Question 1: **Skipped**

Scenario: You are working at OZcorp which is a multi-million dollar company run by Mayor Norman Osborn. Profits from the company are used to fund Norman's operatives, such as a police task force.

At the moment, you have been hired by OZcorp as a Microsoft Azure Synapse Analytics SME.

Given:

OZcorp has an on-premises data warehouse that includes the following fact tables. Both tables have the following columns: DateKey, ProductKey, RegionKey.

• **Table - Sales:** The table is 600 GB in size. DateKey is used extensively in the `WHERE` clause queries. ProductKey is used extensively in join operations. RegionKey is used for grouping. Seventy-five percent of the records relate to one of forty regions.

• **Table - Invoice:** The table is 6 GB in size. DateKey and ProductKey are used extensively in the `WHERE` clause queries. RegionKey is used for grouping.

• There are 120 unique product keys and 65 unique region keys.

• Queries that use the data warehouse take a long time to complete.

Required:

The team plans to migrate the solution to use Azure Synapse Analytics and they need to ensure that the Azure-based solution optimizes query performance and minimizes processing skew.

Proposed Solution:

The team has chosen to use the following:

• **Table - Sales:** Distribution type: Hash-distributed, Distribution column: ProductKey

• **Table - Invoice:** Distribution type: Round-robin, Distribution column: RegionKey

Azure Synapse Analytics SME, the team looks to you for reassurance that they made the right choices. Did they?

- ○ Yes
- ○ No (Correct)

Explanation

No, the team did not choose the correct option; both hashes are > 2GB. The Invoice table RegionKey cannot be used with Round-robin distribution as Round-robin does not take a distribution key. Hash-distributed for the Distribution type and ProductKey for the Distribution column is correct for the Sales table.

This is because ProductKey is used extensively in joins and Hash-distributed tables improve query performance on large fact tables.

What is a distributed table?

A distributed table appears as a single table, but the rows are actually stored across 60 distributions. The rows are distributed with a hash or round-robin algorithm.

Hash-distributed tables improve query performance on large fact tables, and are the focus of this article. **Round-robin tables** are useful for improving loading speed. These design choices have a significant impact on improving query and loading performance.

Another table storage option is to replicate a small table across all the Compute nodes. For more information, see Design guidance for replicated tables. To quickly choose among the three options, see Distributed tables in the tables overview.

As part of table design, understand as much as possible about your data and how the data is queried. For example, consider these questions:

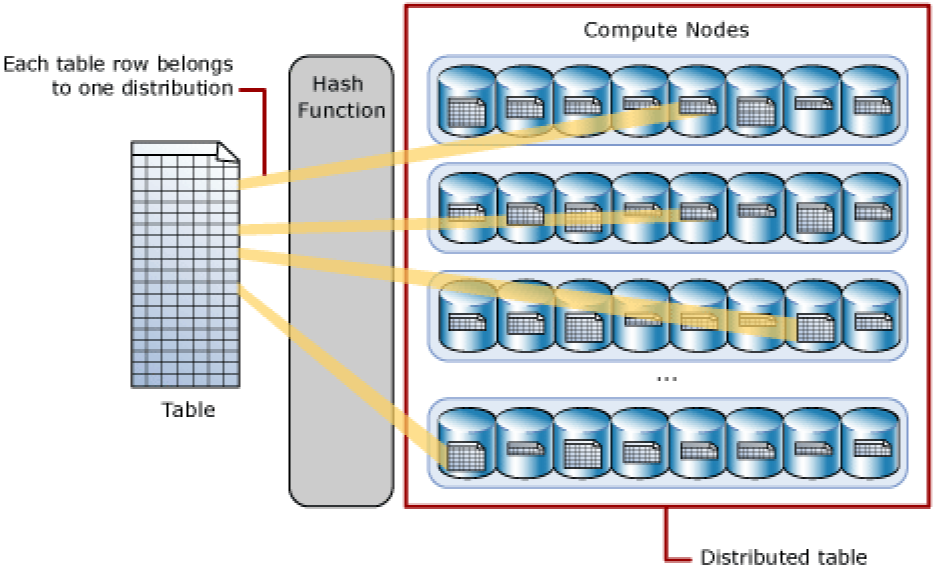
- How large is the table?

- How often is the table refreshed?

- Do I have fact and dimension tables in a dedicated SQL pool?

Hash distributed

A hash-distributed table distributes table rows across the Compute nodes by using a deterministic hash function to assign each row to one distribution.



Since identical values always hash to the same distribution, SQL Analytics has built-in knowledge of the row locations. In dedicated SQL pool this knowledge is used to minimize data movement during queries, which improves query performance.

Hash-distributed tables work well for large fact tables in a star schema. They can have very large numbers of rows and still achieve high performance. There are, of course, some design considerations that help you to get the performance the distributed system is designed to provide. Choosing a good distribution column is one such consideration that is described in this article.

Consider using a hash-distributed table when:

- The table size on disk is more than 2 GB.

- The table has frequent insert, update, and delete operations.

Round-robin distributed

A round-robin distributed table distributes table rows evenly across all distributions. The assignment of rows to distributions is random. Unlike hash-distributed tables, rows with equal values are not guaranteed to be assigned to the same distribution.

