[Q33-Q51 2024 Updated Databricks-Certified-Data-Engineer-Associate Tests Engine pdf - All Free Dumps Guaranteed!

2024 Updated Databricks-Certified-Data-Engineer-Associate Tests Engine pdf - All Free Dumps Guaranteed!
Latest Databricks Certification Databricks-Certified-Data-Engineer-Associate Actual Free Exam Questions

NEW QUESTION 33

A data engineer has developed a data pipeline to ingest data from a JSON source using Auto Loader, but the engineer has not provided any type inference or schema hints in their pipeline. Upon reviewing the data, the data engineer has noticed that all of the columns in the target table are of the string type despite some of the fields only including float or boolean values.

Which of the following describes why Auto Loader inferred all of the columns to be of the string type?
* There was a type mismatch between the specific schema and the inferred schema
* JSON data is a text-based format
* Auto Loader only works with string data
* All of the fields had at least one null value
* Auto Loader cannot infer the schema of ingested data

Explanation

JSON data is a text-based format that uses strings to represent all values. When Auto Loader infers the schema of JSON data, it assumes that all values are strings. This is because Auto Loader cannot determine the type of a value based on its string representation. https://docs.databricks.com/en/ingestion/auto-loader/schema.html Forexample, the following JSON string represents a value that is logically a boolean: JSON “true” Use code with caution. Learn more However, Auto Loader would infer that the type of this value is string. This is because Auto Loader cannot determine that the value is a boolean based on its string representation. In order to get Auto Loader to infer the correct types for columns, the data engineer can provide type inference or schema hints. Type inference hints can be used to specify the types of specific columns. Schema hints can be used to provide the entire schema of the data.

Therefore, the correct answer is B. JSON data is a text-based format.

NEW QUESTION 34

A data engineer has realized that they made a mistake when making a daily update to a table. They need to use Delta time travel to restore the table to a version that is 3 days old. However, when the data engineer attempts to time travel to the older version, they are unable to restore the data because the data files have been deleted.

Which of the following explains why the data files are no longer present?
* The VACUUM command was run on the table
* The TIME TRAVEL command was run on the table
* The DELETE HISTORY command was run on the table
* The OPTIMIZE command was nun on the table
* The HISTORY command was run on the table

The VACUUM command is used to remove files that are no longer referenced by a Delta table and are older than the retention threshold1. The default retention period is 7 days2, but it can be changed by setting the delta.logRetentionDuration and delta.deletedFileRetentionDuration configurations3. If the VACUUM command was run on the table with a retention period shorter than 3 days, then the data files that were needed to restore the table to a 3-day-old version would have been deleted. The other commands do not delete data files from the table. The TIME TRAVEL command is used to query a historical version of the table4. The DELETE HISTORY command is not a valid command in Delta Lake. The OPTIMIZE command is used to improve the performance of the table by compacting small files into larger ones5. The HISTORY command is used to retrieve information about the operations performed on the table. References: 1: VACUUM | Databricks on AWS 2: Work with Delta Lake table history | Databricks on AWS 3: [Delta Lake configuration | Databricks on AWS] 4: Work with Delta Lake table history – Azure Databricks 5: [OPTIMIZE | Databricks on AWS] : [HISTORY | Databricks on AWS]

NEW QUESTION 35

A data engineer is running code in a Databricks Repo that is cloned from a central Git repository. A colleague of the data engineer informs them that changes have been made and synced to the central Git repository. The data engineer now needs to sync their Databricks Repo to get the changes from the central Git repository.

Which of the following Git operations does the data engineer need to run to accomplish this task?
* Merge
* Push
* Pull
* Commit
* Clone

Explanation

From the docs:

In Databricks Repos, you can use Git functionality to:

Clone, push to, and pull from a remote Git repository.

Create and manage branches for development work, including merging, rebasing, and resolving conflicts.

Create notebooks—including IPYNB notebooks—and edit them and other files.

Visually compare differences upon commit and resolve merge conflicts.

