to group tables with similar area of interest and allows to manage permissions (like groups in Entra and assigning permissions to groups)

D - not default database does not mean we cannot use managed tables and you can specify your location still; I do not think that storing anything on DBFS is a good idea - even Databricks suggests to use workspaces for your code, not to mention the data.

E - thousand databases - nonsense; you can specify the location of individual tables.

upvoted 2 times

🗳️ 👤 **arekm** 6 months ago

Correction to D explanation - apparently the default location is bfs

upvoted 1 times

🗳️ 👤 **strayda** 1 year ago

**Selected Answer: A**

The most logical answer is A

upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

A is correct



upvoted 2 times

🗳️ 👤 **ojudz08** 1 year, 4 months ago

**Selected Answer: A**

answer is A

upvoted 1 times

  **AziLa** 1 year, 5 months ago

Correct Ans is A

upvoted 2 times

  **Enduresoul** 1 year, 7 months ago

**Selected Answer: A**

A is correct

upvoted 2 times

The data architect has mandated that all tables in the Lakehouse should be configured as external Delta Lake tables. Which approach will ensure that this requirement is met?

- A. Whenever a database is being created, make sure that the LOCATION keyword is used
- B. When configuring an external data warehouse for all table storage, leverage Databricks for all ELT.
- C. Whenever a table is being created, make sure that the LOCATION keyword is used.
- D. When tables are created, make sure that the EXTERNAL keyword is used in the CREATE TABLE statement.
- E. When the workspace is being configured, make sure that external cloud object storage has been mounted.

**Suggested Answer: C**

Community vote distribution

C (100%)

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

suretly C

upvoted 1 times

 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: C**

Note: External keyword is not mandatory.

Location is mandatory the presence implies, that the table is external

upvoted 2 times

 **carah** 6 months, 3 weeks ago

**Selected Answer: C**

A. is not correct:

having schema with LOCATION

```
CREATE SCHEMA my_schema
```

```
LOCATION 's3://<bucket-path>/my_schema';
```

Table Location Scenarios:

Table Without LOCATION:

```
CREATE TABLE my_schema.my_table (id INT);
```

The table will be stored in the default warehouse directory (e.g., dbfs:/user/hive/warehouse/), not the schema's LOCATION.

Table With Explicit LOCATION: If you want the table to be stored under the schema's LOCATION, you need to specify the location explicitly:

```
CREATE TABLE my_schema.my_table (id INT)
```

```
LOCATION 's3://<bucket-path>/my_schema/my_table/';
```

So, if you want all tables under the schema to use the schema's LOCATION, explicitly specify the LOCATION for each table during creation.

upvoted 3 times

 **y2kal** 7 months, 3 weeks ago

It should be A, as the question states "all tables". Once an external DB is created, then all the tables in that would be by default be external.

upvoted 1 times

 **akashdesarda** 7 months, 4 weeks ago

**Selected Answer: A**

A is correct. If a database is created using location keyword then by default all the tables created in it will use that location. They follows <provided location>/\_unity\_catalog/tables/<uuid>



upvoted 1 times

🗨️ 👤 **leopedroso1** 1 year, 4 months ago

C is the correct answer. According to the documentation only the LOCATION is needed to make a table external. Moreover, we can also assume the keyword EXTERNAL is optional in the SQL statement.

<https://docs.databricks.com/en/sql/language-manual/sql-ref-external-tables.html>

upvoted 2 times

🗨️ 👤 **CY** 1 year, 4 months ago

'A' seems more appropriate.

All the tables in Delta lake house should be marked as external.. which can be achieved using location keyword at database level instead of each table level.

upvoted 3 times

🗨️ 👤 **Yogi05** 1 year, 6 months ago

Why not D? i know both C and D are same, but D is more precise

upvoted 2 times

🗨️ 👤 **Yogi05** 1 year, 6 months ago

my bad. D is having EXTERNAL keyword, got confused. C is correct answer

upvoted 2 times

🗨️ 👤 **Laraujo2022** 1 year, 7 months ago

If you set a location in a database level, all tables under this database are automatically external table, in my opinion is A is correct.

upvoted 1 times

🗨️ 👤 **Isio05** 1 year ago

According to what I've found in Databricks forums: "Database location and Table location are independent". So it looks like specifying location at DB level is not sufficient as tables will be still created as managed ones.

upvoted 3 times

🗨️ 👤 **Quadronoid** 1 year, 8 months ago

**Selected Answer: C**

C is correct. Location keyword should be in create script of the table

upvoted 4 times

🗨️ 👤 **mouad\_attaqi** 1 year, 8 months ago

C is correct, the key word to be used is Location, the keyword external is optional

upvoted 3 times

🗨️ 👤 **chokthewa** 1 year, 8 months ago

The correct is D

upvoted 2 times

🗨️ 👤 **mht3336** 1 year, 5 months ago

there is no EXTERNAL key word in databricks, however it is there for other systems like Oracle, Hive, Cassandra etc.

upvoted 1 times

🗨️ 👤 **Dusica** 1 year, 1 month ago

and microsoft synapse

upvoted 1 times

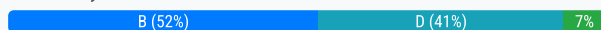
To reduce storage and compute costs, the data engineering team has been tasked with curating a series of aggregate tables leveraged by business intelligence dashboards, customer-facing applications, production machine learning models, and ad hoc analytical queries. The data engineering team has been made aware of new requirements from a customer-facing application, which is the only downstream workload they manage entirely. As a result, an aggregate table used by numerous teams across the organization will need to have a number of fields renamed, and additional fields will also be added.

Which of the solutions addresses the situation while minimally interrupting other teams in the organization without increasing the number of tables that need to be managed?

- A. Send all users notice that the schema for the table will be changing; include in the communication the logic necessary to revert the new table schema to match historic queries.
- B. Configure a new table with all the requisite fields and new names and use this as the source for the customer-facing application; create a view that maintains the original data schema and table name by aliasing select fields from the new table.
- C. Create a new table with the required schema and new fields and use Delta Lake's deep clone functionality to sync up changes committed to one table to the corresponding table.
- D. Replace the current table definition with a logical view defined with the query logic currently writing the aggregate table; create a new table to power the customer-facing application.
- E. Add a table comment warning all users that the table schema and field names will be changing on a given date; overwrite the table in place to the specifications of the customer-facing application.

**Suggested Answer: B**

Community vote distribution



**guilleld** Highly Voted 1 year, 4 months ago

**Selected Answer: B**

B makes way more sense, the number of tables managed do not increase since the old table won't be used anymore, then the view on top of this table is not another table to manage, just maintains the "original API" of the table to avoid breaking changes in downstream applications  
upvoted 7 times

**alexnvo** Highly Voted 1 year, 3 months ago

**Selected Answer: D**

Create view. Can't be B as -> without increasing the number of tables that need to be managed  
upvoted 7 times

**EZZALDIN** 4 months, 1 week ago

Replaces the current table with a view, but still creates a separate table for the customer app, this doesn't cleanly separate the two schemas.  
upvoted 1 times

**carlosmps** 6 months, 3 weeks ago

But option B does not increase the number of tables to maintain; in fact, it replaces the source table. The question states that it should not increase, and from that perspective, the NUMBER of tables does not increase. It only replaces the source and creates a view.  
upvoted 1 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: B**

suretly B  
upvoted 1 times

**EZZALDIN** 4 months, 1 week ago

**Selected Answer: B**

it recommends creating a view that maps the new table back to the original schema. This view lets other teams continue using the table as they always have, with no changes to their queries.  
upvoted 1 times

**RandomForest** 5 months, 2 weeks ago

**Selected Answer: B**

This approach achieves the following key goals:

1. Minimizes Disruption: By creating a view that mirrors the original schema, existing workloads that depend on the current schema remain uninterrupted.

Other teams can continue their queries without needing to adjust their logic for the schema change.

2. Meets New Requirements: The new table accommodates the changes required by the customer-facing application, ensuring that the application's updated requirements are fulfilled.

3. Avoids Table Duplication: Instead of maintaining multiple tables for the same dataset, this approach uses a combination of a new table and a view, reducing the overall management burden.

4. Flexibility for Future Changes: Views can be adjusted as needed, providing a layer of abstraction. Future schema updates can be handled similarly without directly impacting dependent systems.

upvoted 1 times

🗨️ 👤 **HairyTorso** 6 months ago

**Selected Answer: B**

Create view -> number of tables stay the same. Option D has overhead

upvoted 1 times

🗨️ 👤 **arekm** 6 months ago

**Selected Answer: B**

B - but I was wondering between B & D.

I do not like D since you replace the table with a view (query costs + you need to change the currently working workflow). Additionally, you create a table that does similar thing to the view - why?

upvoted 1 times

🗨️ 👤 **Nicks\_name** 7 months, 1 week ago

E. not D because, by converting the aggregate table into a view, might introduce performance overheads as every access now potentially involves running complex query logic to reconstruct the desired dataset on-the-fly. This might not be ideal for performance-sensitive applications like business intelligence dashboards.

upvoted 1 times

🗨️ 👤 **vish007** 7 months, 2 weeks ago

**Selected Answer: B**

Option D will increase the Compute cost significantly as all the downstream teams will run the view which has logic for Aggregate table.

Option B make more sense with less impact to storage and compute cost which is the original ask for the data engineering team in the question.

upvoted 2 times

🗨️ 👤 **benni\_ale** 7 months, 3 weeks ago

**Selected Answer: D**

I am not sure whether B or D... I believe B increases the number of managed Tables as it states that a CREATE TABLE statement is run before a CREATE VIEW ... the fact that the CREATE VIEW will replace the current table is not really specified... still one could argue that it would be dumb not to do it but at this point i would say that D is more precise

upvoted 1 times

🗨️ 👤 **b.b.da.costa** 7 months, 3 weeks ago

The problem with this question is if the order of the sentence matters.

B: Create a table then create a view. Teams are interrupted after the creation of the table.

D: Create a view then create a table. Teams are not interrupted because they are consuming the view first.

upvoted 1 times

🗨️ 👤 **vish007** 7 months, 2 weeks ago

Option D will increase the Compute cost significantly as all the downstream teams will run the view which has logic for Aggregate table.

Option B make more sense with less impact to storage and compute cost which is the original ask for the data engineering team in the question.

upvoted 1 times

🗨️ 👤 **benni\_ale** 7 months, 3 weeks ago

Also B does it really not increase the number of written tables? It states that a CREATE TABLE is run and CREATE VIEW is run... Nothing really points to the fact that the view will replace the table... Indeed I would opt for D

upvoted 1 times

🗨️ 👤 **kimberlysmith** 7 months, 3 weeks ago

**Selected Answer: B**

B is Correct. It does not create additional tables. The view mimics the old schema so not to interrupt downstream consumers. It ensures the aggregates are persisted to save on compute.

D is incorrect mostly due to the aggregates being baked into the view which is not optimal as each time downstream users query the view the joins and aggregates have to be recomputed.

upvoted 1 times

🗨️ 👤 **shaojunni** 8 months, 3 weeks ago

**Selected Answer: D**

D will not increase the number of table. It will create a new table and replace the aggregation table with a view. B will create a new table, a new view match old table name and schema, aggregation table still there.

upvoted 1 times

🗨️ 👤 **KB\_Ai\_Champ** 9 months, 2 weeks ago

option D is correct

docs : <https://docs.databricks.com/en/delta/update-schema.html>

also they specifically says that they dont want to increase managed tables!

upvoted 2 times

🗨️ 👤 **KB\_Ai\_Champ** 9 months, 2 weeks ago

Reasons :

No Increase in Managed Tables: By replacing the current table with a view, you maintain the same number of managed tables.

Backward Compatibility: The view can mimic the original table's schema, ensuring that existing queries and applications continue to function without modification.

Dedicated Table for New Requirements: The new table can be tailored to meet the specific needs of the customer-facing application without affecting other users.

upvoted 2 times

🗨️ 👤 **AndreFR** 10 months, 1 week ago

**Selected Answer: B**

B is correct, no new tables, and minimally interrupting other teams in the organization

A & E excluded, because they interrupt other teams in the organisation, usually answer that require user communication are wrong answers.

C excluded, because it's used for table creation, not after creation

D excluded because it increases the number of tables

upvoted 1 times

🗨️ 👤 **fe3b2fc** 10 months, 1 week ago

**Selected Answer: A**

B,C and D all state creating a new table, therefore increasing the number of tables to manage. This is exactly what the question says to avoid.

"minimally interrupting other teams in the organization without increasing the number of tables that need to be managed"

Answer A is the only one that makes sense and is pretty standard operation procedure for databases. E is wrong because you would never update a column comment to inform users of anything.

upvoted 2 times

🗨️ 👤 **faraaz132** 11 months ago

**Selected Answer: B**

B is correct.

Why not D: Because it will create interruption when you replace the current table with a view and question says minimal interruption

upvoted 2 times

A Delta Lake table representing metadata about content posts from users has the following schema: user\_id LONG, post\_text STRING, post\_id STRING, longitude FLOAT, latitude FLOAT, post\_time TIMESTAMP, date DATE

This table is partitioned by the date column. A query is run with the following filter: longitude < 20 & longitude > -20

Which statement describes how data will be filtered?

- A. Statistics in the Delta Log will be used to identify partitions that might include files in the filtered range.
- B. No file skipping will occur because the optimizer does not know the relationship between the partition column and the longitude.
- C. The Delta Engine will use row-level statistics in the transaction log to identify the files that meet the filter criteria.
- D. Statistics in the Delta Log will be used to identify data files that might include records in the filtered range.
- E. The Delta Engine will scan the parquet file footers to identify each row that meets the filter criteria.

**Suggested Answer: D**

Community vote distribution


D (91%)

9%

 **Enduresoul** Highly Voted 1 year, 7 months ago

**Selected Answer: D**

D is correct. A partition can include multiple files. And the statistics are collected for each file.  
upvoted 12 times

 **KadELbied** Most Recent 1 month, 3 weeks ago


**Selected Answer: D**

suretly D  
upvoted 1 times

 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: B**

Answer B. Single Comparison Filter (e.g., latitude > 66.3): File skipping is highly efficient because Delta can use min/max statistics to directly eliminate files that don't meet the condition.  
Range Filters (e.g., longitude < 20 AND longitude > -20): File skipping is still possible but less efficient, because Delta has to evaluate whether any records in the file might meet the condition, even if the min and max values of the column in the file overlap with the filter range.  
So in summary, file skipping works best with single comparisons like latitude > 66.3 but is less effective with range filters like longitude < 20 AND longitude > -20.  
upvoted 1 times

 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: D**

Do not get confused between option c and d. Given answer is correct.  
upvoted 1 times

 **hebied** 7 months ago

**Selected Answer: D**

D is more suitable  
upvoted 1 times

 **AndreFR** 10 months, 1 week ago



**Selected Answer: D**

Min and max values of each parquet file are stored in Delta Logs  
Delta data skipping automatically collects the stats (min, max, etc.) for the first 32 columns for each underlying Parquet file when you write data into a Delta table. Databricks takes advantage of this information (minimum and maximum values) at query time to skip unnecessary files in order to speed up the queries.  
<https://www.databricks.com/discover/pages/optimize-data-workloads-guide#delta-data>  
upvoted 2 times

 **AziLa** 1 year, 5 months ago

Correct Ans is D



upvoted 2 times

  **Quadronoid** 1 year, 8 months ago

**Selected Answer: C**



I guess C option is right since transaction log contains information about max/min values of first 32 columns, it can be used in order to filter files.

upvoted 1 times

  **Quadronoid** 1 year, 8 months ago

I reread the question and thing that I made a mistake, in option C there is information about row-level statistics, but, I guess, statistics in Delta Log it is more less about columns. So, now D looks fine for me.

upvoted 4 times

  **sturcu** 1 year, 8 months ago

**Selected Answer: D**

D is Correct

upvoted 3 times

A small company based in the United States has recently contracted a consulting firm in India to implement several new data engineering pipelines to power artificial intelligence applications. All the company's data is stored in regional cloud storage in the United States. The workspace administrator at the company is uncertain about where the Databricks workspace used by the contractors should be deployed. Assuming that all data governance considerations are accounted for, which statement accurately informs this decision?

- A. Databricks runs HDFS on cloud volume storage; as such, cloud virtual machines must be deployed in the region where the data is stored.
- B. Databricks workspaces do not rely on any regional infrastructure; as such, the decision should be made based upon what is most convenient for the workspace administrator.
- C. Cross-region reads and writes can incur significant costs and latency; whenever possible, compute should be deployed in the same region the data is stored.
- D. Databricks leverages user workstations as the driver during interactive development; as such, users should always use a workspace deployed in a region they are physically near.
- E. Databricks notebooks send all executable code from the user's browser to virtual machines over the open internet; whenever possible, choosing a workspace region near the end users is the most secure.

**Suggested Answer:** C

Community vote distribution

C (83%)

B (17%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

suretly C

upvoted 1 times

🗳️ 👤 **RandomForest** 5 months, 2 weeks ago

**Selected Answer: C**

C is the correct answer.

upvoted 1 times

🗳️ 👤 **imatheushenrique** 6 months, 4 weeks ago

(C)

The decision is about where the Databricks workspace used by the contractors should be deployed. The contractors are based in India, while all the company's data is stored in regional cloud storage in the United States. When choosing a region for deploying a Databricks workspace, one of the important factors to consider is the proximity to the data sources and sinks. Cross-region reads and writes can incur significant costs and latency due to network bandwidth and data transfer fees. Therefore, whenever possible, compute should be deployed in the same region the data is stored to optimize performance and reduce costs

upvoted 2 times

🗳️ 👤 **spaceexplorer** 11 months, 1 week ago

**Selected Answer: C**

C is the answer.

upvoted 3 times

🗳️ 👤 **RafaelCFC** 12 months ago

**Selected Answer: C**

An important part of data governance is usage cost, and, as a general data engineering practice, egress costs related to moving data between regions is always an important consideration. Having the workspaces located in a different region than the contractors will incur to them in very little nuisance, while greatly saving in this sense.

upvoted 2 times

🗳️ 👤 **Patito** 1 year ago

**Selected Answer: B**

From where data engineering team develops pipelines is independent of where the data objects reside in the cloud storage.

upvoted 1 times

🗳️ 👤 **coercion** 7 months, 1 week ago

These pipelines will create clusters (machines) which will reside in a different region than the data and that will cause latency issues. So C should be the correct option.

upvoted 2 times

  **chokthewa** 1 year, 2 months ago

C is correct.

upvoted 2 times



The downstream consumers of a Delta Lake table have been complaining about data quality issues impacting performance in their applications. Specifically, they have complained that invalid latitude and longitude values in the activity\_details table have been breaking their ability to use other geolocation processes.

A junior engineer has written the following code to add CHECK constraints to the Delta Lake table:

```
ALTER TABLE activity_details
ADD CONSTRAINT valid_coordinates
CHECK (
  latitude >= -90 AND
  latitude <= 90 AND
  longitude >= -180 AND
  longitude <= 180);
```

A senior engineer has confirmed the above logic is correct and the valid ranges for latitude and longitude are provided, but the code fails when executed.

Which statement explains the cause of this failure?

- A. Because another team uses this table to support a frequently running application, two-phase locking is preventing the operation from committing.
- B. The activity\_details table already exists; CHECK constraints can only be added during initial table creation.
- C. The activity\_details table already contains records that violate the constraints; all existing data must pass CHECK constraints in order to add them to an existing table.
- D. The activity\_details table already contains records; CHECK constraints can only be added prior to inserting values into a table.
- E. The current table schema does not contain the field valid\_coordinates; schema evolution will need to be enabled before altering the table to add a constraint.