As a result, the system sometimes needs to invoke a data movement operation to better organize your data before it can resolve a query. This extra step can slow down your queries. For example, joining a round-robin table usually requires reshuffling the rows, which is a performance hit.

Consider using the round-robin distribution for your table in the following scenarios:

- When getting started as a simple starting point since it is the default

- If there is no obvious joining key

- If there is no good candidate column for hash distributing the table

- If the table does not share a common join key with other tables

- If the join is less significant than other joins in the query

- When the table is a temporary staging table

Choosing a distribution column

A hash-distributed table has a distribution column that is the hash key. For example, the following code creates a hash-distributed table with ProductKey as the distribution column.

```
SQL
CREATE TABLE [dbo].[FactInternetSales]
(   [ProductKey]            int          NOT NULL
,   [OrderDateKey]          int          NOT NULL
,   [CustomerKey]           int          NOT NULL
,   [PromotionKey]          int          NOT NULL
,   [SalesOrderNumber]      nvarchar(20) NOT NULL
,   [OrderQuantity]         smallint     NOT NULL
,   [UnitPrice]             money        NOT NULL
,   [SalesAmount]           money        NOT NULL
)
WITH
(   CLUSTERED COLUMNSTORE INDEX
,  DISTRIBUTION = HASH([ProductKey])
)
;
```

Data stored in the distribution column can be updated. Updates to data in the distribution column could result in data shuffle operation.

Choosing a distribution column is an important design decision since the values in this column determine how the rows are distributed. The best choice depends on several factors, and usually involves tradeoffs. Once a distribution column is chosen, you cannot change it.

If you didn't choose the best column the first time, you can use CREATE TABLE AS SELECT (CTAS) to re-create the table with a different distribution column.

Choose a distribution column with data that distributes evenly

For best performance, all of the distributions should have approximately the same number of rows. When one or more distributions have a disproportionate number of rows, some distributions finish their portion of a parallel query before others. Since the query can't complete until all distributions have finished processing, each query is only as fast as the slowest distribution.

Data skew means the data is not distributed evenly across the distributions

Processing skew means that some distributions take longer than others when running parallel queries. This can happen when the data is skewed.

To balance the parallel processing, select a distribution column that:

Has many unique values. The column can have some duplicate values. However, all rows with the same value are assigned to the same distribution. Since there are 60 distributions, the column should have at least 60 unique values. Usually the number of unique values is much greater.

Does not have NULLs, or has only a few NULLs. For an extreme example, if all values in the column are NULL, all the rows are assigned to the same distribution. As a result, query processing is skewed to one distribution, and does not benefit from parallel processing.

Is not a date column. All data for the same date lands in the same distribution. If several users are all filtering on the same date, then only 1 of the 60 distributions do all the processing work.

Choose a distribution column that minimizes data movement

To get the correct query result queries might move data from one Compute node to another. Data movement commonly happens when queries have joins and aggregations on distributed tables. Choosing a distribution column that helps minimize data movement is one of the most important strategies for optimizing performance of your dedicated SQL pool.

To minimize data movement, select a distribution column that:

Is used in `JOIN`, `GROUP BY`, `DISTINCT`, `OVER`, and `HAVING` clauses. When two large fact tables have frequent joins, query performance improves when you distribute both tables on one of the join columns. When a table is not used in joins, consider distributing the table on a column that is frequently in the `GROUP BY` clause.

Is *not* used in `WHERE` clauses. This could narrow the query to not run on all the distributions.

Is *not* a date column. `WHERE` clauses often filter by date. When this happens, all the processing could run on only a few distributions.

https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-tables-distribute

---

Question 2: Skipped

What should be done when a connector in data factory is not supported in mapping data flow in order to transform data from one of these sources? (Select all that apply)

- □ Use a generic ODBC connector. (Correct)
- □ Ingest the data into a supported source using the copy activity. (Correct)
- □ Use a group by activity in Dataflow.
- □ Use a generic REST connector. (Correct)
- □ Use an aggregate transformation in Dataflow.

Explanation

If a connector in Data factory is not supported, create a copy activity of the source data into a supported data source in mapping dataflow and continue the transformations from there.

Integrate with more data stores

Azure Data Factory can reach a very broad set of data stores. If you need to move data to/from a data store that is not in the Azure Data Factory built-in connector list, here are some extensible options:

• For database and data warehouse, usually you can find a corresponding ODBC driver, with which you can use generic ODBC connector.

• For SaaS applications:

- If it provides RESTful APIs, you can use generic REST connector.
- If it has OData feed, you can use generic OData connector.
- If it provides SOAP APIs, you can use generic HTTP connector.
- If it has ODBC driver, you can use generic ODBC connector.

- For others, check if you can load data to or expose data as any ADF supported data stores, e.g. Azure Blob/File/FTP/SFTP/etc, then let ADF pick up from there. You can invoke custom data loading mechanism via Azure Function, Custom activity, Databricks/HDInsight, Web activity, etc.

Question 3: Skipped

What size does `OPTIMIZE` compact small files to?

- ○ Around 1 GB (Correct)
- ○ Around 2 GB
- ○ Around 100 MB
- ○ Around 500 MB

Explanation

The `OPTIMIZE` command compacts small files to around 1GB. The Spark optimization team determined this value to be a good compromise between speed and performance.

https://docs.databricks.com/spark/latest/spark-sql/language-manual/delta-optimize.html

Question 4: Skipped

Scenario: You are working on a project and your team is moving data from an Azure Data Lake Gen2 store to Azure Synapse Analytics. The team is planning to do a data copy activity and you are discussing with integration runtime to use.