Source: https://docs.databricks.com/en/repos/index.html

NEW QUESTION 36

A data engineering team has two tables. The first table march_transactions is a collection of all retail transactions in the month of March. The second table april_transactions is a collection of all retail transactions in the month of April. There are no duplicate records between the tables.

Which of the following commands should be run to create a new table all_transactions that contains all records from march_transactions and april_transactions without duplicate records?
* CREATE TABLE all_transactions AS

SELECT * FROM march_transactions

INNER JOIN SELECT * FROM april_transactions;
* CREATE TABLE all_transactions AS

SELECT * FROM march_transactions

UNION SELECT * FROM april_transactions;
* CREATE TABLE all_transactions AS

SELECT * FROM march_transactions

OUTER JOIN SELECT * FROM april_transactions;
* CREATE TABLE all_transactions AS

SELECT * FROM march_transactions

INTERSECT SELECT * from april_transactions;
* CREATE TABLE all_transactions AS

SELECT * FROM march_transactions

MERGE SELECT * FROM april_transactions;
Explanation

To create a new table all_transactions that contains all records from march_transactions and april_transactions without duplicate records, you should use the UNION operator, as shown in option B. This operator combines the result sets of the two tables while automatically removing duplicate records.

NEW QUESTION 37

In order for Structured Streaming to reliably track the exact progress of the processing so that it can handle any kind of failure by restarting and/or reprocessing, which of the following two approaches is used by Spark to record the offset range of the data being processed in each trigger?
* Checkpointing and Write-ahead Logs
* Structured Streaming cannot record the offset range of the data being processed in each trigger.
* Replayable Sources and Idempotent Sinks
* Write-ahead Logs and Idempotent Sinks
* Checkpointing and Idempotent Sinks

Structured Streaming uses checkpointing and write-ahead logs to record the offset range of the data being processed in each trigger. This ensures that the engine can reliably track the exact progress of the processing and handle any kind of failure by restarting and/or reprocessing. Checkpointing is the mechanism of saving the state of a streaming query to fault-tolerant storage (such as HDFS) so that it can be recovered after a failure. Write-ahead logs are files that record the offset range of the data being processed in each trigger and are written to the checkpoint location before the processing starts. These logs are used to recover the query state and resume processing from the last processed offset range in case of a failure. Reference: Structured Streaming Programming Guide, Fault Tolerance Semantics

NEW QUESTION 38

A data engineer is attempting to drop a Spark SQL table my_table. The data engineer wants to delete all table metadata and data.

They run the following command:

DROP TABLE IF EXISTS my_table

While the object no longer appears when they run SHOW TABLES, the data files still exist.

Which of the following describes why the data files still exist and the metadata files were deleted?
* The table’s data was larger than 10 GB
* The table’s data was smaller than 10 GB
* The table was external
* The table did not have a location
* The table was managed

NEW QUESTION 39

A data engineer has realized that the data files associated with a Delta table are incredibly small. They want to compact the small files to form larger files to improve performance.

Which of the following keywords can be used to compact the small files?
* REDUCE
* OPTIMIZE
* COMPACTION
* REPARTITION
* VACUUM

Explanation

OPTIMIZE can be used to club small files into 1 and improve performance.

NEW QUESTION 40

A data engineer needs to create a table in Databricks using data from their organization’s existing SQLite database.

They run the following command:

```
CREATE TABLE jdbc_customer360
USING _____
OPTIONS (
  url "jdbc:sqlite:/customers.db",
  dbtable "customer360"
)
```


Which of the following lines of code fills in the above blank to successfully complete the task?
* org.apache.spark.sql.jdbc
* autoloader
* DELTA
* sqlite
* org.apache.spark.sql.sqlite

CREATE TABLE new_employees_table

USING JDBC

OPTIONS (

url “<jdbc_url>”,

dbtable “<table_name>”,

user ‘<username>’,

password ‘<password>’

) AS

SELECT * FROM employees_table_vw

https://docs.databricks.com/external-data/jdbc.html#language-sql

NEW QUESTION 41

Which of the following SQL keywords can be used to convert a table from a long format to a wide format?
* PIVOT
* CONVERT
* WHERE
* TRANSFORM
* SUM

The SQL keyword that can be used to convert a table from a long format to a wide format is PIVOT. The PIVOT clause is used to rotate the rows of a table into columns of a new table1. The PIVOT clause can aggregate the values of a column based on the distinct values of another column, and use those values as the column names of the new table1. The PIVOT clause can be useful for transforming data from a long format, where each row represents an observation with multiple attributes, to a wide format, where each row represents an observation with a single attribute and multiple values2. For example, the PIVOT clause can be used to convert a table that contains the sales of different products by different regions into a table that contains the sales of each product by each region as separate columns1.