**Suggested Answer: B**

Community vote distribution

C (100%)

🗳️ 👤 **8605246** Highly Voted 1 year, 10 months ago

incorrect the correct option is C, with constraints, if added to an existing table the existing data in the table must be consistent with the constraint otherwise it fails

<https://docs.databricks.com/en/sql/language-manual/sql-ref-syntax-ddl-alter-table.html#add-constraint>

upvoted 13 times

🗳️ 👤 **KadELbied** Most Recent 1 month, 3 weeks ago

Selected Answer: C

suretly C

upvoted 1 times

🗳️ 👤 **AndreFR** 10 months, 1 week ago

Selected Answer: C

-- CREATE TABLE

create table test\_constraint (t1 varchar(2), n1 int);

-- ADD VALUE

insert into test\_constraint values ('v3', 3);

-- ADD CONSTRAINT VIOLATED BY CURRENT DATA

-- should throw error : 1 row in spark\_catalog.default.test\_constraint violate the new CHECK constraint (n1 < 3)

alter table test\_constraint add constraint valid\_n1 check (n1 < 3);

-- ADD CONSTRAINT NOT VIOLATED BY CURRENT DATA (no error)

alter table test\_constraint add constraint valid\_n1 check (n1 < 100);

upvoted 1 times

🗄️ 👤 **faraaz132** 11 months ago

**Selected Answer: C**

C is correct.

upvoted 1 times

🗄️ 👤 **PrashantTiwari** 1 year, 4 months ago

C is correct

upvoted 1 times

🗄️ 👤 **DAN\_H** 1 year, 4 months ago

correct ans is C

upvoted 1 times

🗄️ 👤 **AziLa** 1 year, 5 months ago

correct ans is C

upvoted 1 times

🗄️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: C**

correct

upvoted 1 times

🗄️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: C**

C is the correct answer

upvoted 1 times

🗄️ 👤 **Patito** 1 year, 6 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗄️ 👤 **hamzaKhribi** 1 year, 6 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗄️ 👤 **Enduresoul** 1 year, 7 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗄️ 👤 **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: C**

When adding a CHECK constraint to an existing table, the operation will fail if there are any rows in the table that do not meet the constraint. Before a CHECK constraint can be added, the data already in the table must be validated to ensure that it complies with the constraint conditions. If any existing records violate the new constraints, they must be corrected or removed before the ALTER TABLE command can be successfully executed.

upvoted 2 times

🗄️ 👤 **BIKRAM063** 1 year, 7 months ago

**Selected Answer: C**

Correct option C : existing data violated check constraint condition

upvoted 1 times

🗄️ 👤 **Quadronoid** 1 year, 8 months ago

**Selected Answer: C**

Right answer is C

upvoted 1 times

🗄️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: C**

C - table already has data

upvoted 1 times

🗄️ 👤 **MarceloManhaes** 1 year, 9 months ago

Yes the correct is option C

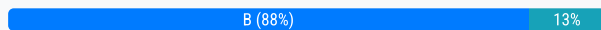
upvoted 1 times

Which of the following is true of Delta Lake and the Lakehouse?

- A. Because Parquet compresses data row by row, strings will only be compressed when a character is repeated multiple times.
- B. Delta Lake automatically collects statistics on the first 32 columns of each table which are leveraged in data skipping based on query filters.
- C. Views in the Lakehouse maintain a valid cache of the most recent versions of source tables at all times.
- D. Primary and foreign key constraints can be leveraged to ensure duplicate values are never entered into a dimension table.
- E. Z-order can only be applied to numeric values stored in Delta Lake tables.

**Suggested Answer: B**

Community vote distribution



🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

Surely B

upvoted 1 times

🗳️ 👤 **SRV\_33** 5 months, 2 weeks ago

**Selected Answer: B**

Complete statement is correct only in this option

upvoted 1 times

🗳️ 👤 **PrashantTiwari** 1 year, 4 months ago

B is correct

upvoted 1 times

🗳️ 👤 **guillesd** 1 year, 4 months ago

**Selected Answer: B**

B is correct

upvoted 2 times

🗳️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: B**

B is correct

upvoted 1 times

🗳️ 👤 **Crocjun** 1 year, 5 months ago

Can anyone explain why D is not correct?

upvoted 1 times

🗳️ 👤 **decisiontree** 5 months, 4 weeks ago

Foreign key constraints have nothing to do with the duplicate values.

upvoted 1 times

🗳️ 👤 **cryptoflam** 1 year, 5 months ago

Because Primary & Foreign Key information is not enforced.

"Primary and foreign keys are informational only and are not enforced" from:

<https://docs.databricks.com/en/tables/constraints.html#declare-primary-key-and-foreign-key-relationships>

upvoted 2 times

🗳️ 👤 **Patito** 1 year, 6 months ago

**Selected Answer: B**

B is correct since statistics are collected for the first 32 columns and stored in the transaction log.


upvoted 3 times

🗳️ 👤 **AndreFR** 10 months, 1 week ago

<https://www.databricks.com/discover/pages/optimize-data-workloads-guide#delta-data>

Delta data skipping automatically collects the stats (min, max, etc.) for the first 32 columns for each underlying Parquet file when you write data into a Delta table. Databricks takes advantage of this information (minimum and maximum values) at query time to skip unnecessary files in order to speed up the queries.

upvoted 1 times



  **ervinshang** 1 year, 6 months ago

**Selected Answer: B**

B is correct

C is error, can't have new cache in view



upvoted 1 times

  **f728f7f** 1 year, 6 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

  **chokthewa** 1 year, 8 months ago

B is correct.

<https://docs.delta.io/2.0.0/table-properties.html>

upvoted 1 times

The view updates represents an incremental batch of all newly ingested data to be inserted or updated in the customers table.

The following logic is used to process these records.

```

MERGE INTO customers
USING (
  SELECT updates.customer_id as merge_key, updates.*
  FROM updates

  UNION ALL

  SELECT NULL as merge_key, updates.*
  FROM updates JOIN customers
  ON updates.customer_id = customers.customer_id
  WHERE customers.current = true AND updates.address <> customers.address
) staged_updates
ON customers.customer_id = mergeKey
WHEN MATCHED AND customers.current = true AND customers.address <> staged_updates.address THEN
  UPDATE SET current = false, end_date = staged_updates.effective_date
WHEN NOT MATCHED THEN
  INSERT(customer_id, address, current, effective_date, end_date)
  VALUES(staged_updates.customer_id, staged_updates.address, true, staged_updates.effective_date,
  null)

```

Which statement describes this implementation?

- A. The customers table is implemented as a Type 3 table; old values are maintained as a new column alongside the current value.
- B. The customers table is implemented as a Type 2 table; old values are maintained but marked as no longer current and new values are inserted.
- C. The customers table is implemented as a Type 0 table; all writes are append only with no changes to existing values.
- D. The customers table is implemented as a Type 1 table; old values are overwritten by new values and no history is maintained.
- E. The customers table is implemented as a Type 2 table; old values are overwritten and new customers are appended.

**Suggested Answer: B**

Community vote distribution

B (100%)

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

Suretly B

upvoted 1 times

 **Tayari** 8 months ago

B is correct

upvoted 1 times

 **imatheushenrique** 1 year ago

B. The customers table is implemented as a Type 2 table; old values are maintained but marked as no longer current and new values are inserted.

A Type 1 table does not track changes in dimensional attributes - the new value overwrites the existing value. Here, we do not preserve historical changes in data.

A Type 2 Table tracks change over time by creating new rows for each change. A new dimension record is inserted with a high-end date or one with NULL. The previous record is "closed" with an end date. This approach maintains a complete history of changes and allows for as-was reporting use cases.

A data warehousing method called Slowly Changing Dimension (SCD) Type 3 is used to track both the old and new values while managing historical changes in data over time. To reflect the historical and present values of an attribute, SCD Type 3 keeps two extra columns in the dimension table.


upvoted 3 times

 **spaceexplorer** 1 year, 5 months ago

Selected Answer: B

B is correct

upvoted 3 times

  **kz\_data** 1 year, 5 months ago

Selected Answer: B

B is correct

upvoted 1 times

  **chokthewa** 1 year, 8 months ago

B is correct.

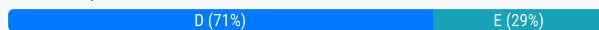
upvoted 2 times

The DevOps team has configured a production workload as a collection of notebooks scheduled to run daily using the Jobs UI. A new data engineering hire is onboarding to the team and has requested access to one of these notebooks to review the production logic. What are the maximum notebook permissions that can be granted to the user without allowing accidental changes to production code or data?

- A. Can Manage
- B. Can Edit
- C. No permissions
- D. Can Read
- E. Can Run

**Suggested Answer: D**

*Community vote distribution*



🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: D**

Surely D

upvoted 1 times

🗳️ 👤 **AlHerd** 3 months ago

**Selected Answer: D**

We don't want to person to run the code in the notebooks as this might change the data, so "can read" is best.

upvoted 1 times

🗳️ 👤 **RasipalayamRDK** 5 months, 1 week ago

**Selected Answer: E**

Maximum permission Can Run .<https://docs.databricks.com/en/notebooks/notebooks-collaborate.html>

upvoted 1 times

🗳️ 👤 **arekm** 5 months, 4 weeks ago

**Selected Answer: D**

I change my mind - answer D. D would still allow to change data (provided the notebook changes the data), however it would require using importing from other notebook, which is more of a nuisance than just clicking Run in the notebook.

upvoted 1 times

🗳️ 👤 **arekm** 5 months, 4 weeks ago

**Selected Answer: E**

Answer E - the question states "maximum notebook permissions", which means "Can Run", see:

<https://docs.databricks.com/en/security/auth/access-control/index.html#notebook-acls>

upvoted 1 times

🗳️ 👤 **benni\_ale** 8 months ago

**Selected Answer: D**

It happen that by only running a production workload scheduled to run only once a day , some directories got removed. I would say D as by only running a program when u should not you cold commit changes to production data.

upvoted 1 times

🗳️ 👤 **nedlo** 8 months, 1 week ago

**Selected Answer: D**

Can Read, because Running could have changed DATA

upvoted 2 times

🗳️ 👤 **olly24** 9 months, 3 weeks ago

Correct answer is C. <https://learn.microsoft.com/en-us/azure/databricks/security/auth/access-control/#--notebook-acls>

upvoted 1 times



🗳️ 👤 **Melik3** 10 months, 1 week ago



**Selected Answer: E**

can run is the correct answer here because the question asked for the maximum possible permission without editing.

upvoted 2 times

  **shynkary** 9 months, 3 weeks ago

disagree, it says "changes to production code or data". Can Run permission allows to run a workflow which will cause changes in data



upvoted 3 times

  **AndreFR** 10 months, 1 week ago

**Selected Answer: D**

<https://docs.databricks.com/en/security/auth/access-control/index.html#notebook-acls>



upvoted 2 times

  **alexvno** 1 year, 7 months ago

**Selected Answer: D**

Correct

upvoted 3 times

  **Quadronoid** 1 year, 8 months ago

**Selected Answer: D**

Correct, D

upvoted 1 times

A table named user\_ltv is being used to create a view that will be used by data analysts on various teams. Users in the workspace are configured into groups, which are used for setting up data access using ACLs.

The user\_ltv table has the following schema:

email STRING, age INT, ltv INT

The following view definition is executed:

```
CREATE VIEW email_ltv AS
SELECT
CASE WHEN
    is_member('marketing') THEN email
    ELSE 'REDACTED'
END AS email,
    ltv
FROM user_ltv
```

An analyst who is not a member of the marketing group executes the following query:

```
SELECT * FROM email_ltv -
```

Which statement describes the results returned by this query?

- A. Three columns will be returned, but one column will be named "REDACTED" and contain only null values.
- B. Only the email and ltv columns will be returned; the email column will contain all null values.
- C. The email and ltv columns will be returned with the values in user\_ltv.
- D. The email, age, and ltv columns will be returned with the values in user\_ltv.
- E. Only the email and ltv columns will be returned; the email column will contain the string "REDACTED" in each row.

**Suggested Answer: E**

Community vote distribution

E (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

🗳️ 👤 **AndreFR** 10 months, 1 week ago

**Selected Answer: E**

A, D incorrect because 2 columns email & ltv are returned.

B incorrect because email will not always contain null values (unless email is null)

The user is not a member of "marketing", so 3 is the correct answer. If the user were a member of "marketing" group, correct answer would have been C

upvoted 2 times

🗳️ 👤 **Isio05** 1 year ago

**Selected Answer: E**

E, only email column is selected and is not allowed to be viewed by the user

upvoted 1 times

🗳️ 👤 **alexvno** 1 year, 7 months ago

**Selected Answer: E**

sure E

upvoted 2 times

🗳️ 👤 **ismoshkov** 1 year, 7 months ago

**Selected Answer: E**

E is correct

upvoted 2 times

  **chokthewa** 1 year, 8 months ago

E is correct.

upvoted 1 times

The data governance team has instituted a requirement that all tables containing Personal Identifiable Information (PH) must be clearly annotated. This includes adding column comments, table comments, and setting the custom table property "contains\_pii" = true.

The following SQL DDL statement is executed to create a new table:

```
CREATE TABLE dev.pii_test
(id INT, name STRING COMMENT "PII")
COMMENT "Contains PII"
TBLPROPERTIES ('contains_pii' = True)
```

Which command allows manual confirmation that these three requirements have been met?

- A. DESCRIBE EXTENDED dev.pii\_test
- B. DESCRIBE DETAIL dev.pii\_test
- C. SHOW TBLPROPERTIES dev.pii\_test
- D. DESCRIBE HISTORY dev.pii\_test
- E. SHOW TABLES dev

**Suggested Answer: A**

Community vote distribution

A (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times

🗳️ 👤 **Tedet** 3 months, 4 weeks ago

**Selected Answer: A**

C is wrong since Property value returned by this statement excludes some properties that are internal to spark and hive. The excluded properties are:

All the properties that start with prefix spark.sql

Property keys such as: EXTERNAL, comment

All the properties generated internally by hive to store statistics. Some of these properties are: numFiles, numPartitions, numRows.

upvoted 1 times

🗳️ 👤 **RandomForest** 5 months, 2 weeks ago

**Selected Answer: A**

Answer A is correct as explained by lexaneon

upvoted 1 times

🗳️ 👤 **lexaneon** 12 months ago

looks like A & C are correct.. <https://docs.databricks.com/en/sql/language-manual/sql-ref-syntax-aux-show-tblproperties.html#show-tblproperties>

upvoted 3 times

🗳️ 👤 **lexaneon** 12 months ago

if we want to see also columns comments then A

upvoted 5 times

🗳️ 👤 **rok21** 1 year ago

**Selected Answer: A**

correct A !

upvoted 4 times

The data governance team is reviewing code used for deleting records for compliance with GDPR. They note the following logic is used to delete records from the Delta Lake table named users.

```
DELETE FROM users
WHERE user_id IN
  (SELECT user_id FROM delete_requests)
```

Assuming that user\_id is a unique identifying key and that delete\_requests contains all users that have requested deletion, which statement describes whether successfully executing the above logic guarantees that the records to be deleted are no longer accessible and why?

- A. Yes; Delta Lake ACID guarantees provide assurance that the DELETE command succeeded fully and permanently purged these records.
- B. No; the Delta cache may return records from previous versions of the table until the cluster is restarted.
- C. Yes; the Delta cache immediately updates to reflect the latest data files recorded to disk.
- D. No; the Delta Lake DELETE command only provides ACID guarantees when combined with the MERGE INTO command.
- E. No; files containing deleted records may still be accessible with time travel until a VACUUM command is used to remove invalidated data files.

**Suggested Answer: E**

Community vote distribution

E (100%)

🗳️ 👤 **kz\_data** 11 months, 3 weeks ago

**Selected Answer: E**

E is correct

upvoted 4 times

🗳️ 👤 **ervinshang** 1 year ago

**Selected Answer: E**

E is correct.

upvoted 3 times

🗳️ 👤 **alexvno** 1 year, 1 month ago

**Selected Answer: E**

Correct

upvoted 2 times

🗳️ 👤 **chokthewa** 1 year, 2 months ago

E is correct.

upvoted 2 times

An external object storage container has been mounted to the location /mnt/finance\_eda\_bucket.

The following logic was executed to create a database for the finance team:

```
CREATE DATABASE finance_eda_db
LOCATION '/mnt/finance_eda_bucket';
GRANT USAGE ON DATABASE finance_eda_db TO finance;
GRANT CREATE ON DATABASE finance_eda_db TO finance;
```

After the database was successfully created and permissions configured, a member of the finance team runs the following code:

```
CREATE TABLE finance_eda_db.tx_sales AS
SELECT *
FROM sales
WHERE state = "TX";
```

If all users on the finance team are members of the finance group, which statement describes how the tx\_sales table will be created?

- A. A logical table will persist the query plan to the Hive Metastore in the Databricks control plane.
- B. An external table will be created in the storage container mounted to /mnt/finance\_eda\_bucket.
- C. A logical table will persist the physical plan to the Hive Metastore in the Databricks control plane.
- D. An managed table will be created in the storage container mounted to /mnt/finance\_eda\_bucket.
- E. A managed table will be created in the DBFS root storage container.

**Suggested Answer: B**

Community vote distribution

D (65%)

E (35%)

  **tkg13** Highly Voted 1 year, 10 months ago

Correct Answer D

<https://docs.databricks.com/en/data-governance/unity-catalog/create-schemas.html#language-SQL>

upvoted 11 times

  **cotardo2077** 1 year, 9 months ago

you are right, it is managed table

upvoted 2 times

  **CertPeople** 1 year, 9 months ago

Nope, you are talking about MANAGED LOCATION (from Unity). In the question states LOCATION (not Unity based), which is not managed

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

upvoted 2 times

  **CertPeople** 1 year, 9 months ago

Effectively doing a test on one of my clusters the table is MANAGED

upvoted 9 times

  **MarceloManhaes** Highly Voted 1 year, 9 months ago

Every unmanaged(external) table creation needs to put keyword LOCATION despite if database, that table resides, is put with LOCATION sententece. So B is incorrect. D is correct because the sentence to creates the table is a managed table.

<https://docs.databricks.com/en/lakehouse/data-objects.html>

upvoted 7 times

  **ASRCA** Most Recent 5 months, 3 weeks ago

**Selected Answer: B**

This is because the storage container is mounted to /mnt/finance\_eda\_bucket, and the code executed by the finance team member would create an external table in that location.

upvoted 2 times

  **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: D**

Answer D. The use of the LOCATION clause with a DBFS path (/mnt/finance\_eda\_bucket) suggests that Hive Metastore and DBFS location are being used. The answer is correct in the context of Hive Metastore and DBFS location, but if Unity Catalog (UC) is in use, the result would be an external table, not a managed one.

upvoted 1 times

🗲️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: D**

No USE DATABASE statement otherwise it would have been external

upvoted 1 times

🗲️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: D**

No USE DATABASE statement otherwise it would have been external

upvoted 1 times

🗲️ 👤 **coercion** 1 year, 1 month ago

**Selected Answer: D**

D as the word LOCATION is not specified. Although the data will be stored in an external location but the table will still be a managed table.

upvoted 2 times

🗲️ 👤 **Curious76** 1 year, 4 months ago

**Selected Answer: E**

E is correct coz this table is managed on top of the external source file. a managed tables are stored on DBFS.

upvoted 2 times

🗲️ 👤 **hal2401me** 1 year, 4 months ago

**Selected Answer: D**

Correct answer D.

just did a test. As with DBR12.2, UC databases are not supported with location on dbfs, but s3/abfss. However, Hive\_metastore databases are supported with location on dbfs. Then, a table created in this database IS a managed table, as verified with describe extend command.

upvoted 5 times

🗲️ 👤 **s\_villahermosa91** 1 year, 5 months ago

**Selected Answer: E**

Correct Answer E

upvoted 2 times

🗲️ 👤 **kz\_data** 1 year, 5 months ago

**Selected Answer: D**

D is the correct answer, the table created is a managed table and not external, and it will be located under the location defined in the database's creation DDL.

upvoted 1 times

🗲️ 👤 **azurelearn2020** 1 year, 6 months ago

**Selected Answer: D**

It will be a managed table created under specified database. Location keyword used for database will make sure all the managed tables are stored in database location.

upvoted 2 times

🗲️ 👤 **Enduresoul** 1 year, 7 months ago

**Selected Answer: D**

D is correct. The table will be created as managed, because no LOCATION is specified on table creation. The table will be created in the location specified with database creation

upvoted 4 times

🗲️ 👤 **Dileepvikram** 1 year, 7 months ago

I think the answer id D

upvoted 1 times

🗲️ 👤 **PearApple** 1 year, 7 months ago



I followed the steps to create schema and table, the answer is D

upvoted 2 times

🗲️ 👤 **jerborder** 1 year, 8 months ago

Correct answer is D. "data for a managed table resides in the location of the database it is registered to

upvoted 1 times

  **sturcu** 1 year, 8 months ago

**Selected Answer: E**

A managed table will be created on DBFS.

upvoted 2 times

  **spudteo** 1 year, 2 months ago

The LOCATION of a database will determine the default location for data of all tables registered to that database.

from the documentation

upvoted 2 times



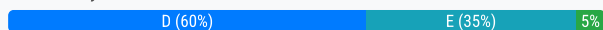
Although the Databricks Utilities Secrets module provides tools to store sensitive credentials and avoid accidentally displaying them in plain text users should still be careful with which credentials are stored here and which users have access to using these secrets.

Which statement describes a limitation of Databricks Secrets?

- A. Because the SHA256 hash is used to obfuscate stored secrets, reversing this hash will display the value in plain text.
- B. Account administrators can see all secrets in plain text by logging on to the Databricks Accounts console.
- C. Secrets are stored in an administrators-only table within the Hive Metastore; database administrators have permission to query this table by default.
- D. Iterating through a stored secret and printing each character will display secret contents in plain text.
- E. The Databricks REST API can be used to list secrets in plain text if the personal access token has proper credentials.

**Suggested Answer: D**

Community vote distribution



🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

🗳️ 👤 **Tedet** 3 months, 4 weeks ago

**Selected Answer: D**

Secret redaction

Storing credentials as Databricks secrets makes it easy to protect your credentials when you run notebooks and jobs. However, it is easy to accidentally print a secret to standard output buffers or display the value during variable assignment.

upvoted 1 times

🗳️ 👤 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: D**

Answer D. `dbutils.secrets.get(scope="myScope", key="myKey")` retrieves the plain text value of a secret, which is then available for use in code.

