Free Questions for Amazon-DEA-C01

Shared by Townsend on 23-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: **MultipleChoice**

A data engineer needs to use Amazon Neptune to develop graph applications.

Which programming languages should the engineer use to develop the graph applications? (Select TWO.)

Options:

A- Gremlin

B- SQL

C- ANSI SQL

D- SPARQL

E- Spark SQL

Answer:

A, D

Explanation:

Amazon Neptune supports graph applications using Gremlin and SPARQL as query languages. Neptune is a fully managed graph database service that supports both property graph and RDF graph models.

Option A: Gremlin Gremlin is a query language for property graph databases, which is supported by Amazon Neptune. It allows the traversal and manipulation of graph data in the property graph model.

Option D: SPARQL SPARQL is a query language for querying RDF graph data in Neptune. It is used to query, manipulate, and retrieve information stored in RDF format.

Other options:

SQL (Option B) and ANSI SQL (Option C) are traditional relational database query languages and are not used for graph databases.

Spark SQL (Option E) is related to Apache Spark for big data processing, not for querying graph databases.

Amazon Neptune Documentation

Gremlin Documentation

SPARQL Documentation

Question 2

Question Type: **MultipleChoice**

An ecommerce company wants to use AWS to migrate data pipelines from an on-premises environment into the AWS Cloud. The company currently uses a third-party too in the on-premises environment to orchestrate data ingestion processes.

The company wants a migration solution that does not require the company to manage servers. The solution must be able to orchestrate Python and Bash scripts. The solution must not require the company to refactor any code.

Which solution will meet these requirements with the LEAST operational overhead?

Options:

A- AWS Lambda

B- Amazon Managed Workflows for Apache Airflow (Amazon MWAA)

C- AWS Step Functions

D- AWS Glue

Answer:

B

Explanation:

The ecommerce company wants to migrate its data pipelines into the AWS Cloud without managing servers, and the solution must orchestrate Python and Bash scripts without refactoring code. Amazon Managed Workflows for Apache Airflow (Amazon MWAA) is the most suitable solution for this scenario.

Option B: Amazon Managed Workflows for Apache Airflow (Amazon MWAA) MWAA is a managed orchestration service that supports Python and Bash scripts via Directed Acyclic Graphs (DAGs) for workflows. It is a serverless, managed version of Apache Airflow, which is commonly used for orchestrating complex data workflows, making it an ideal choice for migrating existing pipelines without refactoring. It supports Python, Bash, and other scripting languages, and the company would not need to manage the underlying infrastructure.

Other options:

AWS Lambda (Option A) is more suited for event-driven workflows but would require breaking down the pipeline into individual Lambda functions, which may require refactoring.

AWS Step Functions (Option C) is good for orchestration but lacks native support for Python and Bash without using Lambda functions, and it may require code changes.

AWS Glue (Option D) is an ETL service primarily for data transformation and not suitable for orchestrating general scripts without modification.

Amazon Managed Workflows for Apache Airflow (MWAA) Documentation

Question 3

Question Type: **MultipleChoice**

A data engineer uses Amazon Managed Workflows for Apache Airflow (Amazon MWAA) to run data pipelines in an AWS account. A workflow recently failed to run. The data engineer needs to use Apache Airflow logs to diagnose the failure of the workflow. Which log type should the data engineer use to diagnose the cause of the failure?

Options:

A- YourEnvironmentName-WebServer

B- YourEnvironmentName-Scheduler

C- YourEnvironmentName-DAGProcessing

D- YourEnvironmentName-Task

Answer:

D

Explanation:

In Amazon Managed Workflows for Apache Airflow (MWAA), the type of log that is most useful for diagnosing workflow (DAG) failures is the Task logs. These logs provide detailed information on the execution of each task within the DAG, including error messages, exceptions, and other critical details necessary for diagnosing failures.

Option D: YourEnvironmentName-Task Task logs capture the output from the execution of each task within a workflow (DAG), which is crucial for understanding what went wrong when a DAG fails. These logs contain detailed execution information, including errors and stack traces, making them the best source for debugging.

Other options (WebServer, Scheduler, and DAGProcessing logs) provide general environment-level logs or logs related to scheduling and DAG parsing, but they do not provide the granular task-level execution details needed for diagnosing workflow failures.

Amazon MWAA Logging and Monitoring

Apache Airflow Task Logs

Question 4

Question Type: MultipleChoice

A company wants to migrate data from an Amazon RDS for PostgreSQL DB instance in the eu-east-1 Region of an AWS account named Account_

Options:

A- The company will migrate the data to an Amazon Redshift cluster in the eu-west-1 Region of an AWS account named Account_B. Which solution will give AWS Database Migration Service (AWS DMS) the ability to replicate data between two data stores?

