Free Questions for Databricks-Certified-Professional-Data-Engineer

Shared by Ball on 21-03-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

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?

Options:

A- configure
B- fs
C- jobs
D- libraries
E- workspace

Answer:

D

Explanation:

The libraries command group allows you to install, uninstall, and list libraries on Databricks clusters. You can use the libraries install command to install a custom Python Wheel on a cluster by specifying the --whl option and the path to the wheel file. For example, you can use the following command to install a custom Python Wheel named mylib-0.1-py3-none-any.whl on a cluster with the id 1234-567890-abcde123:

databricks libraries install --cluster-id 1234-567890-abcde123 --whl dbfs:/mnt/mylib/mylib-0.1-py3-none-any.whl

This will upload the custom Python Wheel to the cluster and make it available for use with a production job. You can also use the libraries uninstall command to uninstall a library from a cluster, and the libraries list command to list the libraries installed on a cluster.

Libraries CLI (legacy): https://docs.databricks.com/en/archive/dev-tools/cli/libraries-cli.html

Library operations: https://docs.databricks.com/en/dev-tools/cli/commands.html#library-operations

Install or update the Databricks CLI: https://docs.databricks.com/en/dev-tools/cli/install.html

Question 2

Question Type: MultipleChoice

The data engineer is using Spark's MEMORY_ONLY storage level.

Which indicators should the data engineer look for in the spark UI's Storage tab to signal that a cached table is not performing optimally?

Options:

A- Size on Disk is> 0
B- The number of Cached Partitions> the number of Spark Partitions
C- The RDD Block Name included the '' annotation signaling failure to cache
D- On Heap Memory Usage is within 75% of off Heap Memory usage

Answer:

C

Explanation:

In the Spark UI's Storage tab, an indicator that a cached table is not performing optimally would be the presence of the _disk annotation in the RDD Block Name. This annotation indicates that some partitions of the cached data have been spilled to disk because there wasn't enough memory to hold them. This is suboptimal because accessing data from disk is much slower than from memory. The goal of caching is to keep data in memory for fast access, and a spill to disk means that this goal is not fully achieved.

Question 3

Question Type: MultipleChoice

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?

Options:

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.

Answer:

A

Explanation:

The reason why the junior data engineer cannot transfer ''Owner'' privileges to the ''DevOps'' group is that Databricks jobs must have exactly one owner, and the owner must be an individual user, not a group. A job cannot have more than one owner, and a job cannot have a group as an owner. The owner of a job is the user who created the job, or the user who was assigned the ownership by another user. The owner of a job has the highest level of permission on the job, and can grant or revoke permissions to other users or groups. However, the owner cannot transfer the ownership to a group, only to another user. Therefore, the junior data engineer's attempt to transfer ''Owner'' privileges to the ''DevOps'' group is not possible.Reference:

Jobs access control: https://docs.databricks.com/security/access-control/table-acls/index.html

Job permissions: https://docs.databricks.com/security/access-control/table-acls/privileges.html#job-permissions

Question 4

Question Type: MultipleChoice

A user wants to use DLT expectations to validate that a derived table report contains all records from the source, included in the table validation_copy.

The user attempts and fails to accomplish this by adding an expectation to the report table definition.

Which approach would allow using DLT expectations to validate all expected records are present in this table?

Options:

A- Define a SQL UDF that performs a left outer join on two tables, and check if this returns null values for report key values in a DLT expectation for the report table.
B- Define a function that performs a left outer join on validation_copy and report and report, and check against the result in a DLT expectation for the report table
C- Define a temporary table that perform a left outer join on validation_copy and report, and define an expectation that no report key values are null
D- Define a view that performs a left outer join on validation_copy and report, and reference this view in DLT expectations for the report table

Answer:

D

Explanation:

To validate that all records from the source are included in the derived table, creating a view that performs a left outer join between the validation_copy table and the report table is effective. The view can highlight any discrepancies, such as null values in the report table's key columns, indicating missing records. This view can then be referenced in DLT (Delta Live Tables) expectations for the report table to ensure data integrity. This approach allows for a comprehensive comparison between the source and the derived table.

Databricks Documentation on Delta Live Tables and Expectations: Delta Live Tables Expectations

Question 5

Question Type: MultipleChoice

The data engineer team has been tasked with configured connections to an external database that does not have a supported native connector with Databricks. The external database already has data security configured by group membership. These groups map directly to user group already created in Databricks that represent various teams within the company.

A new login credential has been created for each group in the external database. The Databricks Utilities Secrets module will be used to make these credentials available to Databricks users.

Assuming that all the credentials are configured correctly on the external database and group membership is properly configured on Databricks, which statement describes how teams can be granted the minimum necessary access to using these credentials?