Limitation: Once the secret is retrieved, if improperly handled (e.g., logged or iterated), its plain text value can be exposed. Option E: The REST API can list secrets in plain text if proper credentials (e.g., a personal access token) are provided. This is unrelated to `dbutils.secrets.get` but is a valid limitation of the overall secrets management framework in Databricks. Note that the difference between Option D or E is if it is a limitation related to Databricks Utilities Secret (`dbutils.secrets`), in this case option D is the correct option.

upvoted 1 times

🗳️ 👤 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: D**

Cannot be option E as it justs lists the Secret value. It does not print the content therein

upvoted 1 times

🗳️ 👤 **fe3b2fc** 10 months, 1 week ago

**Selected Answer: D**

`value = dbutils.secrets.get(scope="myScope", key="myKey")`

for char in value:

`print(char, end=" ")`

Out:

`y o u r _ v a l u e`

upvoted 4 times

🗳️ 👤 **coercion** 1 year, 1 month ago

**Selected Answer: E**

Only through REST API or CLI you can fetch the secret if you have valid token

upvoted 2 times

🗨️ 👤 **Er5** 1 year, 2 months ago

E: <https://docs.databricks.com/api/azure/workspace/secrets/listsecrets>

GET /api/2.0/secrets/list won't list secrets in plain text.

D: if print it without iterating it in a for loop the output is kind of encrypted where it is showing [REDACTED]. But, if I do it as shown in the screenshot, I'm able to see the value of the secret key.

<https://community.databricks.com/t5/data-engineering/how-to-avoid-databricks-secret-scope-from-exposing-the-value-of/td-p/12254>

<https://docs.databricks.com/en/security/secrets/redaction.html>

Secret redaction for notebook cell output applies only to literals. The secret redaction functionality does not prevent deliberate and arbitrary transformations of a secret literal.

upvoted 2 times

🗨️ 👤 **Lucario95** 1 year, 4 months ago

**Selected Answer: E**

Both D and E seems correct.

They are poorly written thought because for D just printing the characters (not separated by spaces, newlines or something) would not work, while E if launched inside databricks workspace would not work neither.

upvoted 2 times

🗨️ 👤 **PrashantTiwari** 1 year, 4 months ago

D is correct

upvoted 2 times

🗨️ 👤 **guillesd** 1 year, 4 months ago

**Selected Answer: D**

D is for sure correct (tried it several times on a Databricks environment).

upvoted 2 times

🗨️ 👤 **guillesd** 1 year, 4 months ago

Regarding E, it can list secrets (with scopes) but I am not sure it can list secret contents.

upvoted 1 times

🗨️ 👤 **DAN\_H** 1 year, 4 months ago

**Selected Answer: D**

D is correct

upvoted 3 times

🗨️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: D**

D is correct

upvoted 2 times

🗨️ 👤 **Def21** 1 year, 5 months ago

**Selected Answer: E**

At least E is a correct answer.

B: You can't see secrets in Admin console. Only via REST API, CLI etc.

C: Secrets are. not stored in Hive Metastore.

D: I am not sure if iterating through secret character by character would work?

E: This is at least correct. Using this.

upvoted 1 times

🗨️ 👤 **ranith** 1 year, 5 months ago

B and E both seems to be correct:

<https://community.databricks.com/t5/data-engineering/how-to-avoid-databricks-secret-scope-from-exposing-the-value-of/td-p/12254/page/2>

upvoted 1 times

🗨️ 👤 **Jay\_98\_11** 1 year, 5 months ago

**Selected Answer: D**

For sure it's D

upvoted 2 times

🗨️ 👤 **hkay** 1 year, 6 months ago



Answer is E:

```
/api/2.0/secrets/get
```

```
{  
  "key": "string",  
  "value": "string"  
}
```

The REST API can potentially expose secrets in plain text if a user with appropriate permissions (including access to both secrets/list and secrets/get) uses a personal access token.

upvoted 3 times

  **Patito** 1 year, 6 months ago

**Selected Answer: D**

Iterating through the secrets provides a way to see the secret's password.

upvoted 2 times

What statement is true regarding the retention of job run history?

- A. It is retained until you export or delete job run logs
- B. It is retained for 30 days, during which time you can deliver job run logs to DBFS or S3
- C. It is retained for 60 days, during which you can export notebook run results to HTML
- D. It is retained for 60 days, after which logs are archived
- E. It is retained for 90 days or until the run-id is re-used through custom run configuration

**Suggested Answer: B**

Community vote distribution



**stuart\_gta1** 1 year, 4 months ago

B is wrong, Should be C.  
upvoted 9 times

**Yogi05** 1 year ago

C is correct answer. <https://docs.databricks.com/en/workflows/jobs/monitor-job-runs.html>  
upvoted 7 times

**KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

suretly C  
upvoted 1 times

**Tedet** 3 months, 4 weeks ago

**Selected Answer: C**

To export notebook run results for a job with a single task:

On the job detail page, click the View Details link for the run in the Run column of the Completed Runs (past 60 days) table.  
Click Export to HTML.

To export notebook run results for a job with multiple tasks:

On the job detail page, click the View Details link for the run in the Run column of the Completed Runs (past 60 days) table.  
Click the notebook task to export.  
Click Export to HTML.  
upvoted 1 times

**Tedet** 3 months, 4 weeks ago

**Selected Answer: C**

Databricks maintains a history of your job runs for up to 60 days. If you need to preserve job runs, Databricks recommends exporting results before they expire.  
upvoted 1 times

**janeZ** 6 months, 1 week ago

**Selected Answer: C**

<https://learn.microsoft.com/en-us/azure/databricks/jobs/monitor>  
upvoted 1 times

**hal2401me** 10 months, 1 week ago

**Selected Answer: C**

<https://learn.microsoft.com/en-us/azure/databricks/workflows/jobs/monitor-job-runs>

Azure Databricks maintains a history of your job runs for up to 60 days. If you need to preserve job runs, Databricks recommends exporting results before they expire. For more information, see Export job run results.  
upvoted 2 times

**ATLTennis** 11 months, 4 weeks ago

**Selected Answer: C**

C is the correct answer  
upvoted 3 times

  **Patito** 1 year ago

**Selected Answer: C**

c is correct  
upvoted 2 times

  **SwastikaM** 1 year ago

Option C is correct  
upvoted 2 times

  **f728f7f** 1 year ago

**Selected Answer: D**

A secret CAN be printer character-by-character, so it's not really that secure.  
upvoted 1 times

  **f728f7f** 1 year ago

Whoops, answer meant for previous question in the bank. Admin, please delete or move.  
upvoted 1 times

  **rok21** 1 year ago



**Selected Answer: C**

C is correct  
upvoted 3 times

  **azurelearn2020** 1 year ago

**Selected Answer: C**

Correct Answer is C  
upvoted 1 times

  **sturcu** 1 year, 2 months ago

**Selected Answer: C**

C is correct: retention is 60 days and export to html  
upvoted 1 times

  **8605246** 1 year, 4 months ago

this is incorrect databricks maintains a history of job runs for 60 days<https://docs.databricks.com/en/workflows/jobs/monitor-job-runs.html#:~:text=Databricks%20maintains%20a%20history%20of,see%20Export%20job%20run%20results>.  
upvoted 3 times

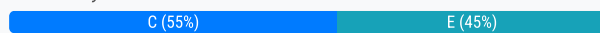
A data engineer, User A, has promoted a new pipeline to production by using the REST API to programmatically create several jobs. A DevOps engineer, User B, has configured an external orchestration tool to trigger job runs through the REST API. Both users authorized the REST API calls using their personal access tokens.

Which statement describes the contents of the workspace audit logs concerning these events?

- A. Because the REST API was used for job creation and triggering runs, a Service Principal will be automatically used to identify these events.
- B. Because User B last configured the jobs, their identity will be associated with both the job creation events and the job run events.
- C. Because these events are managed separately, User A will have their identity associated with the job creation events and User B will have their identity associated with the job run events.
- D. Because the REST API was used for job creation and triggering runs, user identity will not be captured in the audit logs.
- E. Because User A created the jobs, their identity will be associated with both the job creation events and the job run events.

**Suggested Answer: C**

Community vote distribution



**hal2401me** Highly Voted 1 year, 4 months ago

**Selected Answer: E**

<https://docs.databricks.com/api/azure/workspace/jobs/create>

API/jobs/create:run\_as

object

Write-only setting, available only in Create/Update/Reset and Submit calls. Specifies the user or service principal that the job runs as. If not specified, the job runs as the user who created the job.

In the question, it's not stated that user A creates a service principal. So runas can only be himself.

upvoted 9 times

**carlosmps** 6 months, 3 weeks ago

but the documentation says:

The REST API operation path, such as /api/2.0/clusters/get, to get information for the specified cluster.

Remember User B use his token to orchestration.

The answer should be C

upvoted 2 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: C**

suretly C

upvoted 1 times

**capt2101akash** 3 months ago

**Selected Answer: C**

Both uses their own credential for specific tasks

upvoted 1 times

**Nate\_** 3 months, 3 weeks ago

**Selected Answer: C**

User A created the jobs via the REST API using their personal access token, so the workspace audit logs will record these job creation events with User A's identity. Conversely, when User B triggers job runs through the REST API (again, using their own personal access token) via an external orchestration tool, those events will be logged with User B's identity.

upvoted 1 times

**arekm** 5 months, 4 weeks ago

**Selected Answer: C**

Answer C - the run\_as property is not said to be configured, so the job will run with the permissions of the creator - user A. However, still user B will be the one that triggered the run, which is what the question is about.

upvoted 2 times

🗨️ 👤 **LuminaBerry** 6 months, 1 week ago

**Selected Answer: C**

C should be the correct answer.

Although User A has its user associated to the creation, and by default the run as user is omitted on the creation of the job, the question specifies the Audit Logs (Run Event Logs) associated to the run.

I've tried it this out on a job and for a job which was created and has a run as user different from mine, if I go to the run event logs, there are logs which stated that my user triggered a "started" event type

upvoted 2 times

🗨️ 👤 **janeZ** 6 months, 1 week ago

**Selected Answer: C**

based on the standard understanding of how personal access tokens typically work, each user's actions should be logged separately with their respective identities. Therefore, "C" would be the standard answer unless there is a specific behavior or configuration in Databricks that causes the job run events to be attributed back to User A.

upvoted 1 times

🗨️ 👤 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: C**

Answer C. The audit logs distinguish between actions like job creation and job execution, so User A and User B will be identified separately for these actions.

upvoted 1 times

🗨️ 👤 **benni\_ale** 7 months ago

**Selected Answer: E**

I tried myself and E seems correct

upvoted 1 times

🗨️ 👤 **JB90** 7 months ago

**Selected Answer: C**

When you use the API to commit the jobs the creation is logged using the PAT info, the same happens when you start a run using a different PAT.

upvoted 1 times

🗨️ 👤 **benni\_ale** 7 months, 1 week ago

**Selected Answer: E**

Specifies the user, service principal or group that the job/pipeline runs as. If not specified, the job/pipeline runs as the user who created the job/pipeline.

Either user\_name or service\_principal\_name should be specified. If not, an error is thrown.

upvoted 1 times

🗨️ 👤 **rsmf** 7 months, 3 weeks ago

**Selected Answer: C**

C is the right answer

upvoted 1 times

🗨️ 👤 **Carkeys** 8 months, 1 week ago

**Selected Answer: C**

In Databricks, audit logs capture the identity of the user associated with each distinct event, whether it's creating or running a job. Since User A used their personal access token to create the jobs and User B used theirs to trigger job runs, the audit logs will reflect User A's identity for job creation events and User B's identity for job run events.

upvoted 1 times

🗨️ 👤 **quaternion** 10 months, 3 weeks ago

**Selected Answer: E**

By default, jobs run as the identity of the job owner. This means that the job assumes the permissions of the job owner. You can change the identity that the job is running as to a service principal. Then, the job assumes the permissions of that service principal instead of the owner.

<https://docs.databricks.com/en/jobs/create-run-jobs.html#run-a-job-as-a-service-principal>

upvoted 2 times

🗨️ 👤 **spudteo** 1 year, 4 months ago

**Selected Answer: E**

When you create a job your role is IS OWNER and RUN AS. So when you trigger a job, it will run as the RUN AS entity. And it should be user A if someone doesn't have changed it



upvoted 1 times

  **spaceexplorer** 1 year, 5 months ago

**Selected Answer: C**

C is correct

upvoted 3 times

  **rok21** 1 year, 6 months ago

**Selected Answer: C**

C is correct

upvoted 3 times



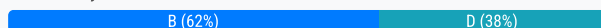
A user new to Databricks is trying to troubleshoot long execution times for some pipeline logic they are working on. Presently, the user is executing code cell-by-cell, using `display()` calls to confirm code is producing the logically correct results as new transformations are added to an operation. To get a measure of average time to execute, the user is running each cell multiple times interactively.

Which of the following adjustments will get a more accurate measure of how code is likely to perform in production?

- A. Scala is the only language that can be accurately tested using interactive notebooks; because the best performance is achieved by using Scala code compiled to JARs, all PySpark and Spark SQL logic should be refactored.
- B. The only way to meaningfully troubleshoot code execution times in development notebooks is to use production-sized data and production-sized clusters with Run All execution.
- C. Production code development should only be done using an IDE; executing code against a local build of open source Spark and Delta Lake will provide the most accurate benchmarks for how code will perform in production.
- D. Calling `display()` forces a job to trigger, while many transformations will only add to the logical query plan; because of caching, repeated execution of the same logic does not provide meaningful results.
- E. The Jobs UI should be leveraged to occasionally run the notebook as a job and track execution time during incremental code development because Photon can only be enabled on clusters launched for scheduled jobs.

**Suggested Answer: D**

Community vote distribution



**guillesd** Highly Voted 1 year, 4 months ago

**Selected Answer: B**

Both B and D are correct statements. However, D is not an adjustment (see the question), it is just an affirmation which happens to be correct. B, however, is an adjustment, and it will definitely help with profiling.

upvoted 7 times

**sammy2025** Most Recent 1 month ago

**Selected Answer: D**

Lazy Evaluation in Spark: Many transformations in Spark only modify the logical query plan until an action (like `display()`, `collect()`, or `write()`) is triggered.

upvoted 1 times

**KadELbied** 1 month, 3 weeks ago

**Selected Answer: B**

suretly B

upvoted 1 times

**Tedet** 3 months, 4 weeks ago

**Selected Answer: D**

Explanation:

Using `display()` in Databricks forces a job to trigger and display the output, which can lead to an inaccurate measure of performance when benchmarking code. This is because `display()` triggers the job and materializes the result, which does not accurately reflect how the code will perform in production when the job is run without the display output.

Additionally, repeated execution of the same logic (with caching) may not give you meaningful performance results since the results are cached in memory and not representative of fresh computations, as they would occur in a production environment.

To get a more accurate measure of execution time, the user should focus on using appropriate job execution techniques, such as running the notebook with "Run All" and avoiding reliance on `display()` calls, which are not representative of how the pipeline would behave in production.

upvoted 3 times

**arekm** 5 months, 4 weeks ago

**Selected Answer: B**

Answer B, see discussion under benni\_ale.

upvoted 1 times

🗨️ 👤 **ultimomassimo** 3 months ago

in any real life commercial project answer B is not feasible, sorry. You always use representative sample, but using same data volumes (especially when they are massive) is impractical and no one would sing off on the cost

upvoted 1 times

🗨️ 👤 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: D**

Answer D. While Option D doesn't directly provide an alternative adjustment, it points out a critical issue in the way interactive notebooks might give misleading results. It would be advisable to avoid using `display()` as a benchmark for performance in production-like environments.

upvoted 3 times

🗨️ 👤 **carlosmps** 6 months, 3 weeks ago

**Selected Answer: B**

Without much thought, I would vote for option B, but since it says 'the ONLY,' it makes me hesitate. While option D only points out the issues with the data engineer's executions, it doesn't really provide the adjustments that need to be made. On the other hand, option B at least gives you a way to simulate production behavior. I'll vote for B, but as I said, the word 'only' makes me doubt, because it's not the only way.

upvoted 1 times

🗨️ 👤 **benni\_ale** 7 months, 1 week ago

**Selected Answer: D**

Answer: D.

Explanation:

Lazy Evaluation: Spark employs lazy evaluation, meaning transformations are not executed until an action (e.g., `display()`, `count()`, `collect()`) is called. Using `display()` triggers the execution of the transformations up to that point.

Caching Effects: Repeatedly executing the same cell can lead to caching, where Spark stores intermediate results. This caching can cause subsequent executions to be faster, not reflecting the true performance of the code.

Why not B:

Production-Sized Data and Clusters: While using production-sized data and clusters (as mentioned in option B) can provide insights into performance, it's not the only way to troubleshoot execution times. Proper testing can often be conducted on smaller datasets and clusters, especially during the development phase.

upvoted 1 times

🗨️ 👤 **af4a20a** 6 months, 3 weeks ago

Yep, what if your production size is 10 TB... But you have a 10GB sample. No idea what's actually right for the test, but D is correct.

upvoted 1 times

🗨️ 👤 **arekm** 5 months, 4 weeks ago

D is correct. However, it does not show direction on what to do to troubleshoot the problem, which is the first statement in the question.

The only way to troubleshoot performance problems is to start with the data & processing platform of size that is representative of production. That is why I think B is a better choice.

upvoted 1 times

🗨️ 👤 **practitioner** 10 months, 2 weeks ago

**Selected Answer: B**

B and D are correct. The question says "which statements" which suggests us that this is a question with multiple choices

upvoted 2 times

🗨️ 👤 **HelixAbdu** 11 months, 1 week ago

Both D and B are correct. But in real life some times clients dose not accept to gave you there production data to test easily. Also it says in B it is "the only way" ans this is not true for me

So i will go with D

upvoted 4 times

🗨️ 👤 **RyanAck24** 9 months ago

I would add to this and say that this \*could\* be a multi-choice question (possibly) as practitioner mentions above. But if it isn't, I would go with D as well.

upvoted 1 times

🗳️ 👤 **ffsdffdsfdfsdfsdf** 1 year, 3 months ago

**Selected Answer: B**

These people voting D have no reading comprehension.  
upvoted 4 times

🗳️ 👤 **alexvno** 1 year, 3 months ago

**Selected Answer: B**

Close env size volumes as possible so results make sense  
upvoted 2 times

🗳️ 👤 **halleysg** 1 year, 3 months ago

**Selected Answer: D**

D is correct  
upvoted 3 times

🗳️ 👤 **Curious76** 1 year, 4 months ago

**Selected Answer: D**

I will go with D  
upvoted 1 times

🗳️ 👤 **agreddy** 1 year, 4 months ago

D is the correct answer

A. Scala is the only language accurately tested using notebooks: Not true. Spark SQL and PySpark can be accurately tested in notebooks, and production performance doesn't solely depend on language choice.

B. Production-sized data and clusters are necessary: While ideal, it's not always feasible for development. Smaller datasets and clusters can provide indicative insights.

C. IDE and local Spark/Delta Lake: Local environments won't replicate production's scale and configuration fully.

E. Jobs UI and Photon: True that Photon benefits scheduled jobs, but Jobs UI can track execution times regardless of Photon usage. However, Jobs UI runs might involve additional overhead compared to notebook cells.

Option D addresses the specific limitations of using `display()` for performance measurement

upvoted 4 times

🗳️ 👤 **DAN\_H** 1 year, 4 months ago

**Selected Answer: D**

As B not talking about how to deal with `display()` function. We know that way to testing performance for the whole notebook need to avoid using `display` as it is way to test the code and `display` the data  
upvoted 3 times

🗳️ 👤 **arekm** 5 months, 4 weeks ago

True, it is not addressing the `display()` function. However, D does not give any hint on how to go about the problem. On top of that `display()` function is an action that might help you out in investigating by triggering the actual processing. You still need the data volume that represents the inherent problem - which means that you need the production size of the data, which I think is the first step anyway. Not the last though :)

upvoted 1 times

🗳️ 👤 **zzzzx** 1 year, 5 months ago

B is correct  
upvoted 1 times

A production cluster has 3 executor nodes and uses the same virtual machine type for the driver and executor.

When evaluating the Ganglia Metrics for this cluster, which indicator would signal a bottleneck caused by code executing on the driver?