Which Azure Data Factory integration runtime should be used in a data copy activity?

- ○ Azure (Correct)
- ○ Datasets
- ○ Activities
- ○ Linked Services
- ○ Self-hosted
- ○ Azure-SSIS

Explanation

When moving data between Azure data platform technologies, the Azure Integration runtime is used when copying data between two Azure data platform.

Integration runtime types

Data Factory offers three types of Integration Runtime, and you should choose the type that best serve the data integration capabilities and network environment needs you are looking for. These three types are:

- Azure
- Self-hosted
- Azure-SSIS

You can explicitly define the Integration Runtime setting in the **connectVia** property, if this is not defined, then the default Integration Runtime is used with the property set to Auto-Resolve.

The following describes the capabilities and network support for each of the integration runtime types:

IR type: Azure

Public network: Data Flow Data movement Activity dispatch

Private network: --

IR type: Self-hosted

Public network: Data movement Activity dispatch

Private network: Data movement Activity dispatch

IR type: Azure-SSIS

Public network: SSIS package execution

Private network: SSIS package execution

https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime

Question 5: Skipped

How do you list files in DBFS within a notebook?

- ○ `%dfs ls /my-file-path`
- ○ `%fs ls /my-file-path` (Correct)
- ○ `ls /my-file-path`
- ○ `%fs dir /my-file-path`

Explanation

DBFS and local driver node paths

You can work with files on DBFS or on the local driver node of the cluster. You can access the file system using magic commands such as `%fs` or `%sh`.

You add the file system magic to the cell before executing the ls command.

https://docs.microsoft.com/en-us/azure/databricks/data/databricks-file-system

Question 6: Skipped

A data warehouse that is built on a Massively Parallel Processing (MPP) system is built for processing and analyzing large datasets. As such they perform well with larger batch type loads and updates that can be distributed across the compute nodes and storage.

Which of the following is the best approach if singleton or smaller transaction batch loads must be added to an MPP data warehouse?

- ○ None of the listed options.
- ○ Manually create an append file with a trigger that once the contents of the manually created file reach a predetermined size, an automation process will be triggered to append the data to the data warehouse.
- ○ Develop a process that writes the outputs of an INSERT statement to a to the target file automatically, avoiding the need to do the INSERT manually.
- ○ Develop two processes: one that writes the outputs of an INSERT statement to a file, and then another process to periodically load this file.
  (Correct)
- ○ All the approaches are equally valid.

Explanation

A data warehouse that is built on a Massively Parallel Processing (MPP) system is built for processing and analyzing large datasets. As such they perform well with larger batch type loads and updates that can be distributed across the compute nodes and storage.

Singleton or smaller transaction batch loads should be grouped into larger batches to optimize the Synapse SQL Pools processing capabilities. To be clear, A one-off load to a small table with an INSERT statement may be the best approach, if it is a one-off.

However, if you need to load thousands or millions of rows throughout the day, then singleton `INSERT`s aren't optimal against an MPP system. One way to solve this issue is to develop one process that writes the outputs of an `INSERT` statement to a file, and then another process to periodically load this file to take advantage of the parallelism that Azure Synapse Analytics.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-best-practices

Question 7: Skipped

Scenario: You are working in a department which requires preparation of data for ad hoc data exploration and analysis based on market fluctuations. The Department Head has tasked you with determining the most effective resource model in Azure Synapse Analytics to employ.

Which of the following should you choose?

- ○ Databricks
- ○ Serverless (Correct)
- ○ Dedicated
- ○ IoT Central
- ○ Pipelines

Explanation

Serverless SQL pool is a pay per query service that doesn't require you to pick the right size. The system automatically adjusts based on your requirements, freeing you up from managing your infrastructure and picking the right size for your solution.

The serverless resource model is the ideal resource model in this scenario as it makes use of the resources when required.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/resource-consumption-models

Question 8: **Skipped**

Scenario: You are working as a consultant at Avengers Security and at the moment, you are working with the data engineering team which manages Azure HDInsight clusters at the company. The group spends an enormous amount of time creating and destroying clusters each day due to the fact that the majority of the data pipeline process runs in minutes.

Required: Utilize a solution which will deploy multiple HDInsight clusters with minimal effort.

Which of the following should recommend to the IT team to implement?

- ○ Azure Databricks
- ○ Azure PowerShell
- ○ Azure Traffic Manager
- ○ Azure Resource Manager templates (Correct)

Explanation

A Resource Manager template makes it easy to create the following resources for your application in a single, coordinated operation:

• HDInsight clusters and their dependent resources (such as the default storage account).

• Other resources (such as Azure SQL Database to use Apache Sqoop).

In the template, you define the resources that are needed for the application. You also specify deployment parameters to input values for different environments.

The template consists of JSON and expressions that you use to construct values for your deployment.

https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-create-linux-clusters-arm-templates

Question 9: **Skipped**

True or False: In simple terms, you could view DataFrames as you might see in excel, which we could also refer to as a table of data

- ○ False
- ○ True (Correct)

Explanation

What are dataframes?