A- Set up an AWS DMS replication instance in Account_B in eu-west-1.

B- Set up an AWS DMS replication instance in Account_B in eu-east-1.

C- Set up an AWS DMS replication instance in a new AWS account in eu-west-1

D- Set up an AWS DMS replication instance in Account_A in eu-east-1.

Answer:

A, A

Explanation:

To migrate data from an Amazon RDS for PostgreSQL DB instance in the eu-east-1 Region (Account_A) to an Amazon Redshift cluster in the eu-west-1 Region (Account_B), AWS DMS needs a replication instance located in the target region (in this case, eu-west-1) to facilitate the data transfer between regions.

Option A: Set up an AWS DMS replication instance in Account_B in eu-west-1. Placing the DMS replication instance in the target account and region (Account_B in eu-west-1) is the most efficient solution. The replication instance can connect to the source RDS PostgreSQL in eu-east-1 and migrate the data to the Redshift cluster in eu-west-1. This setup ensures data is replicated across AWS accounts and regions.

Options B, C, and D place the replication instance in either the wrong account or region, which increases complexity without adding any benefit.

AWS Database Migration Service (DMS) Documentation

Cross-Region and Cross-Account Replication

Question 5

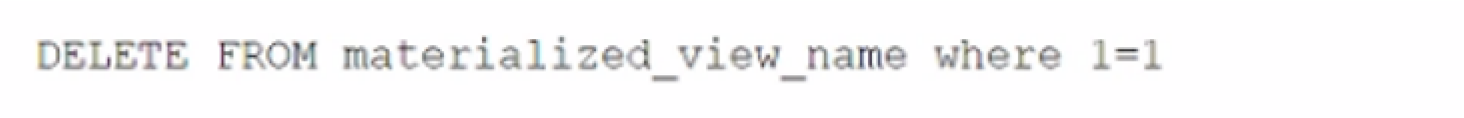
Question Type: MultipleChoice

A data engineer maintains a materialized view that is based on an Amazon Redshift database. The view has a column named load_date that stores the date when each row was loaded.

The data engineer needs to reclaim database storage space by deleting all the rows from the materialized view.

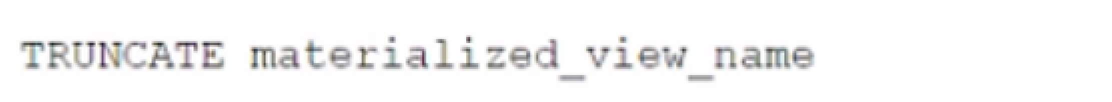
Which command will reclaim the MOST database storage space?

A.



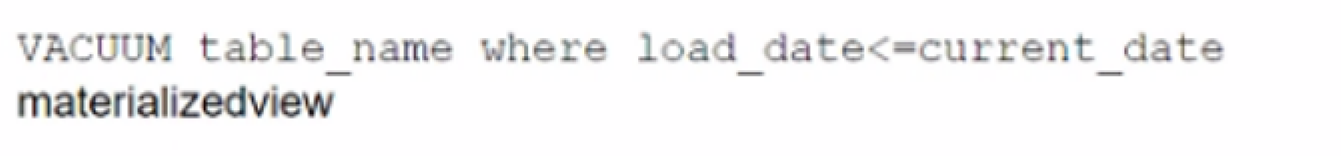
```
DELETE FROM materialized_view_name where 1=1
```

B.