- A. The five Minute Load Average remains consistent/flat
- B. Bytes Received never exceeds 80 million bytes per second
- C. Total Disk Space remains constant
- D. Network I/O never spikes
- E. Overall cluster CPU utilization is around 25%

**Suggested Answer: D**

Community vote distribution

D (53%)

E (47%)

 **BrianNguyen95** Highly Voted 1 year, 10 months ago

Option E: In a Spark cluster, the driver node is responsible for managing the execution of the Spark application, including scheduling tasks, managing the execution plan, and interacting with the cluster manager. If the overall cluster CPU utilization is low (e.g., around 25%), it may indicate that the driver node is not utilizing the available resources effectively and might be a bottleneck.

upvoted 19 times

 **fe3b2fc** 10 months, 1 week ago


A bottleneck occurs when resources are over utilized not underutilized, so that explanation doesn't make too much sense. CPU utilization would be at 100% and you wouldn't see spike in I/O if the driver was the issue. Conversely if the I/O was spiked and CPU utilization was at 25% , then network could be the issue. D is the only logical answer in this case.

upvoted 3 times

 **benni\_ale** 8 months ago

i like this more

upvoted 2 times

 **guillesd** 1 year, 4 months ago

Overall CPU utilization can be misleading. The 25% utilization could be caused by the workload not requiring more than that rather than everything being executed in the driver node.

upvoted 2 times

 **sammy2025** Most Recent 1 month ago

**Selected Answer: E**

If the driver is doing too much of the work (rather than distributing tasks to executors), cluster-wide CPU usage will appear low, which is a red flag for driver bottlenecks in a distributed Spark environment.


upvoted 1 times

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

Surely E

upvoted 1 times

 **JoG1221** 2 months, 1 week ago

**Selected Answer: A**

Option E is valid and insightful, Option A is more targeted when you're specifically trying to detect a bottleneck on the driver.

upvoted 1 times

 **Tedet** 3 months, 4 weeks ago

**Selected Answer: A**

When you see the "Five Minute Load Average" remain consistent or flat, it could indicate that the driver is under heavy load and is struggling to keep up with the workload. In the case of a Spark cluster, if the driver is handling too much work, it can become a bottleneck and prevent the overall job from progressing efficiently.

upvoted 2 times

🗳️ 👤 **srinivasa** 6 months ago

Selected Answer: A

Consistent/Flat Five Minute Load Average: If the load average on the driver node remains consistent and does not fluctuate, it suggests that the driver is under constant, significant load. This could be a sign that the driver is performing a lot of work, potentially leading to a bottleneck.

upvoted 3 times

🗳️ 👤 **AlejandroU** 6 months, 2 weeks ago

Selected Answer: E

Answer E. A low CPU usage could indicate that the driver isn't working as efficiently as expected, which can lead to underutilization of the cluster and slower processing times.

upvoted 2 times

🗳️ 👤 **JB90** 7 months ago

Selected Answer: E

Only when the driver does all or most the work will the overall cluster CPU util be this low since the driver cpu is 25% of the overall cluster CPU amount

upvoted 1 times

🗳️ 👤 **nedlo** 8 months, 1 week ago

Selected Answer: E

bottleneck means data skew means one of the nodes is doing majority of work while other is idle, so E is correct

upvoted 2 times

🗳️ 👤 **m79590530** 8 months, 2 weeks ago

Selected Answer: E

D also means that Driver never send big data chunks to the Worker nodes but as it is not mentioned to be 0 then it has a constant flow of data going in & out between the Driver node and the Worker nodes. Therefore it is not a measure of Driver bottleneck. However Answer E means one of the 4 cluster nodes is always working at 100% which can not be other than the Driver node as it is always working and coordinating work across Executors.

upvoted 1 times

🗳️ 👤 **fe3b2fc** 10 months, 1 week ago

Selected Answer: D

Executors talk between each other and between nodes, if the code/driver is working as intended you would see a spike in I/O while transferring data. If the code/driver was the issue you would see a spike in CPU usage and little network traffic between nodes. The correct answer is D.

upvoted 2 times

🗳️ 👤 **lophonos** 1 year ago

Selected Answer: E

E is correct

upvoted 1 times

🗳️ 👤 **guillesd** 1 year, 4 months ago

Selected Answer: D

If there's no IO between driver and executor nodes then the executor nodes are not working

upvoted 1 times

🗳️ 👤 **Patito** 1 year, 6 months ago

Selected Answer: D

D seems to be right

upvoted 2 times

🗳️ 👤 **rok21** 1 year, 6 months ago

Selected Answer: E

E is correct

upvoted 1 times

🗳️ 👤 **azurelearn2020** 1 year, 6 months ago

Selected Answer: E



25% indicates Cluster CPU under-utilized

upvoted 2 times

🗳️ 👤 **Def21** 1 year, 5 months ago

Not correct. 25% could (in theory) mean driver is using 100% CPU

upvoted 1 times

  **sturcu** 1 year, 8 months ago

**Selected Answer: E**

If the overall cluster CPU utilization is around 25%, it means that only one out of the four nodes (driver + 3 executors) is using its full CPU capacity, while the other three nodes are idle or underutilized

upvoted 3 times

Where in the Spark UI can one diagnose a performance problem induced by not leveraging predicate push-down?

- A. In the Executor's log file, by grepping for "predicate push-down"
- B. In the Stage's Detail screen, in the Completed Stages table, by noting the size of data read from the Input column
- C. In the Storage Detail screen, by noting which RDDs are not stored on disk
- D. In the Delta Lake transaction log, by noting the column statistics
- E. In the Query Detail screen, by interpreting the Physical Plan

**Suggested Answer: E**

Community vote distribution

E (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

🗳️ 👤 **Tedet** 3 months, 4 weeks ago

**Selected Answer: E**

Predicate push-down is an optimization where conditions (such as filters) are pushed as close to the data source as possible (often to the database or file system level), reducing the amount of data read and processed. If predicate push-down isn't being leveraged, it can result in reading unnecessary data, leading to performance degradation.

Execute a query --> Click View and go to Spark UI --> Navigate to SQL/DataFrame tab in SparkUI --> Click on any stage --> Navigate to details to find Physical Plan

upvoted 1 times

🗳️ 👤 **shaswat1404** 4 months, 3 weeks ago

**Selected Answer: B**

when predicated pushdown is working properly, the amount of data read should be much lower because the data source is able to filter out the rows at read time based on the query predicates. if predicate pushdown is not leveraged, stages might read a much larger volume of data than necessary, which can be observed in the input column in the stage detail screen

therefore B is the correct option

not A : executor logs might contain some information, but they are not the most direct way to assess predicate push-down performance

not C : used to check RDD caching and persistence, not predicate push-down

not D : it holds meta data and statistics but is not viewed via the spark UI for diagnosing query performance

not E : while physical plan in the query detail screen might filter push-down, interpreting it requires more expertise, and the metric on the input data size(option B) is more straight forward indicator.

upvoted 1 times

🗳️ 👤 **benni\_ale** 7 months, 4 weeks ago

**Selected Answer: E**

E

upvoted 1 times

🗳️ 👤 **dd1192d** 8 months, 3 weeks ago

**Selected Answer: E**

E is correct : <https://docs.datastax.com/en/dse/6.9/spark/predicate-push-down.html>

upvoted 2 times

🗳️ 👤 **P1314** 1 year, 4 months ago

**Selected Answer: E**

Query plan. Correct is E

upvoted 1 times

Review the following error traceback:

```

AnalysisException                                Traceback (most recent call last)
<command-3293767849433948> in <module>
----> 1 display(df.select(3*"heartrate"))

/databricks/spark/python/pyspark/sql/dataframe.py in select(self, *cols)
   1690         [Row(name='Alice', age=12), Row(name='Bob', age=15)]
   1691         """
-> 1692         jdf = self._jdf.select(self._jcols(*cols))
   1693         return DataFrame(jdf, self.sql_ctx)
   1694

/databricks/spark/python/lib/py4j-0.10.9-src.zip/py4j/java_gateway.py in __call__(self, *args)
   1302
   1303         answer = self.gateway_client.send_command(command)
-> 1304         return_value = get_return_value(
   1305             answer, self.gateway_client, self.target_id, self.name)
   1306

/databricks/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
   121         # Hide where the exception came from that shows a non-Pythonic
   122         # JVM exception message.
--> 123         raise converted from None
   124     else:
   125         raise

AnalysisException: cannot resolve ''heartrateheartrateheartrate'' given input columns:
[spark_catalog.database.table.device_id, spark_catalog.database.table.heartrate,
spark_catalog.database.table.mrn, spark_catalog.database.table.time];
'Project ['heartrateheartrateheartrate]
+- SubqueryAlias spark_catalog.database.table
   +- Relation[device_id#75L,heartrate#76,mrn#77L,time#78] parquet

```

Which statement describes the error being raised?


- A. The code executed was PySpark but was executed in a Scala notebook.
- B. There is no column in the table named heartrateheartrateheartrate
- C. There is a type error because a column object cannot be multiplied.
- D. There is a type error because a DataFrame object cannot be multiplied.
- E. There is a syntax error because the heartrate column is not correctly identified as a column.

**Suggested Answer: E**

Community vote distribution


B (74%)

E (26%)

 **CertPeople** Highly Voted 1 year, 3 months ago


**Selected Answer: B**

It's B, there is no column with that name  
upvoted 8 times

 **rok21** Highly Voted 1 year ago

**Selected Answer: E**

E is correct  
upvoted 5 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: B**

Surely B  
upvoted 1 times

 **Stalker200** 2 months, 1 week ago

**Selected Answer: B**



B is correct.

I created a simple df and ran this code : `display(df.select(3*"heartrate"))` and I got this error:

AnalysisException: [UNRESOLVED\_COLUMN.WITH\_SUGGESTION] A column or function parameter with name `heartrateheartrateheartrate` cannot be resolved. Did you mean one of the following? [`heartrate`, `device\_id`, `time`, `mrn`].;

'Project [heartrateheartrateheartrate]

+ LogicalRDD [device\_id#2L, heartrate#3L, mrn#4L, time#5], false

upvoted 1 times

  **examtopicsms99** 2 months, 2 weeks ago

**Selected Answer: C**

```
from pyspark.sql import SparkSession
```

```
from pyspark.sql.functions import col
```

```
# Initialize Spark session
```

```
spark = SparkSession.builder.appName("ErrorReproduction").getOrCreate()
```

```
# Create a sample DataFrame with similar structure
```

```
data = [
```

```
(1, 72, 12345, "2023-01-01 10:00:00"),
```

```
(2, 68, 67890, "2023-01-01 10:01:00"),
```

```
(3, 75, 54321, "2023-01-01 10:02:00")
```

```
]
```

```
columns = ["device_id", "heartrate", "mrn", "time"]
```

```
df = spark.createDataFrame(data, columns)
```

```
# This will produce the AnalysisException
```


```
# The error occurs because you're trying to multiply a string "heartrate" by 3 literally
```

```
# instead of referencing the column and multiplying its values
```

```
display(df.select(3*"heartrate"))
```

```
# display(df.select(3*col("heartrate")))
```

upvoted 1 times

  **guillesd** 10 months, 3 weeks ago

**Selected Answer: B**

It's B. Regarding E, a syntax error would mean that the query is not valid due to a wrongfully written SQL statement. However, this is not the case. The column just does not exist.

upvoted 2 times

  **Jay\_98\_11** 11 months, 3 weeks ago

**Selected Answer: B**

<https://sparkbyexamples.com/spark/spark-cannot-resolve-given-input-columns/>

upvoted 1 times

  **Gulenur\_GS** 1 year ago

the answer is E, because

```
df.select(3*df['heartrate']).show()
```

 perfectly returns

upvoted 2 times

  **chokthewa** 11 months, 2 weeks ago

3\*"heartrate" is triple of string "heartrate" ,isn't value of heartrate multiplied by 3.

upvoted 1 times

  **Gulenur** 1 year ago

Answer is E

```
df.select(3*df['heartrate'])
```

 returns perfect result without error



upvoted 2 times

  **npc0001** 1 year, 1 month ago

**Selected Answer: B**



Answer B

upvoted 2 times

  **Dileepvikram** 1 year, 1 month ago

Answer is B

upvoted 2 times

  **sturcu** 1 year, 2 months ago

**Selected Answer: B**

No such column found

upvoted 2 times

Which distribution does Databricks support for installing custom Python code packages?

- A. sbt
- B. CRAN. npm
- D. Wheels
- E. jars

**Suggested Answer:** D

*Community vote distribution*

D (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer:** D

suretly D

upvoted 1 times

🗳️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer:** D

I think D is correct

upvoted 1 times

🗳️ 👤 **hal2401me** 1 year, 4 months ago

**Selected Answer:** D

<https://learn.microsoft.com/en-us/azure/databricks/workflows/jobs/how-to/use-python-wheels-in-workflows>

upvoted 4 times

🗳️ 👤 **sodere** 1 year, 6 months ago

**Selected Answer:** D

<https://learn.microsoft.com/en-us/azure/databricks/workflows/jobs/how-to/use-python-wheels-in-workflows>

upvoted 1 times

🗳️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer:** D

Wheels should be ok

upvoted 2 times


Which Python variable contains a list of directories to be searched when trying to locate required modules?

- A. `importlib.resource_path`
- B. `sys.path`
- C. `os.path`
- D. `pypi.path`
- E. `pylib.source`

**Suggested Answer:** B

Community vote distribution

B (100%)

 **alexvno** Highly Voted 1 year, 6 months ago


**Selected Answer:** B

`sys.path` is a built-in variable within the `sys` module. It contains a list of directories that the interpreter will search in for the required module  
upvoted 7 times

 **KadELbied** Most Recent 1 month, 3 weeks ago


**Selected Answer:** B

suretly B  
upvoted 1 times

 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer:** B

`sys.path` is a list in Python that contains the directories the interpreter searches for modules when importing them. It is initialized with the default paths when Python starts and can be modified during runtime if needed.  
upvoted 1 times

 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer:** B

`sys.path` is a built-in variable within the `sys` module. It contains a list of directories that the interpreter will search in for the required module.  
upvoted 1 times

Incorporating unit tests into a PySpark application requires upfront attention to the design of your jobs, or a potentially significant refactoring of existing code.

Which statement describes a main benefit that offset this additional effort?

- A. Improves the quality of your data
- B. Validates a complete use case of your application
- C. Troubleshooting is easier since all steps are isolated and tested individually
- D. Yields faster deployment and execution times
- E. Ensures that all steps interact correctly to achieve the desired end result

**Suggested Answer:** C

*Community vote distribution*

C (100%)

  **alexvno**  1 year, 6 months ago

**Selected Answer:** C

Unit tests are small, isolated tests that are used to check specific parts of the code, such as functions or classes  
upvoted 5 times

  **KadELbied**  1 month, 3 weeks ago

**Selected Answer:** C

Surety C  
upvoted 1 times

  **nedlo** 8 months, 1 week ago

**Selected Answer:** C

D is integration tests (how they relate to each other how connect), E is E2E test, C is "testing individually" which is only one fittign definition of unittest  
upvoted 2 times

  **nedlo** 8 months, 1 week ago

i mean E is integration test B is E2E test  
upvoted 1 times

  **jmjm21** 1 year ago

**Selected Answer:** C

Answer is C.  
upvoted 1 times

Which statement describes integration testing?

- A. Validates interactions between subsystems of your application
- B. Requires an automated testing framework
- C. Requires manual intervention
- D. Validates an application use case
- E. Validates behavior of individual elements of your application

**Suggested Answer: A**

*Community vote distribution*

A (100%)

🗨️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

🗨️ 👤 **robodog** 10 months, 2 weeks ago

**Selected Answer: A**

Answer is A

upvoted 2 times

🗨️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: A**

Integration testing is a type of software testing where components of the software are gradually integrated and then tested as a unified group

upvoted 4 times

Which REST API call can be used to review the notebooks configured to run as tasks in a multi-task job?

- A. /jobs/runs/list
- B. /jobs/runs/get-output
- C. /jobs/runs/get
- D. /jobs/get
- E. /jobs/list

**Suggested Answer:** D

Community vote distribution

D (83%)

B (17%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer:** D

suretly D

upvoted 2 times

🗳️ 👤 **JoG1221** 2 months, 1 week ago

**Selected Answer:** E

Provides a list of all jobs in the workspace along with their configurations

upvoted 1 times

🗳️ 👤 **AlejandroU** 6 months, 2 weeks ago

**Selected Answer:** E

The correct answer is E. /jobs/list, not C. /jobs/runs/get. Here's why:

/jobs/list: Provides a list of all jobs in the workspace along with their configurations, including task details like the notebooks assigned to each task.

This makes it the best choice for reviewing notebooks configured as tasks in a multi-task job.

/jobs/get: Can also be used if the goal is to review the tasks (and notebooks) of a specific job. However, the question does not limit the scope to a single job.

upvoted 1 times

🗳️ 👤 **imatheushenrique** 1 year ago

multi-task: /jobs/get

single-task: /jobs/runs/get

upvoted 2 times

🗳️ 👤 **hal2401me** 1 year, 4 months ago

**Selected Answer:** D

<https://docs.databricks.com/api/workspace/jobs/get>

responses/settings/tasks/notebook\_task/notebook\_path

upvoted 1 times

🗳️ 👤 **divingbell17** 1 year, 6 months ago

**Selected Answer:** D

The question asks for notebooks configured for a job, not a instance of a job run. D is correct.

upvoted 4 times

🗳️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer:** D

Get

Multi-task format jobs return an array of task data structures containing task settings.

upvoted 1 times

🗳️ 👤 **hamzaKhribi** 1 year, 6 months ago

**Selected Answer:** D

/jobs/get response under task array shows all the desired notebooks

upvoted 1 times

  **arye777** 1 year, 7 months ago

**Selected Answer: B**

should be B

upvoted 1 times



A Databricks job has been configured with 3 tasks, each of which is a Databricks notebook. Task A does not depend on other tasks. Tasks B and C run in parallel, with each having a serial dependency on task A.

If tasks A and B complete successfully but task C fails during a scheduled run, which statement describes the resulting state?

- A. All logic expressed in the notebook associated with tasks A and B will have been successfully completed; some operations in task C may have completed successfully.
- B. All logic expressed in the notebook associated with tasks A and B will have been successfully completed; any changes made in task C will be rolled back due to task failure.
- C. All logic expressed in the notebook associated with task A will have been successfully completed; tasks B and C will not commit any changes because of stage failure.
- D. Because all tasks are managed as a dependency graph, no changes will be committed to the Lakehouse until all tasks have successfully been completed.
- E. Unless all tasks complete successfully, no changes will be committed to the Lakehouse; because task C failed, all commits will be rolled back automatically.

**Suggested Answer: A**


Community vote distribution

A (100%)

 **IT3008** Highly Voted 1 year, 9 months ago

Should be 'A' only, as ACID compliance is applicable at operation level. For example if task C is having 3 target delta table writes (in independent Notebook cells) then it could have after 1 write the task fails during 2nd write. In that case 1st write will still be persisted. The ACID compliance will be applicable for only the 2nd write.

upvoted 9 times

 **alexvno** Highly Voted 1 year, 6 months ago

**Selected Answer: A**

A - for sure this is NOT ACID operations


upvoted 5 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

Surely A


upvoted 1 times

 **dd1192d** 8 months, 3 weeks ago

**Selected Answer: A**

<https://community.databricks.com/t5/data-engineering/does-cancelling-a-job-run-rollback-any-actions-performed-by/td-p/8135>

upvoted 2 times

 **tkg13** 1 year, 10 months ago

Correct answer should be B as Databricks is ACID compliant

upvoted 2 times

 **arekm** 5 months, 4 weeks ago

A single SQL command is ACID compliant, not a whole notebook.

upvoted 1 times

 **eli91** 1 year, 9 months ago

What if an operation of C is to delete a file, will the file be created after a roll back?

upvoted 2 times

A Delta Lake table was created with the below query:

```
CREATE TABLE prod.sales_by_stor
USING DELTA
LOCATION "/mnt/prod/sales_by_store"
```

Realizing that the original query had a typographical error, the below code was executed:

```
ALTER TABLE prod.sales_by_stor RENAME TO prod.sales_by_store
```

Which result will occur after running the second command?

- A. The table reference in the metastore is updated and no data is changed.
- B. The table name change is recorded in the Delta transaction log.
- C. All related files and metadata are dropped and recreated in a single ACID transaction.
- D. The table reference in the metastore is updated and all data files are moved.
- E. A new Delta transaction log is created for the renamed table.