Options:

A- ''Read'' permissions should be set on a secret key mapped to those credentials that will be used by a given team.
B- No additional configuration is necessary as long as all users are configured as administrators in the workspace where secrets have been added.
C- ''Read'' permissions should be set on a secret scope containing only those credentials that will be used by a given team.
D- ''Manage'' permission should be set on a secret scope containing only those credentials that will be used by a given team.

Answer:

C

Explanation:

In Databricks, using the Secrets module allows for secure management of sensitive information such as database credentials. Granting 'Read' permissions on a secret key that maps to database credentials for a specific team ensures that only members of that team can access these credentials. This approach aligns with the principle of least privilege, granting users the minimum level of access required to perform their jobs, thus enhancing security.

Databricks Documentation on Secret Management: Secrets

Question 6

Question Type: MultipleChoice

Which is a key benefit of an end-to-end test?

Options:

A- It closely simulates real world usage of your application.
B- It pinpoint errors in the building blocks of your application.
C- It provides testing coverage for all code paths and branches.
D- It makes it easier to automate your test suite

Answer:

A

Explanation:

End-to-end testing is a methodology used to test whether the flow of an application, from start to finish, behaves as expected. The key benefit of an end-to-end test is that it closely simulates real-world, user behavior, ensuring that the system as a whole operates correctly.

Software Testing: End-to-End Testing

Question 7

Question Type: MultipleChoice

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 approval for the sales org.

Which of the following solutions addresses the situation while emphasizing simplicity?

Options:

A- Create a view on the marketing table selecting only these fields approved for the sales team alias the names of any fields that should be standardized to the sales naming conventions.
B- Use a CTAS statement to create a derivative table from the marketing table configure a production jon to propagation changes.
C- Add a parallel table write to the current production pipeline, updating a new sales table that varies as required from marketing table.
D- 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.

Answer:

A

Explanation:

Creating a view is a straightforward solution that can address the need for field name standardization and selective field sharing between departments. A view allows for presenting a transformed version of the underlying data without duplicating it. In this scenario, the view would only include the approved fields for the sales team and rename any fields as per their naming conventions.

Databricks documentation on using SQL views in Delta Lake:
https://docs.databricks.com/delta/quick-start.html#sql-views

Question 8

Question Type: MultipleChoice

The DevOps team has configured a production workload as a collection of notebooks scheduled to run daily using the Jobs Ul. 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?

Options:

A- Can manage
B- Can edit
C- Can run
D- Can Read

Answer:

D

Explanation:

Granting a user 'Can Read' permissions on a notebook within Databricks allows them to view the notebook's content without the ability to execute or edit it. This level of permission ensures that the new team member can review the production logic for learning or auditing purposes without the risk of altering the notebook's code or affecting production data and workflows. This approach aligns with best practices for maintaining security and integrity in production environments, where strict access controls are essential to prevent unintended modifications. Reference: Databricks documentation on access control and permissions for notebooks within the workspace (https://docs.databricks.com/security/access-control/workspace-acl.html).

Question 9

Question Type: MultipleChoice

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?

Options:

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.

Answer:

C

Explanation:

To deduplicate data against previously processed records as it is inserted into a Delta table, you can use the merge operation with an insert-only clause. This allows you to insert new records that do not match any existing records based on a unique key, while ignoring duplicate records that match existing records. For example, you can use the following syntax:

MERGE INTO target_table USING source_table ON target_table.unique_key = source_table.unique_key WHEN NOT MATCHED THEN INSERT *

This will insert only the records from the source table that have a unique key that is not present in the target table, and skip the records that have a matching key. This way, you can avoid inserting duplicate records into the Delta table.

https://docs.databricks.com/delta/delta-update.html#upsert-into-a-table-using-merge

https://docs.databricks.com/delta/delta-update.html#insert-only-merge

Question 10

Question Type: MultipleChoice

A team of data engineer are adding tables to a DLT pipeline that contain repetitive expectations for many of the same data quality checks.

One member of the team suggests reusing these data quality rules across all tables defined for this pipeline.

What approach would allow them to do this?

Options:

A- Maintain data quality rules in a Delta table outside of this pipeline's target schema, providing the schema name as a pipeline parameter.
B- Use global Python variables to make expectations visible across DLT notebooks included in the same pipeline.
C- Add data quality constraints to tables in this pipeline using an external job with access to pipeline configuration files.
D- Maintain data quality rules in a separate Databricks notebook that each DLT notebook of file.

Answer:

A

Explanation:

Maintaining data quality rules in a centralized Delta table allows for the reuse of these rules across multiple DLT (Delta Live Tables) pipelines. By storing these rules outside the pipeline's target schema and referencing the schema name as a pipeline parameter, the team can apply the same set of data quality checks to different tables within the pipeline. This approach ensures consistency in data quality validations and reduces redundancy in code by not having to replicate the same rules in each DLT notebook or file.

Databricks Documentation on Delta Live Tables: Delta Live Tables Guide