

# Demo Questions

## Microsoft DP-201 Exam

### Designing an Azure Data Solution

Thank you for downloading DP-201 Exam PDF

#### Question #1 Topic 1

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are designing an HDInsight/Hadoop cluster solution that uses Azure Data Lake Gen1 Storage.

The solution requires POSIX permissions and enables diagnostics logging for auditing. You need to recommend solutions that optimize storage.

Proposed Solution: Ensure that files stored are larger than 250MB.

Does the solution meet the goal?

- A. Yes
- B. No

#### Correct Answer: A

Depending on what services and workloads are using the data, a good size to consider for files is 256 MB or greater. If the file sizes cannot be batched when landing in Data

Lake Storage Gen1, you can have a separate compaction job that combines these files into larger ones.

Note: POSIX permissions and auditing in Data Lake Storage Gen1 comes with an overhead that becomes apparent when working with numerous small files. As a best practice, you must batch your data into larger files versus writing thousands or millions of small files to Data Lake Storage Gen1. Avoiding small file sizes can have multiple benefits, such as:

- ☞ Lowering the authentication checks across multiple files
- ☞ Reduced open file connections
- ☞ Faster copying/replication
- ☞ Fewer files to process when updating Data Lake Storage Gen1 POSIX permissions

Reference:

<https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-best-practices>

### **Question #2Topic 1**

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are designing an HDInsight/Hadoop cluster solution that uses Azure Data Lake Gen1 Storage.

The solution requires POSIX permissions and enables diagnostics logging for auditing. You need to recommend solutions that optimize storage.

Proposed Solution: Implement compaction jobs to combine small files into larger files.

Does the solution meet the goal?

- A. Yes
- B. No

### **Correct Answer: A**

Depending on what services and workloads are using the data, a good size to consider for files is 256 MB or greater. If the file sizes cannot be batched when landing in Data Lake Storage Gen1, you can have a separate compaction job that combines these files into larger ones.

Note: POSIX permissions and auditing in Data Lake Storage Gen1 comes with an overhead that becomes apparent when working with numerous small files. As a best practice, you must batch your data into larger files versus writing thousands or millions

of small files to Data Lake Storage Gen1. Avoiding small file sizes can have multiple benefits, such as:

- ⇒ Lowering the authentication checks across multiple files
- ⇒ Reduced open file connections
- ⇒ Faster copying/replication
- ⇒ Fewer files to process when updating Data Lake Storage Gen1 POSIX permissions

Reference:

<https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-best-practices>

### **Question #3 Topic 1**

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are designing an HDInsight/Hadoop cluster solution that uses Azure Data Lake Gen1 Storage.

The solution requires POSIX permissions and enables diagnostics logging for auditing.

You need to recommend solutions that optimize storage.

Proposed Solution: Ensure that files stored are smaller than 250MB.

Does the solution meet the goal?

- A. Yes
- B. No

### **Correct Answer: B**

Ensure that files stored are larger, not smaller than 250MB.

You can have a separate compaction job that combines these files into larger ones.

Note: The file POSIX permissions and auditing in Data Lake Storage Gen1 comes with an overhead that becomes apparent when working with numerous small files. As a best practice, you must batch your data into larger files versus writing thousands or millions of small files to Data Lake Storage Gen1. Avoiding small file sizes can have multiple benefits, such as:

- ⇒ Lowering the authentication checks across multiple files
- ⇒ Reduced open file connections
- ⇒ Faster copying/replication
- ⇒ Fewer files to process when updating Data Lake Storage Gen1 POSIX permissions

Reference:

<https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-best-practices>

### **Question #4 Topic 1**

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are designing an Azure SQL Database that will use elastic pools. You plan to store data about customers in a table. Each record uses a value for CustomerID.

You need to recommend a strategy to partition data based on values in CustomerID.

Proposed Solution: Separate data into customer regions by using vertical partitioning.

Does the solution meet the goal?

- A. Yes
- B. No

### **Correct Answer: B**

Vertical partitioning is used for cross-database queries. Instead we should use Horizontal Partitioning, which also is called charding.

Reference:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-query-overview>

### **Question #5 Topic 1**

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You are designing an Azure SQL Database that will use elastic pools. You plan to store data about customers in a table. Each record uses a value for CustomerID.

You need to recommend a strategy to partition data based on values in CustomerID.

Proposed Solution: Separate data into customer regions by using horizontal partitioning. Does the solution meet the goal?

- A. Yes
- B. No

**Correct Answer: B**

We should use Horizontal Partitioning through Sharding, not divide through regions.

Note: Horizontal Partitioning - Sharding: Data is partitioned horizontally to distribute rows across a scaled out data tier. With this approach, the schema is identical on all participating databases. This approach is also called "sharding". Sharding can be performed and managed using (1) the elastic database tools libraries or (2) self-sharding. An elastic query is used to query or compile reports across many shards.

Reference:

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-query-overview>