**Suggested Answer: A**

Community vote distribution


A (100%)

  **hal2401me** Highly Voted 9 months, 3 weeks ago

**Selected Answer: A**

did a test. No data is changed. dir & filename not changed. the rename is not recorded in transition log neither.



upvoted 8 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times

  **Tamele001** 9 months, 3 weeks ago

B is the correct answer. When you alter a table name in Delta Lake, the change is logged in the transaction log that Delta Lake uses to maintain a versioned history of all changes to the table. This is how Delta Lake maintains ACID properties and ensures a consistent view of the data. The transaction log is key to supporting features like time travel, auditing, and rollbacks in Delta Lake. The metadata and the actual data remain intact, and the reference to the table in the metastore is updated to reflect the new name.

upvoted 2 times

  **arekm** 5 months, 4 weeks ago

Transaction log captures operations on the data (including adding columns, renaming them). This operation is a change of the name of the external table - just a change in the metastore.

upvoted 1 times

  **adenis** 11 months ago

**Selected Answer: A**

A is Correct

upvoted 4 times

The data engineering team maintains a table of aggregate statistics through batch nightly updates. This includes total sales for the previous day alongside totals and averages for a variety of time periods including the 7 previous days, year-to-date, and quarter-to-date. This table is named `store_sales_summary` and the schema is as follows:

```
store_id INT, total_sales_qtd FLOAT, avg_daily_sales_qtd FLOAT, total_sales_ytd FLOAT,
avg_daily_sales_ytd FLOAT, previous_day_sales FLOAT, total_sales_7d FLOAT, avg_daily_sales_7d
FLOAT, updated TIMESTAMP
```

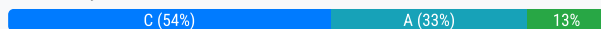
The table `daily_store_sales` contains all the information needed to update `store_sales_summary`. The schema for this table is: `store_id INT`, `sales_date DATE`, `total_sales FLOAT`

If `daily_store_sales` is implemented as a Type 1 table and the `total_sales` column might be adjusted after manual data auditing, which approach is the safest to generate accurate reports in the `store_sales_summary` table?

- A. Implement the appropriate aggregate logic as a batch read against the `daily_store_sales` table and overwrite the `store_sales_summary` table with each Update.
- B. Implement the appropriate aggregate logic as a batch read against the `daily_store_sales` table and append new rows nightly to the `store_sales_summary` table.
- C. Implement the appropriate aggregate logic as a batch read against the `daily_store_sales` table and use upsert logic to update results in the `store_sales_summary` table.
- D. Implement the appropriate aggregate logic as a Structured Streaming read against the `daily_store_sales` table and use upsert logic to update results in the `store_sales_summary` table.
- E. Use Structured Streaming to subscribe to the change data feed for `daily_store_sales` and apply changes to the aggregates in the `store_sales_summary` table with each update.

**Suggested Answer: C -**

Community vote distribution



**hammer\_1234\_h** Highly Voted 1 year, 9 months ago

The answer should be A. it is the safest to generate accurate report  
upvoted 12 times

**Def21** 1 year, 5 months ago

This is confusing: "overwrite the `store_sales_summary` table with each Update." sounds like it is only doing updates, not inserting new possible stories.  
upvoted 4 times

**alexvno** 1 year, 6 months ago

Incorrect BATCH processing and OVERWRITE will give partial results  
upvoted 2 times

**djohn\_prasad** 4 months ago

But what if each batch did not have historical data but only updates? Then doing an overwrite will risk removing historical data in the target table so Option C is much preferable  
upvoted 2 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: C**

suretly C  
upvoted 1 times

**ultimomassimo** 3 months ago

people that claim type 1 scd is not maintained by upsert need to make some more reading before posting anything here...  
upvoted 1 times

**Jamuro** 3 months, 3 weeks ago

**Selected Answer: C**

It makes no sense to recalculate all aggregations for all historical data and then overwrite. It would lead to a non scalable solution because all data should be recalculated to not "delete" rows in target table. Option C lets us filter by date ranges or even partitions to merge only target periods and get the same results in a much more performant, safer and cheaper way.

upvoted 1 times

🗨️ **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: C**

The answer is C. Option A is correct in ensuring accuracy, as it recalculates the entire store\_sales\_summary table based on the full historical data in daily\_store\_sales. However, it is computationally expensive and may not scale well.

Option C (upsert logic) could be a better choice in most real-world scenarios, as it focuses only on the records that have changed, reducing computational costs and minimizing disruption for downstream systems.

upvoted 1 times

🗨️ **arekm** 5 months, 4 weeks ago

How do you know which records have changed? I think A is the safest answer.

upvoted 1 times

🗨️ **djohn\_prasad** 4 months ago

Option C tells us based on upsert logic where matching records (matching key assumed as storeid and sales date) are checked for differences in sales values and the value from batch is taken over that of the target table. C is computational much easier then

upvoted 1 times

🗨️ **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: A**

A it is

SCD Type 1, so clearly Append and Upsert logic should not be used.

upvoted 3 times

🗨️ **ultimomassimo** 3 months ago

you dont know what is scd type 1 , do you? you dont understand the difference between overwriting a column value and overwriting the whole table, do you?

upvoted 1 times

🗨️ **benni\_ale** 6 months, 3 weeks ago

**Selected Answer: A**

A as the table does not require history

upvoted 1 times

🗨️ **shaojunni** 8 months, 3 weeks ago

**Selected Answer: A**

daily\_store\_sales is type1 table, no history is maintained. You have to treat every record as new record and do aggregation for every store. Overwrite is much efficient than upsert.

upvoted 1 times

🗨️ **Ati1362** 1 year ago

**Selected Answer: C**

I will go with c. upsert

upvoted 3 times

🗨️ **MDWPartners** 1 year, 1 month ago

**Selected Answer: C**

A is not correct because the table is daily. If you overwrite you delete all history. You need to insert/update to keep history.

upvoted 4 times

🗨️ **fe3b2fc** 10 months, 2 weeks ago

Incorrect. The daily store sales table contains all of the history needed to update the table. The summary table holds no historical records. Seeing as this is a nightly job, any manual changes made to daily store sales will be captured. A is the correct answer.

upvoted 2 times

🗨️ **ThoBustos** 1 year, 2 months ago

**Selected Answer: A**

Not sure if that's right but I would go for A. What do you think?

Type1: Data is overwritten

Type 2: History is maintained, new data is inserted as new rows

Type 3: Stores two versions per record: a previous and a current value

A. batch + overwrite -> Match Type 1 requirements. YES

B: batch + append new rows -> Would be for type 2. NO

C. Batch + Upsert -> Data is not being overwritten (which is required for Type 1). NO

D. ReadStream + Upsert -> Data is not being overwritten (which is required for Type 1). NO

E. Change Data Feed to update -> Problem is manual edits + not overwriting (required for type 1). No

I have doubts around "which approach is the safest". Maybe because due to some manual changes it is hard to track changes or do upsert, so to make sure that the stats are right

overwriting is safer.

upvoted 4 times

  **vikram12apr** 1 year, 3 months ago

**Selected Answer: C**


Not A because overwriting will only provide a daily based data not the history of it.

Not B because it will not fix the issue of incorrect sales amount

As these data are fit for natch processing so neither D or E.

C will only upsert the changes while making sure we are updating the records based on sales\_date & store\_id

upvoted 2 times

  **Rinscy** 1 year, 5 months ago

E definitely because it say that the total\_sales column may be change by manual auditing so not via a job, so streaming with CDF is the only option here !

upvoted 1 times

  **Somesh512** 1 year, 5 months ago

**Selected Answer: A**

I would go with Option A.

Because it has manual auditing hence values can change. Uses type 1 hence replace original data


upvoted 3 times

  **spaceexplorer** 1 year, 5 months ago

**Selected Answer: E**

It should be E, as structure streaming has built-in fault-tolerance feature.

upvoted 1 times

  **Rinscy** 1 year, 5 months ago

It said type 1 so A is the correct answer !

upvoted 2 times

  **divingbell17** 1 year, 6 months ago

The question is unclear whether the aggregated table needs to support a rolling history. Note the aggregated table does not have a date column to distinguish which date the summary is generated for so one could assume the table is maintained only for the current snapshot.

Assuming the above - A would be the safest option as all stores and aggregates would need to be refreshed nightly

upvoted 2 times

A member of the data engineering team has submitted a short notebook that they wish to schedule as part of a larger data pipeline. Assume that the commands provided below produce the logically correct results when run as presented.

**Cmd 1**

```
rawDF = spark.table("raw_data")
```

**Cmd 2**

```
rawDF.printSchema()
```

**Cmd 3**

```
flattenedDF = rawDF.select("...", "values.*")
```

**Cmd 4**

```
finalDF = flattenedDF.drop("values")
```

**Cmd 5**

```
finalDF.explain()
```

**Cmd 6**

```
display(finalDF)
```

**Cmd 7**

```
finalDF.write.mode("append").saveAsTable("flat_data")
```

Which command should be removed from the notebook before scheduling it as a job?

- A. Cmd 2
- B. Cmd 3
- C. Cmd 4
- D. Cmd 5
- E. Cmd 6

**Suggested Answer:** E

Community vote distribution

E (93%)

7%


 **petrv** Highly Voted 1 year, 7 months ago

**Selected Answer: E**

When scheduling a Databricks notebook as a job, it's generally recommended to remove or modify commands that involve displaying output, such as using the `display()` function. Displaying data using `display()` is an interactive feature designed for exploration and visualization within the notebook interface and may not work well in a production job context.

The `finalDF.explain()` command, which provides the execution plan of the DataFrame transformations and actions, is often useful for debugging and optimizing queries. While it doesn't display interactive visualizations like `display()`, it can still be informative for understanding how Spark is executing the operations on your DataFrame.

upvoted 10 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

Surely E

upvoted 1 times

🗨️ 👤 **Carkeys** 8 months, 1 week ago

**Selected Answer: D**

Cmd 5 (`finalDF.explain()`) is used for debugging and understanding the logical and physical plans of a DataFrame. It provides insights into how Spark plans to execute the query but does not produce output that is necessary for the scheduled job. Including this command in a scheduled job is unnecessary and could clutter the job logs without adding value to the final output.

upvoted 1 times

🗨️ 👤 **arekm** 5 months, 4 weeks ago

`display()` is more costly operation than `finalDF.explain()`. The DataFrame might contain millions of rows that you would be trying to print out each time the notebook is run.

upvoted 1 times

🗨️ 👤 **benni\_ale** 8 months, 2 weeks ago

**Selected Answer: E**

if i was multiple solutions than i would have gone for `.explain` method and print schema as well as they do not contribute in any sort of ETL operation but as a rule of thumb `display` should always be omitted first so -> E

upvoted 1 times

🗨️ 👤 **71dfab9** 10 months, 2 weeks ago

**Selected Answer: E**

I agree with petrV and KhoaLe, but I will add that not displaying the finalDF would be wise as it could display and log PII data and that to me is why I choose E. Like hal2401 said, commands 2, 5 & 6 can be removed as they don't manipulate the data.

upvoted 1 times

🗨️ 👤 **hal2401me** 1 year, 4 months ago

**Selected Answer: E**

perhaps it's a multi-choice question in exam. I'll select E and D. if single choice then E.

upvoted 1 times

🗨️ 👤 **KhoaLe** 1 year, 4 months ago

**Selected Answer: E**

Looking through at all steps, Cmd 2,5,6 can be eliminated without impacting to the whole process.

However, in terms of duration cost, Cmd 2 and 5 does not impact much as they only show the current results of logical query plan. In contrast, `display()` in Cmd6 is actually a transformation, which will take much time to run.

upvoted 2 times

🗨️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: E**

No `display()`

upvoted 3 times

🗨️ 👤 **60ties** 1 year, 7 months ago

**Selected Answer: D**

No actions on production scripts. D is best

upvoted 1 times

🗨️ 👤 **ofed** 1 year, 7 months ago

in order to display a dataframe you also need to calculate it. So `display` also acts as an action.

upvoted 1 times

🗨️ 👤 **Karen1232123** 1 year, 7 months ago

Why not D?

upvoted 2 times

The business reporting team requires that data for their dashboards be updated every hour. The total processing time for the pipeline that extracts transforms, and loads the data for their pipeline runs in 10 minutes.

Assuming normal operating conditions, which configuration will meet their service-level agreement requirements with the lowest cost?



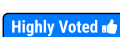
- A. Manually trigger a job anytime the business reporting team refreshes their dashboards
- B. Schedule a job to execute the pipeline once an hour on a new job cluster
- C. Schedule a Structured Streaming job with a trigger interval of 60 minutes
- D. Schedule a job to execute the pipeline once an hour on a dedicated interactive cluster
- E. Configure a job that executes every time new data lands in a given directory

**Suggested Answer: C**

Community vote distribution

B (83%)

C (17%)

  **divingbell17**  1 year, 6 months ago

**Selected Answer: B**

B is correct I think.

With option C, the cluster remains on 24/7 with trigger = 60 mins which is more costly

If there is an option with structure streaming with trigger = availablenow, and job scheduled per hour, that would be even more efficient.




<https://www.databricks.com/blog/2017/05/22/running-streaming-jobs-day-10x-cost-savings.html>

upvoted 10 times

  **arekm** 5 months, 4 weeks ago

Always a job cluster.

upvoted 1 times

  **KadELbied**  1 month, 3 weeks ago

**Selected Answer: B**

Surely B

upvoted 1 times

  **robodog** 10 months, 1 week ago

**Selected Answer: C**

C. The lowest cost is obtained by using job cluster

upvoted 1 times

  **robodog** 10 months, 1 week ago

B answer i mean

upvoted 2 times

  **Curious76** 1 year, 4 months ago

**Selected Answer: C**

Databricks recommends using Structured Streaming with trigger AvailableNow for incremental workloads that do not have low latency requirements.



upvoted 2 times

  **spaceexplorer** 1 year, 5 months ago

**Selected Answer: B**

B is correct

upvoted 4 times

  **alexvno** 1 year, 6 months ago

**Selected Answer: B**

B : Job cluster is cheap , hourly = 60 minutes

upvoted 4 times



🗨️ 👤 **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: B**

Scheduling a job to execute the pipeline on an hourly basis aligns with the requirement for data to be updated every hour. Using a job cluster (which is brought up for the job and torn down upon completion) rather than a dedicated interactive cluster will usually be more cost-effective. This is because you are only paying for the compute resources when the job is running, which is 10 minutes out of every hour, rather than paying for an interactive cluster that would be up and running (and incurring costs) continuously.

upvoted 2 times

🗨️ 👤 **ofed** 1 year, 7 months ago

It's either B or D. I think B, because we want the lowest cost.

upvoted 1 times

A Databricks SQL dashboard has been configured to monitor the total number of records present in a collection of Delta Lake tables using the following query pattern:

```
SELECT COUNT (*) FROM table -
```




Which of the following describes how results are generated each time the dashboard is updated?

- A. The total count of rows is calculated by scanning all data files
- B. The total count of rows will be returned from cached results unless REFRESH is run
- C. The total count of records is calculated from the Delta transaction logs
- D. The total count of records is calculated from the parquet file metadata
- E. The total count of records is calculated from the Hive metastore

**Suggested Answer: A**

Community vote distribution

C (100%)

  **aragorn\_brego**  1 year, 7 months ago

**Selected Answer: C**

Delta Lake maintains a transaction log that records details about every change made to a table. When you execute a count operation on a Delta table, Delta Lake can use the information in the transaction log to calculate the total number of records without having to scan all the data files. This is because the transaction log includes information about the number of records in each file, allowing for an efficient aggregation of these counts to get the total number of records in the table.

upvoted 6 times

  **Syd**  1 year, 7 months ago

Answer C

[https://delta.io/blog/2023-04-19-faster-aggregations-](https://delta.io/blog/2023-04-19-faster-aggregations-metadata/#:~:text=You%20can%20get%20the%20number,a%20given%20Delta%20table%20version.)

[metadata/#:~:text=You%20can%20get%20the%20number,a%20given%20Delta%20table%20version.](https://delta.io/blog/2023-04-19-faster-aggregations-metadata/#:~:text=You%20can%20get%20the%20number,a%20given%20Delta%20table%20version.)

upvoted 5 times

  **c315d10**  1 month ago

**Selected Answer: A**

Metadata could be outdated



upvoted 1 times

  **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

Surely C

upvoted 1 times

  **AlejandroU** 6 months, 2 weeks ago

**Selected Answer: D**

Answer D. Parquet Metadata Usage: Delta Lake does utilize Parquet file metadata for COUNT(\*) operations. Parquet files store metadata, including row counts. Delta efficiently reads this metadata to get the total count without scanning the actual data within the files. This is a key optimization for performance.

Why not always scan: Scanning all data files for every COUNT(\*) would be extremely inefficient, especially for large tables. This defeats the purpose of using a columnar storage format like Parquet and the optimizations built into Delta Lake and Spark.

The transaction log tracks changes to the table (adds, deletes, updates) but doesn't store pre-computed row counts. It's used for time travel, ACID properties, and other Delta features.

upvoted 2 times

🗄️ 👤 **arekm** 5 months, 4 weeks ago  
Definitely C - see link posted by Syd  
upvoted 1 times

🗄️ 👤 **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: C**

"stats": {"numRecords": 3, "minValues": {"x": 1}, "maxValues": {"x": 3}, "nullCount": {"x": 0}},  
numRecords - In Delta tx logs will give you the value  
upvoted 1 times

🗄️ 👤 **Ati1362** 1 year ago

**Selected Answer: C**

Delta transaction log  
upvoted 2 times

🗄️ 👤 **sodere** 1 year, 6 months ago

**Selected Answer: C**

Transaction log provides statistics about the delta table.  
upvoted 4 times

🗄️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: C**

C - transaction logs contains info about files rows count  
upvoted 3 times

🗄️ 👤 **Dileepvikram** 1 year, 7 months ago

The answer is C  
upvoted 2 times

🗄️ 👤 **PearApple** 1 year, 7 months ago

**Selected Answer: C**

The answer should be C  
upvoted 2 times

🗄️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: C**

total rows will be calculated from delta logs  
upvoted 3 times

A Delta Lake table was created with the below query:

```
CREATE TABLE prod.sales_by_store
AS (
  SELECT *
  FROM prod.sales a
  INNER JOIN prod.store b
  ON a.store_id = b.store_id
)
```

Consider the following query:

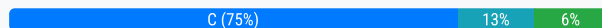
DROP TABLE prod.sales\_by\_store -

If this statement is executed by a workspace admin, which result will occur?

- A. Nothing will occur until a COMMIT command is executed.
- B. The table will be removed from the catalog but the data will remain in storage.
- C. The table will be removed from the catalog and the data will be deleted.
- D. An error will occur because Delta Lake prevents the deletion of production data.
- E. Data will be marked as deleted but still recoverable with Time Travel.

**Suggested Answer: D**

Community vote distribution



hal2401me Highly Voted 1 year, 3 months ago

Selected Answer: C

According to the exam courses answer is C, for a managed table dropped.

But, as after Nov'23, UNDROP is introduced and I have test it working with UC managed tables.

<https://docs.databricks.com/en/sql/language-manual/sql-ref-syntax-ddl-undrop-table.html>

However, I don't see any official doc says UNDROP related to 'time travel'.

So, be aware of the above info; in exam, watch the question carefully if it is updated.

upvoted 10 times

db22 9 months, 2 weeks ago

Agree that the answer is C - the question is misleading in saying it is Delta Lake table. However, it is managed table b/c there is no USING delta clause.

upvoted 1 times

KadELbied Most Recent 1 month, 3 weeks ago

Selected Answer: C

suretly C

upvoted 1 times

kishanu 2 months, 3 weeks ago

Selected Answer: C

UNDROP can be used, within a 7 day retention period .

upvoted 1 times

Er5 1 year, 2 months ago

C. is only correct statement. Though the table can be UNDROP in 7 days

<https://learn.microsoft.com/en-us/azure/databricks/sql/language-manual/sql-ref-syntax-ddl-undrop-table>

E. Time Travel can retrieve versioned records but not tables.

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

upvoted 4 times

🗳️ 👤 **Curious76** 1 year, 4 months ago

**Selected Answer: E**

I think E is better answer

upvoted 2 times

🗳️ 👤 **Sriramiyer92** 6 months, 2 weeks ago

Feel this is incorrect.

Why?

Undrop Syntax: `UNDROP TABLE { table_name | WITH ID table_id }`

So no question if Time travel

upvoted 1 times

🗳️ 👤 **Luv4data** 1 year, 6 months ago

E. Since the table is still recoverable from transaction logs.

upvoted 1 times

🗳️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: C**

C : AS SELECT - Managed table

Will remove table and data

upvoted 2 times

🗳️ 👤 **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: C**

In Delta Lake, when a DROP TABLE command is executed, it removes both the metadata entry for the table from the catalog and the data in storage associated with that table. Workspace administrators typically have the necessary permissions to drop tables, and unless there are additional protections or retention policies in place, the data is not recoverable through normal operations after the table is dropped.

upvoted 3 times

🗳️ 👤 **60ties** 1 year, 7 months ago

I meant C is correct, not D

upvoted 3 times

🗳️ 👤 **60ties** 1 year, 7 months ago

**Selected Answer: D**

D is most correct

upvoted 1 times

🗳️ 👤 **Dileepvikram** 1 year, 7 months ago

Answer is C as it is a managed table

upvoted 1 times

🗳️ 👤 **lokvamsi** 1 year, 8 months ago

**Selected Answer: C**

it is a managed table

upvoted 1 times

🗳️ 👤 **lokvamsi** 1 year, 8 months ago

**Selected Answer: A**

its a as it is managed table

upvoted 1 times

🗳️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: C**

it is a managed table. So both table def and data will be deleted

upvoted 1 times

🗳️ 👤 **jyothsna12496** 1 year, 8 months ago

**Selected Answer: C**

Drop will usually delete the table structure and data if its managed, hence c

upvoted 1 times

Two of the most common data locations on Databricks are the DBFS root storage and external object storage mounted with `dbutils.fs.mount()`.

