



**Free Questions for DP-203 by actualtestdumps**

**Shared by Mendoza on 15-04-2024**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

You have an Azure data factor/ connected to a Git repository that contains the following branches:

\* mam: Collaboration branch

\* abc: Feature branch

\* xyz: Feature branch

You save changes to a pipeline in the xyz branch.

You need to publish the changes to the live service

What should you do first?

### Options:

---

**A-** Push the code to a remote origin.

**B-** Publish the data factory.

**C-** Create a pull request to merge the changes into the abc branch.

**D-** Create a pull request to merge the changes into the main branch.

**Answer:**

---

D

## Question 2

---

**Question Type:** MultipleChoice

---

You have an Azure Synapse Analytics dedicated SQL pool named Pool1.

Pool1 contains two tables named SalesFact\_Staging and SalesFact. Both tables have a matching number of partitions, all of which contain data.

You need to load data from SalesFact\_Staging to SalesFact by switching a partition.

What should you specify when running the alter TABLE statement?

**Options:**

---

**A-** WITH NOCHECK

**B-** WITH (TRUNCATE\_TARGET = ON)

**C-** WITH (TRACK\_COLUMNS\_UPDATED = ON)

**D-** WITH CHECK

**Answer:**

---

B

## Question 3

---

**Question Type:** MultipleChoice

---

You have an Azure data factory that connects to a Microsoft Purview account. The data 'factory is registered in Microsoft Purview.

You update a Data Factory pipeline.

You need to ensure that the updated lineage is available in Microsoft Purview.

What should you do first?

**Options:**

---

**A-** Disconnect the Microsoft Purview account from the data factory.

**B-** Locate the related asset in the Microsoft Purview portal.

**C-** Execute an Azure DevOps build pipeline.

**D-** Execute the pipeline.

**Answer:**

---

D

## Question 4

---

**Question Type: Hotspot**

---

You have an Azure subscription that contains an Azure Synapse Analytics workspace named workspace1. Workspace1 contains a dedicated SQL pool named SQL Pool and an Apache Spark pool named sparkpool. Sparkpool1 contains a DataFrame named pyspark.df.

You need to write the contents of pyspark\_df to a table in SQLPoolM by using a PySpark notebook.

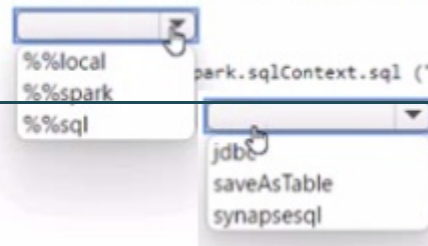
How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

## Answer Area

Answer:

```
pyspark_df.createOrReplaceTempView("pysparkdftemptable")
park.sqlContext.sql ("select * from pysparkdftemptable")
("sqlpool1.dbo.PySparkTable", Constants.INTERNAL)
```



## Question 5

Question Type: Hotspot

You have an Azure Synapse Analytics serverless SQL pool, an Azure Synapse Analytics dedicated SQL pool, an Apache Spark pool, and an Azure Data Lake Storage Gen2 account.

You need to create a table in a lake database. The table must be available to both the serverless SQL pool and the Spark pool.

Where should you create the table, and Which file format should you use for data in the table? TO answer, select the appropriate options in the answer are

a.

NOTE: Each correct selection is worth one point.

Create the table in:

- The dedicated SQL pool
- The serverless SQL pool
- The Spark pool

**Answer:**

File format:

- Apache Parquet
- Delta
- JSON

## Question 6

---

**Question Type:** MultipleChoice

---

You have an Azure Synapse Analytics dedicated SQL pool.

You need to Create a fact table named Table1 that will store sales data from the last three years. The solution must be optimized for the following query operations:

Show order counts by week.

- \* Calculate sales totals by region.
- \* Calculate sales totals by product.
- \* Find all the orders from a given month.

Which data should you use to partition Table1?

## Options:

---

- A- region
- B- product
- C- week
- D- month

## Answer:

---

D

## Explanation:

---

Table partitions enable you to divide your data into smaller groups of data. In most cases, table partitions are created on a date column.

Benefits to queries

Partitioning can also be used to improve query performance. A query that applies a filter to partitioned data can limit the scan to only the qualifying partitions. This method of filtering can avoid a full table scan and only scan a smaller subset of data. With the introduction of clustered columnstore indexes, the predicate elimination performance benefits are less beneficial, but in some cases there can be a benefit to queries.

For example, if the sales fact table is partitioned into 36 months using the sales date field, then queries that filter on the sale date can skip searching in partitions that don't match the filter.



Note: Benefits to loads

The primary benefit of partitioning in dedicated SQL pool is to improve the efficiency and performance of loading data by use of partition deletion, switching and merging. In most cases data is partitioned on a date column that is closely tied to the order in which the data is loaded into the SQL pool. One of the greatest benefits of using partitions to maintain data is the avoidance of transaction logging. While simply inserting, updating, or deleting data can be the most straightforward approach, with a little thought and effort, using partitioning during your load process can substantially improve performance.

## Question 7

---

**Question Type: DragDrop**

---

You have an Azure subscription.

You plan to build a data warehouse in an Azure Synapse Analytics dedicated SQL pool named pool1 that will contain staging tables and a dimensional model Pool1 will contain the following tables.

Table distribution types	Answer Area
Hash	Common.Date: Table distribution type
<b>Answer:</b>	Marketing.WebSessions: Table distribution type
Round-robin  n	Staging.WebSessions: Table distribution type

## Question 8

---

### Question Type: MultipleChoice

---

You have an Azure Databricks workspace and an Azure Data Lake Storage Gen2 account named storage1.

New files are uploaded daily to storage1.

\* Incrementally process new files as they are uploaded to storage1 as a structured streaming source. The solution must meet the following requirements:

- \* Minimize implementation and maintenance effort.
- \* Minimize the cost of processing millions of files.
- \* Support schema inference and schema drift.

Which should you include in the recommendation?

### Options:

---

**A-** Auto Loader

**B-** Apache Spark FileStreamSource

**C-** COPY INTO

D- Azure Data Factory

**Answer:**

---

D

## Question 9

---

**Question Type:** Hotspot

---

You have an Azure Data Factory pipeline that contains a data flow. The data flow contains the following expression.

**Answer Area**

Number of columns:

**Answer:**

---

Number of rows:

## Question 10

---

**Question Type:** DragDrop

---

You have an Azure subscription that contains an Azure Databricks workspace. The workspace contains a notebook named Notebook1. In Notebook1, you create an Apache Spark DataFrame named df\_sales that contains the following columns:

- \* Customer
- \* Salesperson
- \* Region
- \* Amount

You need to identify the three top performing salespersons by amount for a region named HQ.

How should you complete the query? To answer, drag the appropriate values to the correct targets. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Values	Answer Area
<code>agg(col('SalesPerson'))</code>	<code>df_sales.filter(col('Region')== 'HQ').</code> <input type="text"/>
<code>filter(col('SalesPerson'))</code>	<code>.agg(sum('Amount').alias</code> <input type="text"/> <code>.limit(3)</code>
<b>Answer:</b>	
<code>groupBy(col('SalesPerson'))</code>	
<code>groupBy(col('TotalAmount'))</code>	
<code>orderBy(col('TotalAmount'))</code>	
<code>orderBy(desc('TotalAmount'))</code>	

**To Get Premium Files for DP-203 Visit**

**<https://www.p2pexams.com/products/dp-203>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/microsoft/pdf/dp-203>**

