Accelerate Your Certification with Microsoft DP-800 Practice Questions

Last chance to prepare smart! Get your hands on free Microsoft Developing AI-Enabled Database Solutions exam PDF questions. Study real DP-800 dumps with verified answers and fast-track your certification success with PrepBolt DP-800 exam pdf questions and answers.

Thank you for Downloading DP-800 exam PDF Demo

https://prepbolt.com/DP-800.html

QUESTIONS & ANSWERS

DEMO VERSION

(LIMITED CONTENT)

Question 1

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

DP-800 Mix Questions IN THIS CASE STUDY

You have a SQL database in Microsoft Fabric that contains a table named dbo.Orders, dbo.Orders has a clustered index, contains three years of data, and is partitioned by a column named OrderDate by month.

You need to remove all the rows for the oldest month. The solution must minimize the impact on other queries that access the data in dbo.orders.

Solution; Identify the partition scheme (or the oldest month, and then run the following Transact-SQL statement.

ALTER TABLE dbo.Orders

DROP PARTITION SCHEME (partition_scheme_name);

Does this meet the goal?

Options:

A- Yes

B- No

Answer:

B

Explanation:

This also does not meet the goal. DROP PARTITION SCHEME removes the partition scheme object from the database; it is not the command used to remove just the rows for the oldest month from a partitioned table. Microsoft's DROP PARTITION SCHEME documentation is explicit that the statement removes the partition scheme itself.

For removing only the oldest month's rows with minimal impact, Microsoft points to partition-level maintenance operations such as truncating a single partition on a partitioned table. That targets only the needed data subset and is more efficient for retention workloads.

Question 2

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

DP-800 Mix Questions IN THIS CASE STUDY

You have a SQL database in Microsoft Fabric that contains a table named dbo.Orders, dbo.Orders has a clustered index, contains three years of data, and is partitioned by a column named OrderDate by month.

You need to remove all the rows for the oldest month. The solution must minimize the impact on other queries that access the data in dbo.orders.

Solution: Run the following Transact-SQL statement.

DELETE FROM dbo.Orders

WHERE OrderDate < DATEADD(nonth, -36, SYSUTCDATETIME());

Does this meet the goal?

Options:

A- Yes
B- No

Answer:

B

Explanation:

This does not meet the goal. A row-by-row DELETE against the oldest month is not the lowest-impact way to purge data from a monthly partitioned table. Microsoft's partitioning guidance specifically says partitioning lets you perform maintenance and retention operations more efficiently by targeting just the relevant partition, including the ability to truncate data in a single partition.

The proposed statement:

DELETE FROM dbo.Orders WHERE OrderDate < DATEADD(month, -36, SYSUTCDATETIME());

would log row deletions and can hold locks longer, creating more overhead for other queries than a partition-level maintenance operation. Since the table is already partitioned by month, the expected low-impact approach is to operate on the oldest partition directly, not issue a broad delete predicate over rows. Microsoft explicitly highlights partition-targeted truncation as a faster, more efficient retention operation than working against the whole table or rowset.

Question 3

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

DP-800 Mix Questions IN THIS CASE STUDY

You have a SQL database in Microsoft Fabric that contains a table named dbo.Orders, dbo.Orders has a clustered index, contains three years of data, and is partitioned by a column named OrderDate by month.

You need to remove all the rows for the oldest month. The solution must minimize the impact on other queries that access the data in dbo.orders.

Solution: Identify the partition number for the oldest month, and then run the following Transact-SQL statement.

TRUNCATE TABIE dbo.Orders

WITH (PARTITIONS (partition number));

Does this meet the goal?

Options:

A- Yes

B- No

Answer:

A

Explanation:

Yes, this meets the goal. Microsoft documents that on a partitioned table, you can use TRUNCATE TABLE ... WITH (PARTITIONS (...)) to remove data from a specific partition, and that this is an efficient maintenance operation that targets only that data subset rather than the whole table. Microsoft's partitioning guidance explicitly lists truncating a single partition as an example of a fast partition-level maintenance or retention operation.

That matches the requirement to remove the oldest month while minimizing impact on other queries. Because the table is already partitioned by month on OrderDate, identifying the partition number for that oldest month and truncating only that partition is the correct low-impact approach, assuming the table and indexes are aligned as required for partition truncation.

Question 4

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

DP-800 Mix Questions IN THIS CASE STUDY

You have an SDK-style SQL database project stored in a Git repository. The project targets an Azure SQL database.

The CI build fails with unresolved reference errors when the project references system objects.

You need to update the SQL database project to ensure that dotnet build validates successfully by including the correct system objects in the database model for Azure SQL Database.

Solution: Add the Microsoft.SqlServer.Dacpacs.Azure.Master NuGet package to the project.

Does this meet the goal?

Options:

A- Yes
B- No

Answer:

A

Explanation:

This does meet the goal. Microsoft documents that SDK-style SQL projects can add the master.dacpac database reference as a package reference, and for Azure SQL Database the correct package is the Azure-specific master DACPAC package. The Azure SQL system DACPACs are available through NuGet, and this is the recommended way to include the right system objects in the database model for dotnet build validation.

So for an SDK-style SQL database project that targets Azure SQL Database, adding

Microsoft.SqlServer.Dacpacs.Azure.Master is the correct fix for unresolved references to system objects.

Question 5

Question Type: MultipleChoice

Case Study: Mix Questions

Mix Questions

DP-800 Mix Questions IN THIS CASE STUDY

You have an SDK-style SQL database project stored in a Git repository. The project targets an Azure SQL database.

The CI build fails with unresolved reference errors when the project ieferences system objects.

You need to update the SQL database project to ensure that dotnet build validates successfully by including the correct system objects in the database model for Azure SQL Database.

Solution: Add the Microsoft.SqlServer.Dacpacs.Mastet NuGet package to the project.

Does this meet the goal?

Options:

A- Yes
B- No

Answer:

B

Explanation:

The package named Microsoft.SqlServer.Dacpacs.Master is the generic master system DACPAC package, but the question requires the correct system objects for Azure SQL Database. Microsoft's system-objects documentation distinguishes platform-specific system references, and for Azure SQL Database the correct package is the Azure-specific master DACPAC, not the generic master package.

So adding Microsoft.SqlServer.Dacpacs.Master does not meet the goal for an Azure SQL Database-targeted SDK-style project. The expected package is the Azure-specific one.