Which of the following statements is correct?

- A. DBFS is a file system protocol that allows users to interact with files stored in object storage using syntax and guarantees similar to Unix file systems.
- B. By default, both the DBFS root and mounted data sources are only accessible to workspace administrators.
- C. The DBFS root is the most secure location to store data, because mounted storage volumes must have full public read and write permissions.
- D. Neither the DBFS root nor mounted storage can be accessed when using `%sh` in a Databricks notebook.
- E. The DBFS root stores files in ephemeral block volumes attached to the driver, while mounted directories will always persist saved data to external storage between sessions.

**Suggested Answer: E**

Community vote distribution

A (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

🗳️ 👤 **Curious76** 10 months, 1 week ago

**Selected Answer: A**

A is correct . For E, This statement is partially incorrect. The DBFS root does use ephemeral storage, but not block volumes. Data saved there is lost when the cluster terminates unless explicitly persisted elsewhere. Mounted storage, however, can persist data between sessions depending on the underlying storage service and configuration.

upvoted 4 times

🗳️ 👤 **sodere** 1 year ago

**Selected Answer: A**

DBFS is a layer on top of cloud storage providers.

upvoted 2 times

🗳️ 👤 **aragorn\_brego** 1 year, 1 month ago

**Selected Answer: A**

Databricks File System (DBFS) is a layer over a cloud object storage (like AWS S3, Azure Blob Storage, or GCP Cloud Storage) that allows users to interact with data as if they were using a traditional file system. It provides familiar file system semantics and is designed to be consistent with POSIX-like file system behavior, which includes commands and actions similar to those used in Unix and Linux file systems.

upvoted 3 times

🗳️ 👤 **Dileepvikram** 1 year, 1 month ago

Answer is A

upvoted 1 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

**Selected Answer: A**

it is not E.

The only one that would be plausible is A

upvoted 2 times

The following code has been migrated to a Databricks notebook from a legacy workload:

```
%sh
git clone https://github.com/foo/data_loader;
python ./data_loader/run.py;
mv ./output /dbfs/mnt/new_data
```

The code executes successfully and provides the logically correct results, however, it takes over 20 minutes to extract and load around 1 GB of data.

Which statement is a possible explanation for this behavior?

- A. %sh triggers a cluster restart to collect and install Git. Most of the latency is related to cluster startup time.
- B. Instead of cloning, the code should use %sh pip install so that the Python code can get executed in parallel across all nodes in a cluster.
- C. %sh does not distribute file moving operations; the final line of code should be updated to use %fs instead.
- D. Python will always execute slower than Scala on Databricks. The run.py script should be refactored to Scala.
- E. %sh executes shell code on the driver node. The code does not take advantage of the worker nodes or Databricks optimized Spark.

**Suggested Answer: C**

Community vote distribution

E (100%)

🗳️ 👤 **aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: E**

When using %sh in a Databricks notebook, the commands are executed in a shell environment on the driver node. This means that only the resources of the driver node are used, and the execution does not leverage the distributed computing capabilities of the worker nodes in the Spark cluster. This can result in slower performance, especially for data-intensive tasks, compared to an approach that distributes the workload across all nodes in the cluster using Spark.

upvoted 9 times

🗳️ 👤 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

Surely E

upvoted 1 times

🗳️ 👤 **robodog** 10 months, 1 week ago

**Selected Answer: E**

Option E correct

upvoted 1 times

🗳️ 👤 **Freyr** 1 year, 1 month ago

**Selected Answer: E**

Option E: Correct. The %sh magic command in Databricks runs shell commands on the driver node only. This means the operations within %sh do not leverage the distributed nature of the Databricks cluster. Consequently, the Git clone, Python script execution, and file move operations are all performed on a single node (the driver), which explains why it takes a long time to process and move 1 GB of data. This approach does not utilize the parallel processing capabilities of the worker nodes or the optimization features of Databricks Spark.

Option C: Incorrect. %sh does not inherently distribute any operations, but the issue here is broader than just file moving operations. Using %fs for file operations is a best practice, but it does not resolve the inefficiency of running all commands on the driver node.

upvoted 2 times

🗳️ 👤 **Dileepvikram** 1 year, 7 months ago

E is the answer as the command is ran in the driver node and other nodes in the cluster are not used

upvoted 2 times

🗳️ 👤 **sturcu** 1 year, 8 months ago





**Selected Answer: E**

%sh run Bash commands on the driver node of the cluster.



<https://www.databricks.com/blog/2020/08/31/introducing-the-databricks-web-terminal.html>

upvoted 3 times

  **sturcu** 1 year, 8 months ago

you can use mv with %sh, but the syntax is not correct , it is missing the destination operand

upvoted 1 times

  **sturcu** 1 year, 8 months ago

I just noticed there is a space between the paths, so syntax is correct

upvoted 2 times

The data science team has requested assistance in accelerating queries on free form text from user reviews. The data is currently stored in Parquet with the below schema:

item\_id INT, user\_id INT, review\_id INT, rating FLOAT, review STRING

The review column contains the full text of the review left by the user. Specifically, the data science team is looking to identify if any of 30 key words exist in this field.

A junior data engineer suggests converting this data to Delta Lake will improve query performance.

Which response to the junior data engineer's suggestion is correct?

- A. Delta Lake statistics are not optimized for free text fields with high cardinality.
- B. Text data cannot be stored with Delta Lake.
- C. ZORDER ON review will need to be run to see performance gains.
- D. The Delta log creates a term matrix for free text fields to support selective filtering.
- E. Delta Lake statistics are only collected on the first 4 columns in a table.

**Suggested Answer: D**

Community vote distribution



A (100%)

  **aragorn\_brego** Highly Voted 7 months, 1 week ago

**Selected Answer: A**

Delta Lake uses statistics and data skipping to improve query performance, but these optimizations are most effective for columns with low to medium cardinality (i.e., columns with a limited set of distinct values). Free-form text fields like the review column typically have high cardinality, meaning each value in the column (each review text) is unique or nearly unique. Consequently, statistics on such columns do not significantly improve the performance of queries searching for specific keywords within the text.



upvoted 7 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

  **bp\_a\_user** 2 months, 3 weeks ago

**Selected Answer: A**

Collecting statistics on a column containing long values such as string or binary is an expensive operation  
<https://docs.delta.io/latest/optimizations-oss.html>

upvoted 1 times

  **Dileepvikram** 7 months, 3 weeks ago

answer is A

upvoted 2 times

  **mouad\_attaqi** 8 months, 1 week ago

**Selected Answer: A**

A is correct

upvoted 2 times

  **sturcu** 8 months, 1 week ago

**Selected Answer: A**

Collecting statistics on long strings is an expensive operation

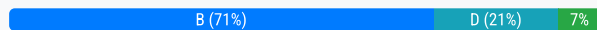
upvoted 2 times

Assuming that the Databricks CLI has been installed and configured correctly, which Databricks CLI command can be used to upload a custom Python Wheel to object storage mounted with the DBFS for use with a production job?

- A. configure
- B. fs
- C. jobs
- D. libraries
- E. workspace

**Suggested Answer: C**

Community vote distribution



**arik90** Highly Voted 9 months, 1 week ago

**Selected Answer: B**

databricks fs cp dist/<...>.whl dbfs:/some/place/appropriate  
upvoted 6 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: B**

Suretly B  
upvoted 1 times

**AlejandroU** 6 months, 2 weeks ago

**Selected Answer: D**

Answer D. The Databricks CLI libraries command is used to manage libraries, including installing custom Python wheels. Specifically, the install subcommand can be used to install a wheel. In contrast, Option B. fs: This command interacts with the Databricks File System (DBFS) to manage files, but it is primarily used for basic file operations (like cp, ls, rm), not specifically for uploading libraries.  
upvoted 1 times

**arekm** 5 months, 4 weeks ago

The question is about upload - answer B.  
upvoted 1 times

**Curious76** 10 months, 1 week ago

**Selected Answer: D**

Here's how you can use the libraries command to upload your wheel:  
Bash

```
databricks libraries upload --file <path_to_wheel_file> --name <library_name>
```

upvoted 1 times

**ojudz08** 10 months, 2 weeks ago

**Selected Answer: C**

this is a bit tricky, question is asked to upload custom Python Wheel, you can use fs command, but since it'll be used in production job, job command might be needed to perform databricks jobs operations?  
<https://docs.databricks.com/en/dev-tools/cli/commands.html>  
upvoted 1 times

**Somesh512** 11 months ago

**Selected Answer: B**

Its asking to upload to DBFS and not install on cluster  
upvoted 2 times

**petrv** 1 year, 1 month ago

**Selected Answer: B**

the question is about copying the file not about installing.

upvoted 3 times

🗨️ 👤 **Enduresoul** 1 year, 1 month ago

**Selected Answer: B**

Answer B is correct:

"... which Databricks CLI command can be used to upload a custom Python Wheel to object storage mounted with the DBFS ..."

The question asks, how to upload the wheel. Not install it or configure it in a job.

<https://docs.databricks.com/en/archive/dev-tools/cli/dbfs-cli.html>

upvoted 4 times

🗨️ 👤 **aragorn\_brego** 1 year, 1 month ago

**Selected Answer: B**

The Databricks CLI fs command is used for interacting with the Databricks File System (DBFS). You can use it to put files into DBFS, which includes uploading custom Python Wheels to a directory in DBFS. The fs command has subcommands like cp that can be used to copy files from your local file system to DBFS, which is backed by an object storage mounted with `dbutils.fs.mount()`.

```
databricks fs cp my_package.whl dbfs:/mnt/my-mount-point/my_package.whl
```

upvoted 2 times

🗨️ 👤 **mouad\_attaqi** 1 year, 2 months ago

**Selected Answer: D**

It is done using the command: `databricks libraries install`

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 2 months ago

**Selected Answer: D**

you can add a library section to the jobs command, but you can install a wheel with the library command

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 2 months ago

```
/api/2.0/libraries/install
```

upvoted 1 times

The business intelligence team has a dashboard configured to track various summary metrics for retail stores. This includes total sales for the previous day alongside totals and averages for a variety of time periods. The fields required to populate this dashboard have the following schema:

```
store_id INT, total_sales_qtd FLOAT, avg_daily_sales_qtd FLOAT, total_sales_ytd
FLOAT, avg_daily_sales_ytd FLOAT, previous_day_sales FLOAT, total_sales_7d FLOAT,
avg_daily_sales_7d FLOAT, updated TIMESTAMP
```

For demand forecasting, the Lakehouse contains a validated table of all itemized sales updated incrementally in near real-time. This table, named `products_per_order`, includes the following fields:

```
store_id INT, order_id INT, product_id INT, quantity INT, price FLOAT,
order_timestamp TIMESTAMP
```

Because reporting on long-term sales trends is less volatile, analysts using the new dashboard only require data to be refreshed once daily. Because the dashboard will be queried interactively by many users throughout a normal business day, it should return results quickly and reduce total compute associated with each materialization.

Which solution meets the expectations of the end users while controlling and limiting possible costs?

- A. Populate the dashboard by configuring a nightly batch job to save the required values as a table overwritten with each update.
- B. Use Structured Streaming to configure a live dashboard against the `products_per_order` table within a Databricks notebook.
- C. Configure a webhook to execute an incremental read against `products_per_order` each time the dashboard is refreshed.
- D. Use the Delta Cache to persist the `products_per_order` table in memory to quickly update the dashboard with each query.
- E. Define a view against the `products_per_order` table and define the dashboard against this view.

**Suggested Answer: A**

Community vote distribution

A (100%)

  **dmov** Highly Voted 1 year ago

**Selected Answer: A**


looks like A to me, as long as they only need the data for the aggregates based on the previous day only  
upvoted 14 times

  **Def21** 11 months, 1 week ago

E - a view, could be an option but it would require computation every time used.  
upvoted 1 times

  **42f87fd** 1 week, 3 days ago

A view would not be cost effective. Needs to be refreshed each time.  
upvoted 1 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

Surely A  
upvoted 1 times

  **srinivasa** 6 months ago

**Selected Answer: D**

Delta cache avoids having to read data from the table every time it's queried during the day.  
upvoted 1 times

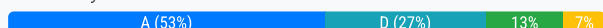
A data ingestion task requires a one-TB JSON dataset to be written out to Parquet with a target part-file size of 512 MB. Because Parquet is being used instead of Delta Lake, built-in file-sizing features such as Auto-Optimize & Auto-Compaction cannot be used.

Which strategy will yield the best performance without shuffling data?

- A. Set `spark.sql.files.maxPartitionBytes` to 512 MB, ingest the data, execute the narrow transformations, and then write to parquet.
- B. Set `spark.sql.shuffle.partitions` to 2,048 partitions ( $1\text{TB} \times 1024 \times 1024 / 512$ ), ingest the data, execute the narrow transformations, optimize the data by sorting it (which automatically repartitions the data), and then write to parquet.
- C. Set `spark.sql.adaptive.advisoryPartitionSizeInBytes` to 512 MB bytes, ingest the data, execute the narrow transformations, coalesce to 2,048 partitions ( $1\text{TB} \times 1024 \times 1024 / 512$ ), and then write to parquet.
- D. Ingest the data, execute the narrow transformations, repartition to 2,048 partitions ( $1\text{TB} \times 1024 \times 1024 / 512$ ), and then write to parquet.
- E. Set `spark.sql.shuffle.partitions` to 512, ingest the data, execute the narrow transformations, and then write to parquet.

**Suggested Answer: B**

Community vote distribution



**aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: A**

This strategy aims to control the size of the output Parquet files without shuffling the data. The `spark.sql.files.maxPartitionBytes` parameter sets the maximum size of a partition that Spark will read. By setting it to 512 MB, you are aligning the read partition size with the desired output file size.

Since the transformations are narrow (meaning they do not require shuffling), the number of partitions should roughly correspond to the number of output files when writing out to Parquet, assuming the data is evenly distributed and there is no data expansion during processing.

upvoted 10 times

**Def21** Highly Voted 1 year, 5 months ago

**Selected Answer: D**

D is the only one that does the trick.

Note, we can not do shuffling.

Wrong answers:

A: `spark.sql.files.maxPartitionBytes` is about reading, not writing. (The maximum number of bytes to pack into a single partition when reading files. This configuration is effective only when using file-based sources such as Parquet, JSON and ORC. )

B: `spark.sql.adaptive.advisoryPartitionSizeInBytes` takes effect while shuffling and sorting does not make sense (The advisory size in bytes of the shuffle partition during adaptive optimization (when `spark.sql.adaptive.enabled` is true). It takes effect when Spark coalesces small shuffle partitions or splits skewed shuffle partition.)

C: Would work but `spark.sql.adaptive.advisoryPartitionSizeInBytes` would need shuffling.

E. `spark.sql.shuffle.partitions` (Configures the number of partitions to use when shuffling data for joins or aggregations.) is not about writing.  
upvoted 6 times

**arekm** 5 months, 4 weeks ago

D does repartition, which the question says we should try to avoid.

upvoted 1 times

**carlosmps** 6 months, 3 weeks ago

`spark.sql.files.maxPartitionBytes` is not just for reading files

upvoted 1 times


**azurefan777** 7 months, 3 weeks ago

Answer D is wrong -> repartition does perform shuffling in Spark. When you use repartition, Spark redistributes the data across the specified number of partitions, which requires moving data between nodes to achieve the new partitioning. Answer A should be correct  
upvoted 4 times

  **KadELbied** Most Recent 2 months ago

**Selected Answer: B**

I found this question in another's Exam test and all of them it looks like B  
upvoted 1 times

  **AlejandroU** 6 months, 1 week ago

**Selected Answer: D**

Answer D. Explicitly repartitioning to 2,048 partitions ensures that the output files are close to the desired size of 512 MB, provided the data distribution is relatively even.

Repartitioning directly addresses the problem by controlling the number of partitions, which directly affects the output file size

Why not option A ? Misinterpretation of `spark.sql.files.maxPartitionBytes` in Option A:



The assessment incorrectly states that this configuration controls the maximum size of files when writing to Parquet. This setting controls the size of partitions when reading data, not during writing.

upvoted 1 times

  **AlejandroU** 6 months, 1 week ago

Given the requirement to avoid shuffling, Option A is the most suitable choice. By setting `spark.sql.files.maxPartitionBytes` to 512 MB, you influence the partitioning during the read phase, which can help in achieving the desired file sizes during the write operation. However, it's important to note that this approach may not guarantee exact file sizes, and some variability may occur. If achieving precise file sizes is critical and shuffling is permissible, Option D would be the preferred strategy.

upvoted 2 times

  **temple1305** 6 months, 3 weeks ago


**Selected Answer: C**

`spark.sql.adaptive.advisoryPartitionSizeInBytes` -

The advisory size in bytes of the shuffle partition during adaptive optimization (when `spark.sql.adaptive.enabled` is true). It takes effect when Spark coalesces small shuffle partitions or splits skewed shuffle partition.

And then we do coalesce - without shuffle - so have to work!

upvoted 1 times

  **nedlo** 8 months, 1 week ago

**Selected Answer: A**

I thought D, but default num of partitions is 200, so you can't do coalesce (2048) (you can't increase num of partitions through coalesce), so it's not possible to do it without repartitioning and shuffle. Only A can be done without Shuffle

upvoted 2 times

  **sdas1** 9 months, 2 weeks ago

Option A

`spark.sql.files.maxPartitionBytes` controls the maximum size of partitions during reading on the Spark cluster, and that reducing this value could lead to more partitions and thus potentially more output files. The key point is that it works best when no shuffles occur, which aligns with the scenario of having narrow transformations only.

upvoted 2 times

  **sdas1** 9 months, 2 weeks ago

Given that no shuffle occurs and you're aiming to control the file sizes during output, adjusting `spark.sql.files.maxPartitionBytes` could help indirectly by determining the partition size for reading. Since the number of input partitions can influence the size of the output files when no shuffle occurs, the partition size may closely match the size of the files being written out.

upvoted 1 times

  **sdas1** 9 months, 2 weeks ago

If the transformations remain narrow, then Spark won't repartition the data unless explicitly instructed to do so (e.g., through a repartition or coalesce operation). In this case, using `spark.sql.files.maxPartitionBytes` to adjust the read partition size to 512 MB could indirectly control the number of output files and ensure they align with the target file size.

upvoted 1 times

  **sdas1** 9 months, 2 weeks ago

Thus, Option A is also a valid strategy:

Set `spark.sql.files.maxPartitionBytes` to 512 MB, process the data with narrow transformations, and write to Parquet.

By reducing the value of `spark.sql.files.maxPartitionBytes`, you ensure more partitions are created during the read phase, leading to output files closer to the desired size, assuming the transformations are narrow and no shuffling occurs.

upvoted 1 times

🗳️ 👤 **vikram12apr** 1 year, 3 months ago

**Selected Answer: A**

D is not correct as it will create 2048 target files of 0.5 MB each

Only A will do the job as it will read this file in 2 partition ( 1 TB =  $512 \times 2$  MB) and as we are not doing any shuffling(not mentioned in option) it will create those many partition file i.e 2 part files

upvoted 1 times

🗳️ 👤 **hal2401me** 1 year, 3 months ago

hey,  $1\text{TB} = 1000\text{GB} = 1^6\text{MB}$ .

upvoted 4 times

🗳️ 👤 **hal2401me** 1 year, 3 months ago

**Selected Answer: D**

ChatGPT says D: This strategy directly addresses the desired part-file size by repartitioning the data. It avoids shuffling during narrow transformations.

Recommended for achieving the desired part-file size without unnecessary shuffling.

upvoted 1 times

🗳️ 👤 **Curious76** 1 year, 4 months ago

**Selected Answer: D**

D is mot suitable.

upvoted 1 times

🗳️ 👤 **vctrhugo** 1 year, 4 months ago

**Selected Answer: A**

This approach ensures that each partition will be approximately the target part-file size, which can improve the efficiency of the data write. It also avoids the need for a shuffle operation, which can be expensive in terms of performance.

upvoted 3 times

🗳️ 👤 **adenis** 1 year, 5 months ago

**Selected Answer: C**

C is correct

upvoted 1 times

🗳️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: A**

Rest of the answers trigger shuffles

upvoted 2 times

🗳️ 👤 **divingbell17** 1 year, 6 months ago

**Selected Answer: A**

A is correct.

The question states Which strategy will yield the best performance without shuffling data.

The other options involve shuffling either manually or through AQE

upvoted 2 times

🗳️ 👤 **911land** 1 year, 6 months ago

C is correct answer

upvoted 1 times

🗳️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: A**

- `spark.sql.files.maxPartitionBytes`: 128MB (The maximum number of bytes to pack into a single partition when reading files. This configuration is effective only when using file-based sources such as Parquet, JSON and ORC.)

upvoted 1 times

🗳️ 👤 **petrv** 1 year, 7 months ago

**Selected Answer: C**

Here's a breakdown of the reasons:

`spark.sql.adaptive.advisoryPartitionSizeInBytes`: This configuration parameter is designed to provide advisory partition sizes for the adaptive query



execution framework. It can help in controlling the partition sizes without triggering unnecessary shuffling.