The other options are not suitable for converting a table from a long format to a wide format. CONVERT is a function that can be used to change the data type of an expression3. WHERE is a clause that can be used to filter the rows of a table based on a condition4. TRANSFORM is a keyword that can be used to apply a user-defined function to a group of rows in a table5. SUM is a function that can be used to calculate the total of a numeric column.

References:

* 1: PIVOT | Databricks on AWS

* 2: Reshaping Data – Long vs Wide Format | Databricks on AWS

* 3: CONVERT | Databricks on AWS

* 4: WHERE | Databricks on AWS

* 5: TRANSFORM | Databricks on AWS

* : [SUM | Databricks on AWS]

NEW QUESTION 42

A data engineer runs a statement every day to copy the previous day’s sales into the table transactions. Each day’s sales are in their own file in the location “/transactions/raw”.

Today, the data engineer runs the following command to complete this task:

```
COPY INTO transactions
FROM "/transactions/raw"
FILEFORMAT = PARQUET;
```

After running the command today, the data engineer notices that the number of records in table transactions has not changed.

Which of the following describes why the statement might not have copied any new records into the table?
* The format of the files to be copied were not included with the FORMAT_OPTIONS keyword.
* The names of the files to be copied were not included with the FILES keyword.
* The previous day’s file has already been copied into the table.
* The PARQUET file format does not support COPY INTO.
* The COPY INTO statement requires the table to be refreshed to view the copied rows.

The COPY INTO statement is an idempotent operation, which means that it will skip any files that have already been loaded into the target table1. This ensures that the data is not duplicated or corrupted by multiple attempts to load the same file. Therefore, if the

data engineer runs the same command every day without specifying the names of the files to be copied with the FILES keyword or a glob pattern with the PATTERN keyword, the statement will only copy the first file that matches the source location and ignore the rest. To avoid this problem, the data engineer should either use the FILES or PATTERN keywords to filter the files to be copied based on the date or some other criteria, or delete the files from the source location after they are copied into the table2. References: 1: COPY INTO | Databricks on AWS 2: Get started using COPY INTO to load data | Databricks on AWS

NEW QUESTION 43

A data engineer wants to create a relational object by pulling data from two tables. The relational object does not need to be used by other data engineers in other sessions. In order to save on storage costs, the data engineer wants to avoid copying and storing physical data.

Which of the following relational objects should the data engineer create?
* Spark SQL Table
* View
* Database
* Temporary view
* Delta Table

A temporary view is a relational object that is defined in the metastore and points to an existing DataFrame. It does not copy or store any physical data, but only saves the query that defines the view. The lifetime of a temporary view is tied to the SparkSession that was used to create it, so it does not persist across different sessions or applications. A temporary view is useful for accessing the same data multiple times within the same notebook or session, without incurring additional storage costs. The other options are either materialized (A, E), persistent (B, C), or not relational objects. References: Databricks Documentation – Temporary View, Databricks Community – How do temp views actually work?, Databricks Community – What’s the difference between a Global view and a Temp view?, Big Data Programmers – Temporary View in Databricks.

NEW QUESTION 44

A Delta Live Table pipeline includes two datasets defined using STREAMING LIVE TABLE. Three datasets are defined against Delta Lake table sources using LIVE TABLE.

The table is configured to run in Production mode using the Continuous Pipeline Mode.