Basically you could view DataFrames as you might see in excel. It's like a box with squares in it, that organizes data, which we could also refer to as a table of data.

What does a table of data mean?

It is a single set of two-dimensional data that can have multiple rows and columns in the data. Each row, is a sample of data. Each column is a variable or parameter that is able to describe the row that contains the sample of data.

A DataFrame creates a data structure and it's one of the core data structures in Spark. In Spark, it is seen as a distributed collection of data that is organized into columns that have names.

What you see in Data Engineering is that you start with reading or loading data that can be unstructured, semi-structured, or structured, which is stored in a DataFrame and start transforming that data in order to get insights. You can use different functionalities in order to do so, like using Spark SQL, PySpark, and others.

Usually when you see 'df' in some code it refers to a dataframe.

You can either create your own dataframe as this example shows:

```
Python

new_rows = [('CA',22, 45000),("WA",35,65000) ,("WA",50,85000)]

demo_df = spark.createDataFrame(new_rows, ['state', 'age', 'salary'])

demo_df.show()
```

Or load a file that contains data into a dataframe like in the below example where the open taxi dataset is used:

```
Python

from azureml.opendatasets import NycTlcYellow
```

```
data = NycTlcYellow()

data_df = data.to_spark_dataframe()

display(data_df.limit(10))
```

Once you're at the stage where you'd like to manipulate the data that is stored in a DataFrame, you can use User-Defined Functions (UDFs) that are column-based and help you transform and manipulate the data stored in a DataFrame.

https://www.tutorialspoint.com/spark_sql/spark_sql_dataframes.htm

Question 10: **Skipped**

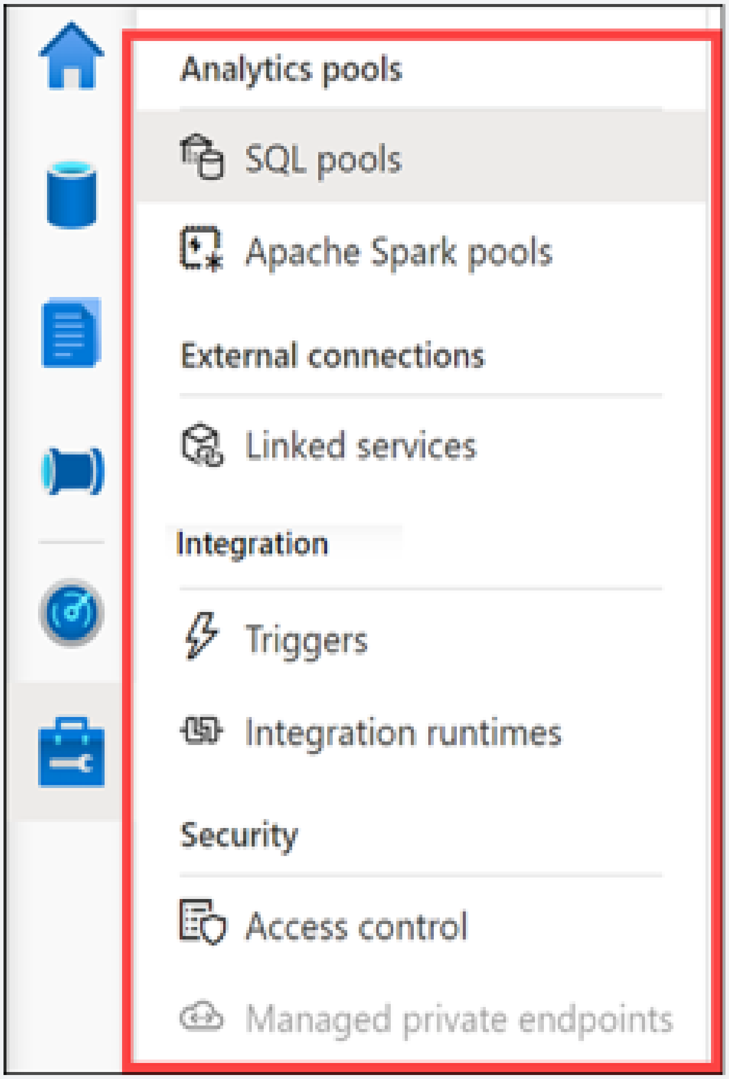
Which hub is where you can grant access to Synapse workspace and resources?

- ○ Integrate hub
  (Correct)
- ○ Integrate hub
- ○ None of the listed options
- ○ Data hub
- ○ Create hub
- ○ Monitor hub

Explanation

You can grant access to Synapse workspace in the integrate hub.

In Azure Synapse Studio, the Manage hub enables you to perform some of the same actions available in the Azure portal, such as managing SQL and Spark pools. However, there is a lot more you can do in this hub that you cannot do anywhere else, such as managing Linked Services and integration runtimes, and creating pipeline triggers.



- **SQL pools**. Lists the provisioned SQL pools and on-demand SQL serverless pools for the workspace. You can add new pools or hover over a SQL pool to **pause** or **scale** it. You should pause a SQL pool when it's not being used to save costs.

- **Apache Spark pools**. Lets you manage your Spark pools by configuring the auto-pause and auto-scale settings. You can provision a new Apache Spark pool from this blade.