`coalesce(2048)`: Coalescing to a specific number of partitions after the narrow transformations allows you to control the number of output files without triggering a shuffle. This helps achieve the target part-file size without incurring the overhead of a full shuffle.

Setting a specific target: The strategy outlines the goal of achieving a target part-file size of 512 MB, which aligns with the requirement.  
upvoted 3 times

A junior data engineer has been asked to develop a streaming data pipeline with a grouped aggregation using DataFrame df. The pipeline needs to calculate the average humidity and average temperature for each non-overlapping five-minute interval. Incremental state information should be maintained for 10 minutes for late-arriving data.

Streaming DataFrame df has the following schema:

"device\_id INT, event\_time TIMESTAMP, temp FLOAT, humidity FLOAT"

Code block:

```
df._____  
    .groupBy(  
        window("event_time", "5 minutes").alias("time"),  
        "device_id"  
    )  
    .agg(  
        avg("temp").alias("avg_temp"),  
        avg("humidity").alias("avg_humidity")  
    )  
    .writeStream  
    .format("delta")  
    .saveAsTable("sensor_avg")
```


Choose the response that correctly fills in the blank within the code block to complete this task.

- A. withWatermark("event\_time", "10 minutes")
- B. awaitArrival("event\_time", "10 minutes")
- C. await("event\_time + '10 minutes'")
- D. slidingWindow("event\_time", "10 minutes")
- E. delayWrite("event\_time", "10 minutes")

**Suggested Answer: D**

Community vote distribution

A (100%)

 **aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: A**

To handle late-arriving data in a streaming aggregation, you need to specify a watermark, which tells the streaming query how long to wait for late data. The withWatermark method is used for this purpose in Spark Structured Streaming. It defines the threshold for how late the data can be relative to the latest data that has been seen in the same window.

upvoted 9 times


 **sturcu** Highly Voted 1 year, 8 months ago

**Selected Answer: A**

withWatermark.

There sliding window is doe through the window function

upvoted 9 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times



  **71dfab9** 10 months, 2 weeks ago

**Selected Answer: A**

The `withWatermark` method is used in streaming DataFrames when processing real-time data streams. This method helps in managing stateful operations, such as aggregations, by specifying a time column to use for watermarking. Watermarking is a mechanism to handle late data (data that arrives later than expected) by defining a threshold time window beyond which late data is considered too late to be included in aggregations.

The `slidingWindow` function mentioned in D is not a standard function in Databricks or Apache Spark.

upvoted 1 times

  **Dileepvikram** 1 year, 7 months ago

Answer is A

upvoted 3 times

A data team's Structured Streaming job is configured to calculate running aggregates for item sales to update a downstream marketing dashboard. The marketing team has introduced a new promotion, and they would like to add a new field to track the number of times this promotion code is used for each item. A junior data engineer suggests updating the existing query as follows. Note that proposed changes are in bold.

Original query:

```
df.groupBy("item")
  .agg(count("item").alias("total_count"),
       mean("sale_price").alias("avg_price"))
  .writeStream
  .outputMode("complete")
  .option("checkpointLocation", "/item_agg/__checkpoint")
  .start("/item_agg")
```

Proposed query:

```
df.groupBy("item")
  .agg(count("item").alias("total_count"),
       mean("sale_price").alias("avg_price"))
  .writeStream
  .outputMode("complete")
  .option("checkpointLocation", "/item_agg/__checkpoint")
  .start("/item_agg")
```

Proposed query:

```
.start("/item_agg")
```

Which step must also be completed to put the proposed query into production?

- A. Specify a new checkpointLocation
- B. Increase the shuffle partitions to account for additional aggregates
- C. Run REFRESH TABLE delta: '/item\_agg'
- D. Register the data in the "/item\_agg" directory to the Hive metastore
- E. Remove .option('mergeSchema', 'true') from the streaming write

**Suggested Answer: A**

 **f728f7f**  1 year ago

This question is broken. Proposed query cannot be identified.  
upvoted 24 times

 **AlejandroU**  6 months, 1 week ago

**Selected Answer: A**

Below is the proposed query:

```
df.groupBy("item") .agg(count("item").alias("total_count"), mean("sale_price").alias("avg_price"), count("promo_code = 'NEW MEMBER'" ) .alias("new member_promo")) writeStream .outputMode("complete") .option('mergeSchema', 'true') .option("checkpointLocation", "/item_agg/ checkpoint") .start("/item_agg")
```

Answer A. When updating the schema of a streaming job by adding new fields (like the new\_member\_promo field), it's important to use a new

checkpoint location. This is because the existing checkpoint location is tied to the old schema, and adding a new field could lead to schema mismatch issues.

upvoted 5 times

  **OnlyPraveen** 6 months ago

Thank you! Also check Question #114 which has the Proposed Query image too.



upvoted 1 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times

  **kino\_1994** 6 months, 2 weeks ago

**Selected Answer: A**

Since the new field is a count (an aggregation), it is non-nullable, making the change incompatible with the existing schema. This requires a new checkpointLocation to avoid schema mismatch issues. Additionally, the "mergeSchema=true" option must remain enabled to allow Spark to handle the schema evolution properly.

However, if the field were nullable and not an aggregation, it would be a backward-compatible change, allowing the checkpoint to remain unchanged, as happens with schema evolution in Kafka. In this case, the correct answer is A.

upvoted 2 times

  **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: A**

The given answer is correct.

In case of addition of new cols (or changes) the checkpoint location also needs to change.

upvoted 1 times

A Structured Streaming job deployed to production has been resulting in higher than expected cloud storage costs. At present, during normal execution, each microbatch of data is processed in less than 3s; at least 12 times per minute, a microbatch is processed that contains 0 records. The streaming write was configured using the default trigger settings. The production job is currently scheduled alongside many other Databricks jobs in a workspace with instance pools provisioned to reduce start-up time for jobs with batch execution.

Holding all other variables constant and assuming records need to be processed in less than 10 minutes, which adjustment will meet the requirement?

- A. Set the trigger interval to 3 seconds; the default trigger interval is consuming too many records per batch, resulting in spill to disk that can increase volume costs.
- B. Increase the number of shuffle partitions to maximize parallelism, since the trigger interval cannot be modified without modifying the checkpoint directory.
- C. Set the trigger interval to 10 minutes; each batch calls APIs in the source storage account, so decreasing trigger frequency to maximum allowable threshold should minimize this cost.
- D. Set the trigger interval to 500 milliseconds; setting a small but non-zero trigger interval ensures that the source is not queried too frequently.
- E. Use the trigger once option and configure a Databricks job to execute the query every 10 minutes; this approach minimizes costs for both compute and storage.

**Suggested Answer: C**

Community vote distribution

C (50%)


E (50%)

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

suretly C


upvoted 1 times

 **AlejandroU** 6 months, 1 week ago

**Selected Answer: C**

Answer C. Setting the trigger interval to 10 minutes (option C) directly aligns with the requirement to process records within a 10-minute window. It achieves the same reduction in processing frequency as option E but without the added complexity of job scheduling or reliance on trigger once. Using the trigger once option requires external orchestration (e.g., a scheduled Databricks job) to execute every 10 minutes. This adds operational overhead and potential delays due to job scheduling or startup times, especially in a shared workspace using instance pools.


upvoted 1 times

 **Urcolbz** 6 months, 1 week ago

**Selected Answer: C**

In my opinion, both C and E met the requirements. But the sentence says 'Holding all other variables constant'. This indicates me that E cannot be the solution, as new variables are introduced.


upvoted 2 times

 **benni\_ale** 6 months, 4 weeks ago

**Selected Answer: E**

The fact that the question mentions instance pools provisioned make me guess that we should go for trigger once option otherwise instance pools are useless.

upvoted 1 times

 **pk07** 9 months ago

**Selected Answer: C**

E WRONG. Using trigger once would stop the stream after one execution, not meeting the requirement of continuous processing.

upvoted 2 times

 **practitioner** 10 months, 2 weeks ago

**Selected Answer: E**

E is correct for two reasons:

- 1) we have been using the connection pool that allows us to start our job instantly
- 2) the questions are about reducing costs. Triggering one per 10 minutes allows not to use running VM (as in option C) and to keep the same SLA (due to 1) ) with lower cost for compute as well as for storage (fewer API calls which are not free )

upvoted 1 times

🗨️ 👤 **Er5** 1 year, 2 months ago

required "to be processed in less than 10 minutes".

C. "set the trigger interval to 10 minutes" means Process time + interval > 10 minutes

E. "trigger once", "execute the query every 10 minutes"

upvoted 3 times

🗨️ 👤 **vikram12apr** 1 year, 3 months ago

**Selected Answer: E**

default trigger time is 0.5 seconds

Hence in a minute there are 120 triggers happens

Each trigger consume 3 seconds to complete

now  $120 \times 3 = 360$  seconds = 6 minutes

Hence the job is completing in 6 minutes

Now there is buffer of 4 minutes which can be utilized in compute spin up

but as we are using the spot instances which will further decrease the start up time

I think E is correct option to decrease the cost.

upvoted 2 times

🗨️ 👤 **hidelux** 1 year, 3 months ago

**Selected Answer: E**

The question indicates that they are using instance pools for fast startup time. option C would block a VM permanently which is not intended. E will grab a VM, run the job, and return it to the pool to be available for other jobs mentioned in the question.

upvoted 3 times

🗨️ 👤 **practitioner** 10 months, 2 weeks ago

you are right. But we need to guarantee SLA and for this reason to block VM (with autoscaling) is a good practice

upvoted 1 times

🗨️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: C**

C is more effective than E as E will incur startup time for spinning new job cluster

upvoted 3 times

🗨️ 👤 **ranith** 1 year, 5 months ago

The default trigger interval is 500ms, but the question says it processes batches with 0 records and the avg time to process is 3s. If the requirement is to process under 10 minutes the best option here is to trigger every 3s.

upvoted 1 times

🗨️ 👤 **divingbell17** 1 year, 6 months ago

**Selected Answer: C**

Both C and E meet the requirement to reduce cloud storage cost. E further reduces compute cost however reducing compute cost is not a requirement in the question.

upvoted 2 times

🗨️ 👤 **alexvno** 1 year, 6 months ago

**Selected Answer: C**

For production -> records need to be processed in less than 10 minutes. So we need to schedule each 10 minutes

upvoted 3 times

🗨️ 👤 **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: E**

Given that there are frequent microbatches with 0 records being processed, it indicates that the job is polling the source too often. Using the "trigger once" option would allow each microbatch to process all available data and then stop. By scheduling the job to run every 10 minutes, you ensure that the system is not constantly checking for new data when there is none, thus reducing the number of read operations from the source storage and potentially reducing costs associated with those reads.

upvoted 4 times

🗨️ 👤 **Gulenur\_GS** 1 year, 6 months ago

in this case why not C? Processing trigger in 10 min ensures the same I guess..  
upvoted 1 times



Which statement describes the correct use of `pyspark.sql.functions.broadcast`?

- A. It marks a column as having low enough cardinality to properly map distinct values to available partitions, allowing a broadcast join.
- B. It marks a column as small enough to store in memory on all executors, allowing a broadcast join.
- C. It caches a copy of the indicated table on attached storage volumes for all active clusters within a Databricks workspace.
- D. It marks a DataFrame as small enough to store in memory on all executors, allowing a broadcast join.
- E. It caches a copy of the indicated table on all nodes in the cluster for use in all future queries during the cluster lifetime.

**Suggested Answer: C**

Community vote distribution

D (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

Selected Answer: D

suretly D

upvoted 1 times

🗳️ 👤 **Freyr** 7 months ago

Selected Answer: D

Correct Answer: D. It marks a DataFrame as small enough to store in memory on all executors, allowing a broadcast join.

Reference: <https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/api/pyspark.sql.functions.broadcast.html>

upvoted 3 times

🗳️ 👤 **aragorn\_brego** 1 year, 1 month ago

Selected Answer: D

The broadcast function in PySpark is used in the context of joins. When you mark a DataFrame with broadcast, Spark tries to send this DataFrame to all worker nodes so that it can be joined with another DataFrame without shuffling the larger DataFrame across the nodes. This is particularly beneficial when the DataFrame is small enough to fit into the memory of each node. It helps to optimize the join process by reducing the amount of data that needs to be shuffled across the cluster, which can be a very expensive operation in terms of computation and time.

upvoted 3 times

🗳️ 👤 **Dileepvikram** 1 year, 1 month ago

Answer is D

upvoted 1 times

🗳️ 👤 **PearApple** 1 year, 1 month ago

The answer is D

upvoted 1 times

🗳️ 👤 **hm358** 1 year, 2 months ago

Selected Answer: D

<https://spark.apache.org/docs/3.1.3/api/python/reference/api/pyspark.sql.functions.broadcast.html>

upvoted 2 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

Selected Answer: D

Marks a DataFrame as small enough for use in broadcast joins.

upvoted 3 times

A data engineer is configuring a pipeline that will potentially see late-arriving, duplicate records.

In addition to de-duplicating records within the batch, which of the following approaches allows the data engineer to deduplicate data against previously processed records as it is inserted into a Delta table?

- A. Set the configuration `delta.deduplicate = true`.
- B. VACUUM the Delta table after each batch completes.
- C. Perform an insert-only merge with a matching condition on a unique key.
- D. Perform a full outer join on a unique key and overwrite existing data.
- E. Rely on Delta Lake schema enforcement to prevent duplicate records.

**Suggested Answer: D**

Community vote distribution

C (100%)

🗳️ **aragorn\_brego** Highly Voted 1 year, 7 months ago

**Selected Answer: C**

To handle deduplication against previously processed records in a Delta table, the MERGE INTO command can be used to perform an upsert operation. This means that if the incoming data has a record that matches an existing record based on a unique key, the MERGE INTO operation can update the existing record (if needed) or simply ignore the duplicate. If there is no match (i.e., the record is new), then the record will be inserted  
upvoted 5 times

🗳️ **sturcu** Highly Voted 1 year, 8 months ago

**Selected Answer: C**

Merge, when not match insert  
upvoted 5 times

🗳️ **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: C**

suretly C  
upvoted 1 times

🗳️ **\_lene\_** 5 months, 2 weeks ago

**Selected Answer: C**

this question was in the Databricks DE Professional exam guide  
upvoted 1 times

🗳️ **hebied** 7 months ago

**Selected Answer: D**

Option C is tricky since it should be merge on on Not match condition rather than matching .. Since Option D is more suitable  
upvoted 1 times

🗳️ **60ties** 1 year, 7 months ago

**Selected Answer: C**

answer is C  
upvoted 2 times

🗳️ **Dileepvikram** 1 year, 7 months ago

Answer is C  
upvoted 2 times

🗳️ **hm358** 1 year, 8 months ago

**Selected Answer: C**



merge will be more efficient  
upvoted 2 times

🗳️ **Crocjun** 1 year, 8 months ago

C

Reference: file:///C:/Users/yuen1/Downloads/databricks-certified-data-engineer-professional-exam-guide.pdf

upvoted 1 times

  **mouad\_attaqi** 1 year, 8 months ago

you are referencing a local pdf in your computer !!!

upvoted 9 times

A data pipeline uses Structured Streaming to ingest data from Apache Kafka to Delta Lake. Data is being stored in a bronze table, and includes the Kafka-generated timestamp, key, and value. Three months after the pipeline is deployed, the data engineering team has noticed some latency issues during certain times of the day.

A senior data engineer updates the Delta Table's schema and ingestion logic to include the current timestamp (as recorded by Apache Spark) as well as the Kafka topic and partition. The team plans to use these additional metadata fields to diagnose the transient processing delays.

Which limitation will the team face while diagnosing this problem?

- A. New fields will not be computed for historic records.
- B. Spark cannot capture the topic and partition fields from a Kafka source.
- C. New fields cannot be added to a production Delta table.
- D. Updating the table schema will invalidate the Delta transaction log metadata.
- E. Updating the table schema requires a default value provided for each field added.

**Suggested Answer: A**

Community vote distribution

A (100%)

  **dmov** Highly Voted 1 year ago



**Selected Answer: A**

Looks like A to me. Does anyone think otherwise?  
upvoted 8 times

  **vctrhugo** Highly Voted 10 months, 4 weeks ago

**Selected Answer: A**

When the schema of a Delta table is updated to include new fields, these fields will only be populated for new records ingested after the schema update. The new fields will not be retroactively computed for historic records already stored in the Delta table. Therefore, the additional metadata fields (current timestamp, Kafka topic, and partition) will not exist in the historic data, limiting the scope of the diagnosis to new data ingested after the schema update.  
upvoted 6 times

  **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: A**

suretly A  
upvoted 1 times

  **RandomForest** 5 months, 2 weeks ago

**Selected Answer: A**

The correct answer is A.  
upvoted 1 times

In order to facilitate near real-time workloads, a data engineer is creating a helper function to leverage the schema detection and evolution functionality of Databricks Auto Loader. The desired function will automatically detect the schema of the source directly, incrementally process JSON files as they arrive in a source directory, and automatically evolve the schema of the table when new fields are detected.

The function is displayed below with a blank:

```
def auto_load_json(source_path: str,
                  checkpoint_path: str,
                  target_table_path: str):
    (spark.readStream
     .format("cloudFiles")
     .option("cloudFiles.format", "json")
     .option("cloudFiles.schemaLocation", checkpoint_path)
     .load(source_path)
     _____
    )
```

Which response correctly fills in the blank to meet the specified requirements?

- .writeStream
- A. .option("mergeSchema", True)
- .start(target\_table\_path)
- .writeStream
- .option("checkpointLocation", checkpoint\_path)
- B. .option("mergeSchema", True)
- .trigger(once=True)
- .start(target\_table\_path)
- .write
- .option("checkpointLocation", checkpoint\_path)
- C. .option("mergeSchema", True)
- .outputMode("append")
- .save(target\_table\_path)
- .write
- .option("mergeSchema", True)
- D. .mode("append")
- .save(target\_table\_path)
- .writeStream
- .option("checkpointLocation", checkpoint\_path)
- E. .option("mergeSchema", True)
- .start(target\_table\_path)

**Suggested Answer: C**

Community vote distribution

E (100%)

☐ **KadELbied** 1 month, 3 weeks ago

**Selected Answer: E**

suretly E

upvoted 1 times

☐ **benni\_ale** 6 months, 3 weeks ago

**Selected Answer: E**

Evolve Schema = mergeSchema option is needed ; Incrementally = checkpointing is needed; Real-Time = WriteStream with default trigger . The only option that catches all of these is E

upvoted 2 times

🗳️ 👤 **35fd6dd** 10 months, 3 weeks ago

**Selected Answer: E**

write is not for spark streaming

upvoted 2 times

🗳️ 👤 **Freyr** 1 year, 1 month ago

**Selected Answer: E**

Reference: <https://docs.databricks.com/en/ingestion/auto-loader/schema.html>

writeStream: Ensures real-time streaming write capabilities, which is essential f

or near real-time workloads.

checkpointLocation: Necessary for fault tolerance and tracking progress.

mergeSchema: Ensures automatic schema evolution, allowing new columns to be detected and added to the target table.

Why Option 'C ' is incorrect?

Uses write instead of writeStream, which is for batch processing, making it inappropriate for real-time streaming.

Why Option 'B ' is incorrect?

Although it includes checkpointLocation and mergeSchema, the addition of trigger(once=True) is not necessary in this context, and it is better suited for batch-like processing.

Reference: <https://docs.databricks.com/en/ingestion/auto-loader/schema.html>

upvoted 4 times

🗳️ 👤 **vikram12apr** 1 year, 3 months ago

**Selected Answer: E**

streamRead & StreamWrite shares the schema using checkpoint location

so cloudFiles.schemaLocation needs to be same for checkpointLocation so that we dont need to specify it manually

also mergeSchema True make sure if any new column detected , it will be added in the target table

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

upvoted 2 times

🗳️ 👤 **hal2401me** 1 year, 3 months ago

**Selected Answer: E**

<https://notebooks.databricks.com/demos/auto-loader/01-Auto-loader-schema-evolution-Ingestion.html>

upvoted 2 times

🗳️ 👤 **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: E**

This response correctly fills in the blank to meet the specified requirements of using Databricks Auto Loader for automatic schema detection and evolution in a near real-time streaming context.

upvoted 1 times

🗳️ 👤 **AzureDE2522** 1 year, 7 months ago

**Selected Answer: E**

Please refer: <https://docs.databricks.com/en/ingestion/auto-loader/schema.html>

upvoted 3 times

🗳️ 👤 **Dileepvikram** 1 year, 7 months ago

It does not mention to write as stream, it mentions to write incrementally, so option C looks correct for me

upvoted 1 times

🗳️ 👤 **mouad\_attaqi** 1 year, 8 months ago

**Selected Answer: E**

Correct answer is E, it is a streaming write, and the default outputMode is Append (so if it's optional in this case)

upvoted 2 times

🗳️ 👤 **sturcu** 1 year, 8 months ago

there is a type in the statement. Is it schema or checkpoint ?