Assuming previously unprocessed data exists and all definitions are valid, what is the expected outcome after clicking Start to update the pipeline?
* All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will persist to allow for additional testing.
* All datasets will be updated once and the pipeline will persist without any processing. The compute resources will persist but go unused.
* All datasets will be updated at set intervals until the pipeline is shut down. The compute resources will be deployed for the update and terminated when the pipeline is stopped.
* All datasets will be updated once and the pipeline will shut down. The compute resources will be terminated.
* All datasets will be updated once and the pipeline will shut down. The compute resources will persist to allow for additional testing.

Explanation

In a Delta Live Table pipeline running in Continuous Pipeline Mode, when you click Start to update the pipeline, the following outcome is expected: All datasets defined using STREAMING LIVE TABLE and LIVE TABLE against Delta Lake table sources will be updated at set intervals. The compute resources will be deployed for the update process and will be active during the execution of the pipeline. The compute resources will be terminated when the pipeline is stopped or shut down. This mode allows

for continuous and periodic updates to the datasets as new data arrives or changes in the underlying Delta Lake tables occur. The compute resources are provisioned and utilized during the update intervals to process the data and perform the necessary operations.

NEW QUESTION 45

Which of the following data workloads will utilize a Gold table as its source?
* A job that enriches data by parsing its timestamps into a human-readable format
* A job that aggregates uncleaned data to create standard summary statistics
* A job that cleans data by removing malformatted records
* A job that queries aggregated data designed to feed into a dashboard
* A job that ingests raw data from a streaming source into the Lakehouse

A Gold table is a table that contains highly refined and aggregated data that powers analytics, machine learning, and production applications. It represents data that has been transformed into knowledge, rather than just information. A Gold table is typically the final output of a medallion lakehouse architecture, where data flows from Bronze to Silver to Gold tables, with each layer improving the structure and quality of data. A job that queries aggregated data designed to feed into a dashboard is an example of a data workload that will utilize a Gold table as its source, as it requires data that is ready for consumption and analysis. The other options are either data workloads that will use a Bronze or Silver table as their source, or data workloads that will produce a Gold table as their output. References: Databricks Documentation – What is the medallion lakehouse architecture?, Databricks Documentation – What is a Medallion Architecture?, K21Academy – Delta Lake Architecture & Azure Databricks Workspace.

NEW QUESTION 46

Which of the following describes when to use the CREATE STREAMING LIVE TABLE (formerly CREATE INCREMENTAL LIVE TABLE) syntax over the CREATE LIVE TABLE syntax when creating Delta Live Tables (DLT) tables using SQL?
* CREATE STREAMING LIVE TABLE should be used when the subsequent step in the DLT pipeline is static.
* CREATE STREAMING LIVE TABLE should be used when data needs to be processed incrementally.
* CREATE STREAMING LIVE TABLE is redundant for DLT and it does not need to be used.
* CREATE STREAMING LIVE TABLE should be used when data needs to be processed through complicated aggregations.
* CREATE STREAMING LIVE TABLE should be used when the previous step in the DLT pipeline is static.

NEW QUESTION 47

A data engineer needs to determine whether to use the built-in Databricks Notebooks versioning or version their project using Databricks Repos.

Which of the following is an advantage of using Databricks Repos over the Databricks Notebooks versioning?
* Databricks Repos automatically saves development progress
* Databricks Repos supports the use of multiple branches
* Databricks Repos allows users to revert to previous versions of a notebook
* Databricks Repos provides the ability to comment on specific changes
* Databricks Repos is wholly housed within the Databricks Lakehouse Platform

Databricks Repos is a visual Git client and API in Databricks that supports common Git operations such as cloning, committing, pushing, pulling, and branch management. Databricks Notebooks versioning is a legacy feature that allows users to link notebooks to GitHub repositories and perform basic Git operations. However, Databricks Notebooks versioning does not support the use of multiple branches for development work, which is an advantage of using Databricks Repos. With Databricks Repos, users can create and manage branches for different features, experiments, or bug fixes, and merge, rebase, or resolve conflicts between them. Databricks recommends using a separate branch for each notebook and following data science and engineering code development best practices using Git for version control, collaboration, and CI/CD. References: Git integration with Databricks Repos – Azure Databricks | Microsoft Learn, Git version control for notebooks (legacy) | Databricks on AWS, Databricks Repos Is Now

Generally Available – New ‘Files’ Feature in …, Databricks Repos – What it is and how we can use it | Adatis.