- **Linked services**. Enables you to manage connections to external resources. Here you can see linked services for our data lake storage account, Azure Key Vault, Power BI, and Synapse Analytics. **Task**: Select **+ New** to show how many types of linked services you can add.

• **Triggers**. Provides you a central location to create or remove pipeline triggers. Alternatively, you can add triggers from the pipeline.

• **Integration runtimes**. Lists the IR for the workspace, which serves as the compute infrastructure for data integration capabilities, like those provided by pipelines. **Task**: Hover over the integration runtimes to show the monitoring, code, and delete (if applicable) links. Click on a **code link** to show how you can modify the parameters in JSON format, including the TTL (time to live) setting for the IR.

• **Access control**. This is where you go to add and remove users to one of three security groups: workspace admin, SQL admin, and Apache Spark for Azure Synapse Analytics admin.

• **Managed private endpoints**. This is where you manage private endpoints, which use a private IP address from within a virtual network to connect to an Azure service or your own private link service. Connections using private endpoints listed here provide access to Synapse workspace endpoints (SQL, SqlOndemand and Dev).

https://techcommunity.microsoft.com/t5/azure-synapse-analytics/explore-the-manage-hub-in-synapse-studio-to-provision-and-secure/ba-p/1987788

Question 11: Skipped

Which is an element of a Spark Pool in Azure Synapse Analytics?

- ○ Spark Instance (Correct)
- ○ HDI
- ○ Spark Console
- ○ Databricks

Explanation

The definition of a Spark pool is that, when instantiated, it is used to create a Spark instance that processes data.

Apache Spark is a parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications. Apache Spark in Azure Synapse Analytics is one of Microsoft's implementations of Apache Spark in the cloud.

Azure Synapse makes it easy to create and configure Spark capabilities in Azure. Azure Synapse provides a different implementation of these Spark capabilities that are documented here.

Spark pools

A serverless Apache Spark pool is created in the Azure portal. It's the definition of a Spark pool that, when instantiated, is used to create a Spark instance that processes data. When a Spark pool is created, it exists only as metadata, and no resources are consumed, running, or charged for. A Spark pool has a series of properties that control the characteristics of a Spark instance. These characteristics include but aren't limited to name, size, scaling behaviour, time to live.

As there's no dollar or resource cost associated with creating Spark pools, any number can be created with any number of different configurations. Permissions can also be applied to Spark pools allowing users only to have access to some and not others.

A best practice is to create smaller Spark pools that may be used for development and debugging and then larger ones for running production workloads.

Spark instances

Spark instances are created when you connect to a Spark pool, create a session, and run a job. As multiple users may have access to a single Spark pool, a new Spark instance is created for each user that connects.

When you submit a second job, if there is capacity in the pool, the existing Spark instance also has capacity. Then, the existing instance will process the job. Otherwise, if capacity is available at the pool level, then a new Spark instance will be created.

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-concepts

Question 12: Skipped

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] is a fully managed cloud service. Analysts, data scientists, developers, and others use [?] to discover, understand, and consume data sources. It features a crowdsourcing model of metadata and annotations. In this central location, an organization's users contribute their knowledge to build a community of data sources that are owned by the organization.

- ○ Azure Cosmos DB
- ○ Azure Databricks
- ○ Azure Storage Explorer
- ○ Azure Data Factory
- ○ Azure Data Catalog **(Correct)**
- ○ Azure Data Lake Storage
- ○ Azure SQL Datawarehouse

Explanation

Azure Data Catalog

Analysts, data scientists, developers, and others use Data Catalog to discover, understand, and consume data sources. Data Catalog features a crowdsourcing model of metadata and annotations. In this central location, an organization's users contribute their knowledge to build a community of data sources that are owned by the organization.

Data Catalog is a fully managed cloud service. Users discover and explore data sources, and they help the organization document information about their data sources.

https://docs.microsoft.com/en-us/azure/data-catalog/overview

Question 13: Skipped

True or False: Mapping data flows are visually displayed data transformations in Azure Data Factory. Data flows allow data engineers to develop data transformation logic with or without writing code.

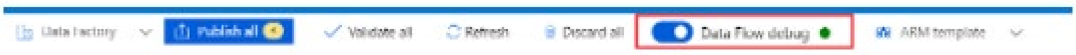
- ○
  False
  (Correct)
- ○
  True

Explanation

Transforming data with the Mapping Data Flow

You can natively perform data transformations with Azure Data Factory code free using the Mapping Data Flow task. **Mapping Data Flows provide a fully visual experience with no coding required.** Your data flows will run on your own execution cluster for scaled-out data processing. Data flow activities can be operationalized via existing Data Factory scheduling, control, flow, and monitoring capabilities.