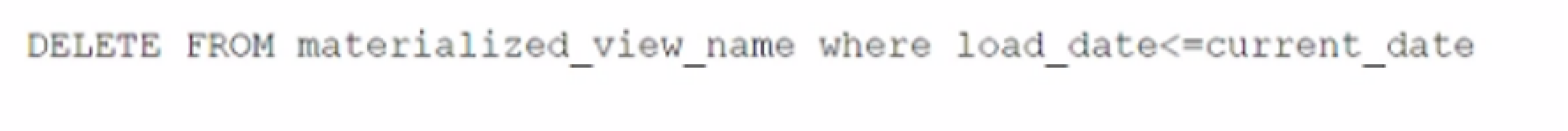
```
TRUNCATE materialized_view_name
```

C.



```
VACUUM table_name where load_date<=current_date
materializedview
```

D.



```
DELETE FROM materialized_view_name where load_date<=current_date
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A

Explanation:

To reclaim the most storage space from a materialized view in Amazon Redshift, you should use a DELETE operation that removes all rows from the view. The most efficient way to remove all rows is to use a condition that always evaluates to true, such as 1=1. This will delete all rows without needing to evaluate each row individually based on specific column values like load_date.

Option A: DELETE FROM materialized_view_name WHERE 1=1; This statement will delete all rows in the materialized view and free up the space. Since materialized views in Redshift store precomputed data, performing a DELETE operation will remove all stored rows.

Other options either involve inappropriate SQL statements (e.g., VACUUM in option C is used for reclaiming storage space in tables, not materialized views), or they don't remove data effectively in the context of a materialized view (e.g., TRUNCATE cannot be used directly on a materialized view).

Amazon Redshift Materialized Views Documentation

Deleting Data from Redshift

Question 6

Question Type: **MultipleChoice**

A company plans to use Amazon Kinesis Data Firehose to store data in Amazon S3. The source data consists of 2 MB csv files. The company must convert the .csv files to JSON format. The company must store the files in Apache Parquet format.

Which solution will meet these requirements with the LEAST development effort?

Options:

A- Use Kinesis Data Firehose to convert the csv files to JSON. Use an AWS Lambda function to store the files in Parquet format.

B- Use Kinesis Data Firehose to convert the csv files to JSON and to store the files in Parquet format.

C- Use Kinesis Data Firehose to invoke an AWS Lambda function that transforms the .csv files to JSON and stores the files in Parquet format.

D- Use Kinesis Data Firehose to invoke an AWS Lambda function that transforms the .csv files to JSON. Use Kinesis Data Firehose to store the files in Parquet format.

Answer:

B

Explanation:

The company wants to use Amazon Kinesis Data Firehose to transform CSV files into JSON format and store the files in Apache Parquet format with the least development effort.

Option B: Use Kinesis Data Firehose to convert the CSV files to JSON and to store the files in Parquet format. Kinesis Data Firehose supports data format conversion natively, including converting incoming CSV data to JSON format and storing the resulting files in Parquet format in Amazon S3. This solution requires the least development effort because it uses built-in transformation features of Kinesis Data Firehose.

Other options (A, C, D) involve invoking AWS Lambda functions, which would introduce additional complexity and development effort compared to Kinesis Data Firehose's native format conversion capabilities.

Amazon Kinesis Data Firehose Documentation

Question 7

Question Type: MultipleChoice

A company has a data warehouse that contains a table that is named Sales. The company stores the table in Amazon Redshift The table includes a column that is named city_name. The company wants to query the table to find all rows that have a city_name that starts with "San" or "El."

Which SQL query will meet this requirement?

Options:

A- Select * from Sales where city_name - '$(San|El)';

B- Select * from Sales where city_name -, ^(San|El) *';

C- Select * from Sales where city_name - '$(San&El)';

D- Select * from Sales where city_name -, ^(San&El)';

Answer:

B

Explanation:

To query the Sales table in Amazon Redshift for city names that start with 'San' or 'El,' the appropriate query uses a regular expression (regex) pattern to match city names that begin with those prefixes.

Option B: Select * from Sales where city_name ~ '^(San|El)'; In Amazon Redshift, the ~ operator is used to perform pattern matching using regular expressions. The ^(San|El) pattern matches city names that start with 'San' or 'El.' This is the correct SQL syntax for this use case.

Other options (A, C, D) contain incorrect syntax or incorrect use of special characters, making them invalid queries.

Amazon Redshift Pattern Matching Documentation

Question 8

Question Type: MultipleChoice

A company is using Amazon Redshift to build a data warehouse solution. The company is loading hundreds of tiles into a tact table that is in a Redshift cluster.

The company wants the data warehouse solution to achieve the greatest possible throughput. The solution must use cluster resources optimally when the company loads data into the tact table.

Which solution will meet these requirements?

Options:

A- Use multiple COPY commands to load the data into the Redshift cluster.

B- Use S3DistCp to load multiple files into Hadoop Distributed File System (HDFS). Use an HDFS connector to ingest the data into the Redshift cluster.

C- Use a number of INSERT statements equal to the number of Redshift cluster nodes. Load the data in parallel into each node.

D- Use a single COPY command to load the data into the Redshift cluster.

Answer:

D

Explanation:

To achieve the highest throughput and efficiently use cluster resources while loading data into an Amazon Redshift cluster, the optimal approach is to use a single COPY command that ingests data in parallel.

Option D: Use a single COPY command to load the data into the Redshift cluster. The COPY command is designed to load data from multiple files in parallel into a Redshift table, using all the cluster nodes to optimize the load process. Redshift is optimized for parallel processing, and a single COPY command can load multiple files at once, maximizing throughput.

Options A, B, and C either involve unnecessary complexity or inefficient approaches, such as using multiple COPY commands or INSERT statements, which are not optimized for bulk loading.

Amazon Redshift COPY Command Documentation