NEW QUESTION 48

A data engineer only wants to execute the final block of a Python program if the Python variable day_of_week is equal to 1 and the Python variable review_period is True.

Which of the following control flow statements should the data engineer use to begin this conditionally executed code block?
* if day_of_week = 1 and review_period:
* if day_of_week = 1 and review_period = “True”:
* if day_of_week == 1 and review_period == “True”:
* if day_of_week == 1 and review_period:
* if day_of_week = 1 & review_period: = “True”:
Explanation

This statement will check if the variable day_of_week is equal to 1 and if the variable review_period evaluates to a truthy value. The use of the double equal sign (==) in the comparison of day_of_week is important, as a single equal sign (=) would be used to assign a value to the variable instead of checking its value. The use of a single ampersand (&) instead of the keyword and is not valid syntax in Python. The use of quotes around True in options B and C will result in a string comparison, which will not evaluate to True even if the value of review_period is True.

NEW QUESTION 49

A data engineer needs to create a table in Databricks using data from their organization’s existing SQLite database.

They run the following command:

```
CREATE TABLE jdbc_customer360
USING _____
OPTIONS (
  url "jdbc:sqlite:/customers.db",
  dbtable "customer360"
)
```


Which of the following lines of code fills in the above blank to successfully complete the task?
* org.apache.spark.sql.jdbc
* autoloader
* DELTA
* sqlite
* org.apache.spark.sql.sqlite
1: In the given command, a data engineer is trying to create a table in Databricks using data from an SQLite database. The correct option to fill in the blank is “sqlite” because it specifies the type of database being connected to in a JDBC connection string. The USING clause should be followed by the format of the data, and since we are connecting to an SQLite database, “sqlite” would be appropriate here. Reference:

Create a table using JDBC

JDBC connection string

SQLite JDBC driver

NEW QUESTION 50

Which of the following describes the type of workloads that are always compatible with Auto Loader?
* Dashboard workloads
* Streaming workloads
* Machine learning workloads
* Serverless workloads
* Batch workloads

Auto Loader is a Structured Streaming source that incrementally and efficiently processes new data files as they arrive in cloud storage. It supports both Python and SQL in Delta Live Tables, which are ideal for building streaming data pipelines. Auto Loader can handle near real-time ingestion of millions of files per hour and provide exactly-once guarantees when writing data into Delta Lake. Auto Loader is not designed for dashboard, machine learning, serverless, or batch workloads, which have different requirements and characteristics. References: What is Auto Loader?, Delta Live Tables

NEW QUESTION 51

A data engineer is attempting to drop a Spark SQL table my_table. The data engineer wants to delete all table metadata and data.

They run the following command:

DROP TABLE IF EXISTS my_table

While the object no longer appears when they run SHOW TABLES, the data files still exist.

Which of the following describes why the data files still exist and the metadata files were deleted?
* The table’s data was larger than 10 GB
* The table’s data was smaller than 10 GB
* The table was external
* The table did not have a location
* The table was managed

An external table is a table that is defined in the metastore and points to an existing location in the storage system. When you drop an external table, only the metadata is deleted from the metastore, but the data files are not deleted from the storage system. This is because external tables are meant to be shared by multiple applications and users, and dropping them should not affect the data availability. On the other hand, a managed table is a table that is defined in the metastore and also managed by the metastore. When you drop a managed table, both the metadata and the data files are deleted from the metastore and the storage system, respectively. This is because managed tables are meant to be exclusive to the application or user that created them, and dropping them should free up the storage space. Therefore, the correct answer is C, because the table was external and only the metadata was deleted when the table was dropped. References: Databricks Documentation – Managed and External Tables, Databricks Documentation – Drop Table

Databricks-Certified-Data-Engineer-Associate Dumps Updated Practice Test and 102 unique questions:
https://www.validbraindumps.com/Databricks-Certified-Data-Engineer-Associate-exam-prep.html]