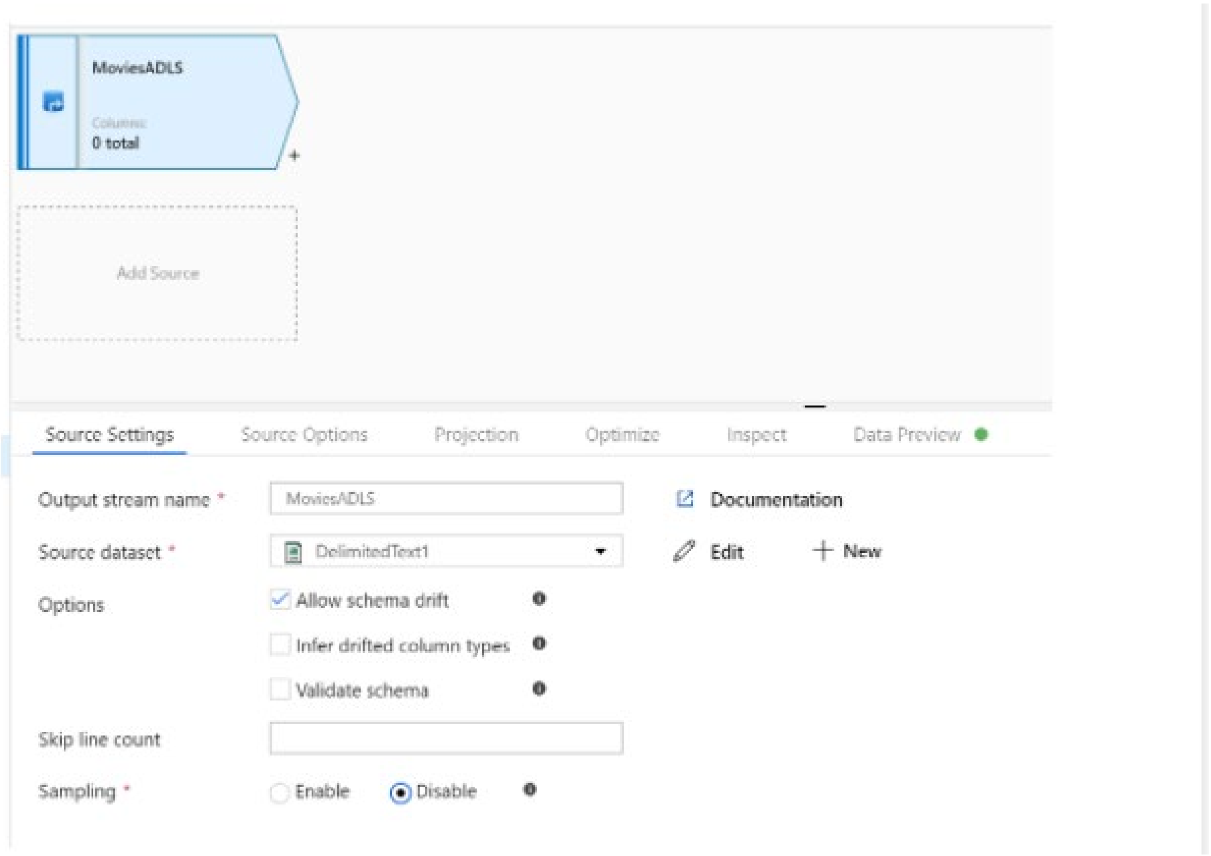
When building data flows, you can enable debug mode, which turns on a small interactive Spark cluster. Turn on debug mode by toggling the slider at the top of the authoring module. Debug clusters take a few minutes to warm up, but can be used to interactively preview the output of your transformation logic.



With the Mapping Data Flow added, and the Spark cluster running, this will enable you to perform the transformation, and run and preview the data. **No coding is required as Azure Data Factory handles all the code translation, path optimization, and execution of your data flow jobs.**

Adding source data to the Mapping Data Flow

Open the Mapping Data Flow canvas. Click on the Add Source button in the Data Flow canvas. In the source dataset dropdown, select your data source, n this case the ADLS Gen2 dataset is used in this example



Output stream name * MoviesADLS

Source dataset * DelimitedText1

Options ☑ Allow schema drift

□ Infer drifted column types

□ Validate schema

Skip line count

Sampling * ○ Enable ● Disable

There are a couple of points to note:

• If your dataset is pointing at a folder with other files and you only want to use one file, you may need to create another dataset or utilize parameterization to make sure only a specific file is read

• If you have not imported your schema in your ADLS, but have already ingested your data, go to the dataset's 'Schema' tab and click 'Import schema' so that your data flow knows the schema projection.

Mapping Data Flow follows an extract, load, transform (ELT) approach and works with staging datasets that are all in Azure. Currently the following datasets can be used in a source transformation:

• Azure Blob Storage (JSON, Avro, Text, Parquet)

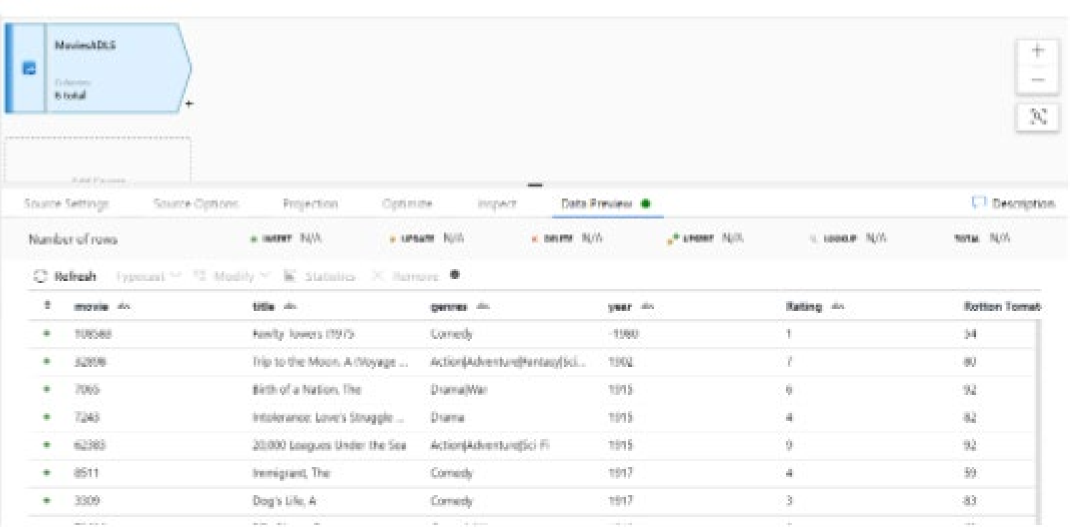
• Azure Data Lake Storage Gen1 (JSON, Avro, Text, Parquet)