Provided answer is not correct. It has to be a writestream, with mode append

upvoted 1 times

The data engineering team maintains the following code:

```
import pyspark.sql.functions as F

(spark.table("silver_customer_sales")
 .groupBy("customer_id")
 .agg(
     F.min("sale_date").alias("first_transaction_date"),
     F.max("sale_date").alias("last_transaction_date"),
     F.mean("sale_total").alias("average_sales"),
     F.countDistinct("order_id").alias("total_orders"),
     F.sum("sale_total").alias("lifetime_value")
 ).write
 .mode("overwrite")
 .table("gold_customer_lifetime_sales_summary")
)
```

Assuming that this code produces logically correct results and the data in the source table has been de-duplicated and validated, which statement describes what will occur when this code is executed?

- A. The silver\_customer\_sales table will be overwritten by aggregated values calculated from all records in the gold\_customer\_lifetime\_sales\_summary table as a batch job.
- B. A batch job will update the gold\_customer\_lifetime\_sales\_summary table, replacing only those rows that have different values than the current version of the table, using customer\_id as the primary key.
- C. The gold\_customer\_lifetime\_sales\_summary table will be overwritten by aggregated values calculated from all records in the silver\_customer\_sales table as a batch job.
- D. An incremental job will leverage running information in the state store to update aggregate values in the gold\_customer\_lifetime\_sales\_summary table.
- E. An incremental job will detect if new rows have been written to the silver\_customer\_sales table; if new rows are detected, all aggregates will be recalculated and used to overwrite the gold\_customer\_lifetime\_sales\_summary table.

**Suggested Answer: E**

Community vote distribution


C (100%)

 **aragorn\_brego** Highly Voted 1 year, 1 month ago

**Selected Answer: C**

The code is performing a batch aggregation operation on the "silver\_customer\_sales" table grouped by "customer\_id". It calculates the first and last transaction dates, the average sales, the total number of distinct orders, and the lifetime value of sales for each customer. The .mode("overwrite") operation specifies that the output table "gold\_customer\_lifetime\_sales\_summary" should be overwritten with the result of this aggregation. This means that every time this code runs, it will replace the existing "gold\_customer\_lifetime\_sales\_summary" table with a new version that reflects the current state of the "silver\_customer\_sales" table.

upvoted 7 times

 **sainandam** Most Recent 3 days, 15 hours ago

**Selected Answer: D**

Writing data to a folder does not register an External table.

upvoted 1 times

 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: C**

Surety C

upvoted 1 times



🗨️ 👤 **hal2401me** 9 months, 4 weeks ago

**Selected Answer: C**

C. there's nowhere implicating streaming.

upvoted 1 times

🗨️ 👤 **Dileepvikram** 1 year, 1 month ago

C is the answer

upvoted 1 times

🗨️ 👤 **mouad\_attaqi** 1 year, 2 months ago

**Selected Answer: C**

Correct Answer is C, it is an overwrite mode

upvoted 3 times

🗨️ 👤 **sturcu** 1 year, 2 months ago

**Selected Answer: C**

it does overwrite, so no incremental load

upvoted 4 times

The data architect has mandated that all tables in the Lakehouse should be configured as external (also known as "unmanaged") Delta Lake tables.

Which approach will ensure that this requirement is met?

- A. When a database is being created, make sure that the LOCATION keyword is used.
- B. When configuring an external data warehouse for all table storage, leverage Databricks for all ELT.
- C. When data is saved to a table, make sure that a full file path is specified alongside the Delta format.
- D. When tables are created, make sure that the EXTERNAL keyword is used in the CREATE TABLE statement.
- E. When the workspace is being configured, make sure that external cloud object storage has been mounted.

**Suggested Answer: D**



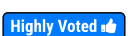
Community vote distribution

C (100%)

  **sturcu**  1 year, 8 months ago

Non of the provided.

It should be: When a table is created, make sure LOCATION is provided  
upvoted 8 times

  **vcrrhugo**  1 year, 4 months ago



**Selected Answer: C**

In Delta Lake, an external (or unmanaged) table is a table created outside of the data lake but is still accessible from the data lake. The data for external tables is stored in a location specified by the user, not in the default directory of the data lake. When you save data to an external table, you need to specify the full file path where the data will be stored. This makes the table "external" because the data itself is not managed by Delta Lake, only the metadata is. This is why specifying a full file path alongside the Delta format when saving data to a table will ensure that the table is configured as an external Delta Lake table.  
upvoted 5 times

  **KadELbied**  1 month, 3 weeks ago

**Selected Answer: C**

Suretly C  
upvoted 1 times

  **Sriramiyer92** 6 months, 2 weeks ago

**Selected Answer: C**

Folks note:

While creating a table - Use of External keyword - Non Mandatory.


Mentioning Location and providing a path - Mandatory.

In option C, it is not mentioned explicitly that Location keyword is used. But since the path is provided.. implies the use of Location keyword indirectly. The devil is in the details. :)

upvoted 3 times

  **hvj** 9 months, 3 weeks ago

'create external table' statement is using in HIVE, so C is correct.  
upvoted 1 times

  **jkhan2405** 1 year, 5 months ago

**Selected Answer: C**

C is correct.  
upvoted 2 times

  **JamesWright** 1 year, 6 months ago

C is correct  
upvoted 1 times

  **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: C**

Here's why the other options may not ensure the requirement is met:

D. Delta Lake does not use the EXTERNAL keyword in the same way as some other SQL-based systems. In Delta Lake, whether a table is external is determined by where the data files are stored, not by a keyword in the CREATE TABLE statement.

%sql

```
CREATE TABLE f1_demo.results_external
USING DELTA
LOCATION '/mnt/formula1dl/demo/results_external'
```

upvoted 3 times

  **Dileepvikram** 1 year, 7 months ago



possible answer is C

upvoted 1 times

  **Laraujo2022** 1 year, 7 months ago



I think it should be A because when a database is created using a location all tables within this database are automatically assign as unmanaged tables.

upvoted 1 times

  **60ties** 1 year, 7 months ago

Not quite. Test & see. The tables are 'managed' though database creation has 'LOCATION' keyword. C is best.

upvoted 5 times

  **sturcu** 1 year, 8 months ago

**Selected Answer: C**

provide path (LOCATION)

upvoted 1 times

  **mouad\_attaqi** 1 year, 8 months ago

**Selected Answer: C**

C is plausible answer, as in this case we are writing the data to an external location

upvoted 1 times

The marketing team is looking to share data in an aggregate table with the sales organization, but the field names used by the teams do not match, and a number of marketing-specific fields have not been approved for the sales org.

Which of the following solutions addresses the situation while emphasizing simplicity?

- A. Create a view on the marketing table selecting only those fields approved for the sales team; alias the names of any fields that should be standardized to the sales naming conventions.
- B. Create a new table with the required schema and use Delta Lake's DEEP CLONE functionality to sync up changes committed to one table to the corresponding table.
- C. Use a CTAS statement to create a derivative table from the marketing table; configure a production job to propagate changes.
- D. Add a parallel table write to the current production pipeline, updating a new sales table that varies as required from the marketing table.
- E. Instruct the marketing team to download results as a CSV and email them to the sales organization.

**Suggested Answer: A**

*Community vote distribution*

A (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

suretly A

upvoted 1 times

🗳️ 👤 **Hadiler** 11 months, 1 week ago

**Selected Answer: A**

A is the simplest one

upvoted 1 times

🗳️ 👤 **vctrhugo** 1 year, 4 months ago

**Selected Answer: A**

Creating a view is a simple and efficient way to provide access to a subset of data from a table. In this case, the view can be configured to include only the fields that have been approved for the sales team. Additionally, any fields that need to be renamed to match the sales team's naming conventions can be aliased in the view. This approach does not require the creation of additional tables or the configuration of jobs to sync data, making it a relatively straightforward solution. However, it's important to note that views do not physically store data, so any changes to the underlying marketing table will be reflected in the view. This means that the sales team will always have access to the most up-to-date approved data.

upvoted 4 times

🗳️ 👤 **spaceexplorer** 1 year, 5 months ago

**Selected Answer: A**

A is the simplest

upvoted 1 times

🗳️ 👤 **dmov** 1 year, 6 months ago

**Selected Answer: A**

Looks like A to me

upvoted 2 times

A CHECK constraint has been successfully added to the Delta table named activity\_details using the following logic:

```
ALTER TABLE activity_details
ADD CONSTRAINT valid_coordinates
CHECK (
  latitude >= -90 AND
  latitude <= 90 AND
  longitude >= -180 AND
  longitude <= 180);
```

A batch job is attempting to insert new records to the table, including a record where latitude = 45.50 and longitude = 212.67.

Which statement describes the outcome of this batch insert?

- A. The write will fail when the violating record is reached; any records previously processed will be recorded to the target table.
- B. The write will fail completely because of the constraint violation and no records will be inserted into the target table.
- C. The write will insert all records except those that violate the table constraints; the violating records will be recorded to a quarantine table.
- D. The write will include all records in the target table; any violations will be indicated in the boolean column named valid\_coordinates.
- E. The write will insert all records except those that violate the table constraints; the violating records will be reported in a warning log.

**Suggested Answer: D**

Community vote distribution

B (100%)

 **aragorn\_brego** Highly Voted 1 year, 1 month ago

**Selected Answer: B**

In systems that support atomic transactions, such as Delta Lake, when a batch operation encounters a record that violates a CHECK constraint, the entire operation fails, and no records are inserted, including those that do not violate the constraint. This is to ensure the atomicity of the transaction, meaning that either all the changes are committed, or none are, maintaining data integrity. The record with a longitude of 212.67 violates the constraint because longitude values must be between -180 and 180 degrees.

upvoted 5 times

 **vctrhugo** Highly Voted 10 months, 4 weeks ago

**Selected Answer: B**

In Delta Lake, when a batch job attempts to insert records into a table that has a CHECK constraint, if any record violates the constraint, the entire write operation fails. This is because Delta Lake enforces strong transactional guarantees, which means that either all changes in a transaction are saved, or none are.

upvoted 5 times

 **KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: B**

Surely B

upvoted 1 times

 **spaceexplorer** 11 months, 1 week ago

**Selected Answer: B**

B is correct

upvoted 1 times

 **Dileepvikram** 1 year, 1 month ago

B is the answer

upvoted 4 times

 **sturcu** 1 year, 2 months ago

**Selected Answer: B**

B is the answer  
upvoted 4 times

  **PearApple** 1 year, 2 months ago

B is the ans  
upvoted 3 times

A junior data engineer has manually configured a series of jobs using the Databricks Jobs UI. Upon reviewing their work, the engineer realizes that they are listed as the "Owner" for each job. They attempt to transfer "Owner" privileges to the "DevOps" group, but cannot successfully accomplish this task.

Which statement explains what is preventing this privilege transfer?

- A. Databricks jobs must have exactly one owner; "Owner" privileges cannot be assigned to a group.
- B. The creator of a Databricks job will always have "Owner" privileges; this configuration cannot be changed.
- C. Other than the default "admins" group, only individual users can be granted privileges on jobs.
- D. A user can only transfer job ownership to a group if they are also a member of that group.
- E. Only workspace administrators can grant "Owner" privileges to a group.

**Suggested Answer: A**

*Community vote distribution*

A (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: A**

Surely A

upvoted 1 times

🗳️ 👤 **hal2401me** 9 months, 4 weeks ago

**Selected Answer: A**

did a test. "group cannot be owner" is displayed.

upvoted 3 times

🗳️ 👤 **vctrhugo** 10 months, 4 weeks ago

**Selected Answer: A**

In Databricks, each job must have exactly one owner, which is typically the user who created the job. This "Owner" privilege allows the user to perform any action on the job, including modifying its settings or deleting it. However, this privilege cannot be assigned to a group. If you want to allow multiple users or a group of users to manage a job, you can use ACLs (Access Control Lists) to grant them the necessary permissions. But the "Owner" privilege will still remain with the individual user who created the job.

upvoted 1 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

**Selected Answer: A**

Correct

A job cannot have more than one owner. A job cannot have a group as an owner

upvoted 4 times

All records from an Apache Kafka producer are being ingested into a single Delta Lake table with the following schema:

key BINARY, value BINARY, topic STRING, partition LONG, offset LONG, timestamp LONG

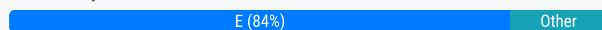
There are 5 unique topics being ingested. Only the "registration" topic contains Personal Identifiable Information (PII). The company wishes to restrict access to PII. The company also wishes to only retain records containing PII in this table for 14 days after initial ingestion. However, for non-PII information, it would like to retain these records indefinitely.

Which of the following solutions meets the requirements?

- A. All data should be deleted biweekly; Delta Lake's time travel functionality should be leveraged to maintain a history of non-PII information.
- B. Data should be partitioned by the registration field, allowing ACLs and delete statements to be set for the PII directory.
- C. Because the value field is stored as binary data, this information is not considered PII and no special precautions should be taken.
- D. Separate object storage containers should be specified based on the partition field, allowing isolation at the storage level.
- E. Data should be partitioned by the topic field, allowing ACLs and delete statements to leverage partition boundaries.

**Suggested Answer: D**

Community vote distribution



**mouad\_attaqi** Highly Voted 1 year, 8 months ago

**Selected Answer: E**

I think answer E is correct, as by default partitioning by a column will create a separate folder for each subset data linked to the partition  
upvoted 13 times

**KadELbied** Most Recent 1 month, 3 weeks ago

**Selected Answer: E**

Surely E  
upvoted 1 times

**benni\_ale** 6 months, 3 weeks ago

**Selected Answer: E**

Partitioning by topic field let delete queries leverage partitioning boundaries  
upvoted 2 times

**benni\_ale** 8 months, 1 week ago

**Selected Answer: E**

E E E E  
upvoted 1 times

**ojudz08** 1 year, 4 months ago

**Selected Answer: D**

i think it's best to isolate the storage to avoid mistakenly deleting tables in the same storage so I go with D  
upvoted 1 times

**spaceexplorer** 1 year, 5 months ago

**Selected Answer: E**

E is correct  
upvoted 1 times

**ervinshang** 1 year, 6 months ago

**Selected Answer: E**

E is correct  
upvoted 2 times

**aragorn\_brego** 1 year, 7 months ago

**Selected Answer: E**



Partitioning data by the topic field would allow the data engineering team to apply access control lists (ACLs) to restrict access to the partition containing the "registration" topic, which holds PII. Furthermore, the team can set up automated deletion policies that specifically target the partition with PII data to delete records after 14 days, without affecting the data in other partitions. This approach meets both the privacy requirements for PII and the data retention goals for non-PII information.

upvoted 2 times

🗨️ 👤 **Dileepvikram** 1 year, 7 months ago

I think answer is E

upvoted 3 times

🗨️ 👤 **[Removed]** 1 year, 8 months ago

**Selected Answer: B**

The solution that meets the requirements is: B. Data should be partitioned by the registration field, allowing ACLs and delete statements to be set for the PII directory.

Partitioning the data by the registration field allows the directory containing PII records to be isolated and access restricted via ACLs. Additionally, the data retention requirements can be met by setting up a separate job or process to remove PII records that are 14 days old. For non-PII records, they can be retained indefinitely utilizing Delta Lake's time travel functionality.

upvoted 1 times

🗨️ 👤 **mouad\_attaqi** 1 year, 8 months ago

There is no such thing as Registration field, it's a distinct topic

upvoted 2 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

you cannot restrict privileges. with ACLs on a partition. Documentations states that Securable objects in the Hive metastore are: DB, Tables, Views, Functions: <https://docs.databricks.com/en/data-governance/table-acls/object-privileges.html#securable-objects>

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

**Selected Answer: D**

Correct

upvoted 1 times

🗨️ 👤 **sturcu** 1 year, 8 months ago

<https://docs.databricks.com/en/data-governance/table-acls/object-privileges.html#securable-objects>

upvoted 1 times

The data architect has decided that once data has been ingested from external sources into the Databricks Lakehouse, table access controls will be leveraged to manage permissions for all production tables and views.

The following logic was executed to grant privileges for interactive queries on a production database to the core engineering group.

```
GRANT USAGE ON DATABASE prod TO eng;  
GRANT SELECT ON DATABASE prod TO eng;
```

Assuming these are the only privileges that have been granted to the eng group and that these users are not workspace administrators, which statement describes their privileges?

- A. Group members have full permissions on the prod database and can also assign permissions to other users or groups.
- B. Group members are able to list all tables in the prod database but are not able to see the results of any queries on those tables.
- C. Group members are able to query and modify all tables and views in the prod database, but cannot create new tables or views.
- D. Group members are able to query all tables and views in the prod database, but cannot create or edit anything in the database.
- E. Group members are able to create, query, and modify all tables and views in the prod database, but cannot define custom functions.

**Suggested Answer: E**

Community vote distribution

D (100%)

  **sturcu** Highly Voted 1 year, 8 months ago



**Selected Answer: D**

Usage and Select .....basically they can only select  
upvoted 6 times

  **KadELbied** Most Recent 1 month, 3 weeks ago


**Selected Answer: D**

Surely D  
upvoted 1 times

  **benni\_ale** 7 months, 4 weeks ago


**Selected Answer: D**

D is ok  
upvoted 1 times

  **Curious76** 1 year, 4 months ago



**Selected Answer: D**

D is correct  
upvoted 1 times

  **vctrhugo** 1 year, 4 months ago

**Selected Answer: D**

The GRANT statements provided in the logic grant the USAGE privilege, allowing the group members to see the existence of the database, and the SELECT privilege, allowing them to query tables and views. However, they do not have permissions to create or edit anything in the database. Therefore, the correct description is that group members can query all tables and views in the prod database but cannot create or edit any objects in the database.  
upvoted 1 times

  **divingbell17** 1 year, 6 months ago

**Selected Answer: D**



D is correct assuming unity catalog is not enabled  
upvoted 1 times

  **aragorn\_brego** 1 year, 7 months ago

**Selected Answer: D**

The GRANT USAGE ON DATABASE statement gives the eng group the ability to access the prod database. This means they can enter the database context and list the tables. The GRANT SELECT ON DATABASE statement additionally grants them permission to perform SELECT queries on all existing tables and views within the prod database. However, these privileges do not include creating new tables or views, modifying existing tables, or assigning permissions to other users or groups.

upvoted 3 times

  **Dileepvikram** 1 year, 7 months ago

D is answer

upvoted 4 times

A distributed team of data analysts share computing resources on an interactive cluster with autoscaling configured. In order to better manage costs and query throughput, the workspace administrator is hoping to evaluate whether cluster upscaling is caused by many concurrent users or resource-intensive queries.

In which location can one review the timeline for cluster resizing events?

- A. Workspace audit logs
- B. Driver's log file
- C. Ganglia
- D. Cluster Event Log
- E. Executor's log file

**Suggested Answer: C**

Community vote distribution

D (100%)

🗳️ 👤 **KadELbied** 1 month, 3 weeks ago

**Selected Answer: D**

Surely D

upvoted 1 times

🗳️ 👤 **Curious76** 10 months, 1 week ago

**Selected Answer: D**

The Cluster Event Log provides detailed information about various events affecting the cluster throughout its lifecycle, including cluster creation, restarts, termination, and resizing events. It displays the timestamp, event type (e.g., "CLUSTER\_RESIZED"), and relevant details for each event, allowing the administrator to review the timeline for cluster scaling behavior and identify potential patterns related to user activity or resource-intensive queries.

upvoted 3 times

🗳️ 👤 **vctrhugo** 10 months, 4 weeks ago

**Selected Answer: D**

The timeline for cluster resizing events can be reviewed in the Cluster Event Log. This log provides information about cluster scaling events, including when the cluster is scaled up or down. You can access this information to understand the reasons behind autoscaling events and whether they are triggered by many concurrent users or resource-intensive queries.

upvoted 1 times

🗳️ 👤 **alexvno** 1 year ago

**Selected Answer: D**

Cluster event log

upvoted 2 times

🗳️ 👤 **aragorn\_brego** 1 year, 1 month ago

**Selected Answer: D**

The Cluster Event Log in Databricks will show the timeline for cluster resizing events, including details about when and why a cluster was resized (scaled up or down). This log would help the workspace administrator determine the causes of cluster scaling, whether due to many concurrent users submitting jobs or a few users running resource-intensive queries.

less suitable:

C. Ganglia provides metrics on system-level performance, such as CPU and memory usage, but does not log specific cluster scaling events.

upvoted 2 times

🗳️ 👤 **PearApple** 1 year, 1 month ago

cluster event log. D

upvoted 2 times

🗳️ 👤 **sturcu** 1 year, 2 months ago

Selected Answer: D

Cluster Event Log

upvoted 3 times