• Azure Data Lake Storage Gen2 (JSON, Avro, Text, Parquet)

- Azure Synapse Analytics
- Azure SQL Database
- Azure CosmosDB

Azure Data Factory has access to over 80 native connectors. To include data from those other sources in your data flow, use the Copy Activity to load that data into one of the supported staging areas.

Once your debug cluster is warmed up, verify your data is loaded correctly via the Data Preview tab. Once you click the refresh button, Mapping Data Flow will show a snapshot of what your data looks like when it is at each transformation.



Number of rows INSERT N/A UPSERT N/A DELETE N/A UPDATE N/A LOOKUP N/A TOTAL N/A

↕	movie	title	genres	year	Rating	Rotton Tomat
●	108583	Fawlty Towers (1975)	Comedy	1980	1	34
●	32898	Trip to the Moon, A (Voyage ...	Action|Adventure|Fantasy|Sci...	1902	7	80
●	7065	Birth of a Nation, The	Drama|War	1915	6	92
●	7243	Intolerance: Love's Struggle ...	Drama	1915	4	82
●	62383	20,000 Leagues Under the Sea	Action|Adventure|Sci-Fi	1915	9	92
●	8511	Immigrant, The	Comedy	1917	4	59
●	3309	Dog's Life, A	Comedy	1917	3	83

https://docs.microsoft.com/en-us/azure/data-factory/concepts-data-flow-overview

Question 14: Skipped

Which Transact-SQL function verifies if a piece of text is valid JSON?

- ○ `ISJSON` (Correct)
- ○ `JSON_VALID`
- ○ None of the listed options
- ○ `JSON_VALUE`
- ○ `JSON_QUERY`

Explanation

`ISJSON` is a Transact-SQL function that verifies if a piece of text is valid JSON.

https://docs.microsoft.com/en-us/sql/t-sql/functions/isjson-transact-sql?view=sql-server-ver15

Question 15: **Skipped**

Synapse Analytics removes the barrier of setting up multiple different services for Spark or SQL. The interoperability between Spark and SQL helps you achieve as follows:

• A shared Hive-compatible metadata system enables you to define tables on files in the data lake such that it can be consumed by either Spark or Hive.

• Both SQL and Spark can directly explore, and analyze Parquet, CSV, TSV, and JSON files stored in the data lake.

• The enablement of fast scalable load and unload for data transferring between SQL and Spark databases.

The Azure Synapse Apache Spark to Synapse SQL connector is designed to efficiently transfer data between serverless Apache Spark pools and SQL pools in Azure Synapse.

Which of the following are valid use cases for Apache Spark and SQL integration within Synapse analytics? (Select all that apply)

- □ Flexibility in the use of Spark and SQL languages and frameworks **(Correct)**
- □ VNet and On-prem sync
- □ Scalability **(Correct)**
- □ Dealing with different type of analytics **(Correct)**
- □ Big data computational powers **(Correct)**

Explanation

Synapse Analytics removes the barrier of setting up multiple different services for Spark or SQL. Therefore, it removes the traditional thinking about these technologies. It enables you to use both technologies within one platform, which allowed you to switch between Spark or SQL based on the needs and expertise you have in-house.

A spark orientated data engineer can now easily communicate with a SQL based data engineer and communicate together on the same platform.

The interoperability between Spark and SQL helps you achieve as follows:

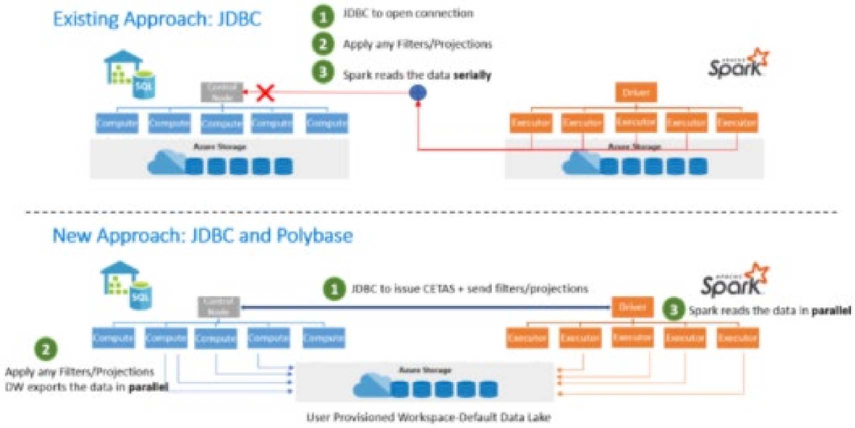
• A shared Hive-compatible metadata system enables you to define tables on files in the data lake such that it can be consumed by either Spark or Hive.

• Both SQL and Spark can directly explore, and analyze Parquet, CSV, TSV, and JSON files stored in the data lake.

• The enablement of fast scalable load and unload for data transferring between SQL and Spark databases.

The question might raise as how would that SQL and Spark integration then work.

That's when the Azure Synapse Apache Spark to Synapse SQL connector comes in place. It is designed to efficiently transfer data between serverless Apache Spark pools (preview) and SQL pools in Azure Synapse. However, at the moment, the Azure Synapse Apache Spark to Synapse SQL connector works on dedicated SQL pools only, it doesn't work with serverless SQL pools.


Existing Approach: JDBC
1 JDBC to open connection
2 Apply any Filters/Projections
3 Spark reads the data serially
Spark
SQL
Control Node
Compute
Compute
Compute
Compute
Compute
Azure Storage
Driver
Executor
Executor
Executor
Executor
Executor
Azure Storage
New Approach: JDBC and Polybase
1 JDBC to issue CETAS + send filters/projections
Spark
SQL
Control Node
Driver
3 Spark reads the data in parallel
Compute
Compute
Compute
Compute
Compute
Executor
Executor
Executor
Executor
Executor
2
Apply any Filters/Projections
DW exports the data in parallel
Azure Storage
User Provisioned Workspace-Default Data Lake

In the commonly used existing approach, you often see the use of the JDBC. The JDBC would open the connection. Then, filters and projections would be applied and spark would read the data serially. Given two distributed systems such as Spark and SQL pools, JDBC could become a bottleneck with serial data transfer.

Therefore the New Approach we would take is JDBC and PolyBase. First, the JDBC issues CETAS and send filters and projections. Then filters and projections would be applied and the DataWarehouse exports the data in parallel. Spark reads the data in parallel all based on the user provisioned workspace default data lake storage.

The Azure Synapse Apache Spark Pool to Synapse SQL connector would then be a data source implementation for apache spark where the ADLS Gen 2 is used as well as PolyBase in the dedicated SQL Pools to transfer data between the Spark instance and SQL pool efficiently.

The use cases for Apache Spark and SQL integration within Synapse analytics are as following:

- **Dealing with different type of analytics**

- **Scalability**

- **Big data computational powers**

- **Flexibility in the use of Spark and SQL languages and frameworks**

Since Apache Spark is integrated in Synapse Analytics, there is more to that than giving use for the big data analytics framework Apache Spark enables. When you deploy a synapse cluster, ADLS Gen2 capacity that can store Spark SQL Tables is provisioned with it.

If you use Spark SQL Tables, you might know that these tables can be queried from a SQL-server-based T-SQL language without you having to use commands like CREATE EXTERNAL TABLE. Within synapse analytics, these queries integrate natively with data files that are stored in an Apache Parquet format.

The other thing to take in mind is that beyond the capabilities mentioned above, the Azure Synapse Studio experience gives you an integrated notebook experience. Within this notebook experience, you can attach a SQL or Spark pool, and develop and execute, for example, transformation pipelines using Python, Scala, and native Spark SQL.

So, let's say you would like to write to a SQL pool after you've performed engineering tasks in spark. You can reference the SQL Pool data as a source for joining with Spark Dataframes that can contain data from other files. When you decide to use the Azure Synapse Apache Spark to Synapse SQL connector, you're now able to efficiently transfer data between the Spark and SQL Pools.

The Azure Synapse Apache Spark pool to Synapse SQL connector is a data source implementation for Apache Spark. It uses the Azure Data Lake Storage Gen2 and PolyBase in SQL pools to efficiently transfer data between the Spark cluster and the Synapse SQL instance.

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/synapse-spark-sql-pool-import-export

Question 16: **Skipped**

Scenario: Dr. Karl Malus works for the Power Broker Corporation (PBC) founded by Curtiss Jackson, using technology to service various countries and their military efforts. You have been contracted by the company to assist Dr. Malus with their Microsoft Azure Synapse projects.

PBC has an Azure subscription that contains a logical Microsoft SQL server named Server1. Server1 hosts an Azure Synapse Analytics SQL dedicated pool named Pool1.

Dr. Malus is looking for a recommendation for a Transparent Data Encryption (TDE) solution for Server1. The solution must meet the following requirements:

Required:

• Track the usage of encryption keys.

• Maintain the access of client apps to Pool1 in the event of an Azure datacentre outage that affects the availability of the encryption keys.

Which of the following should you include in the recommendation for the "Track the usage of encryption key" requirement?

- ○ TDE with customer-managed keys **(Correct)**
- ○ Always Encrypted
- ○ Any of the options listed will meet the requirement
- ○ TDE with platform-managed keys
- ○ None of the options listed will meet the requirement

Explanation

You should include in "TDE with customer-managed keys" in the recommendation for the first requirement listed.

Customer-managed keys are stored in the Azure Key Vault. You can monitor how and when your key vaults are accessed, and by whom. You can do this by enabling logging for Azure Key Vault, which saves information in an Azure storage account that you provide.

After you create one or more key vaults, you'll likely want to monitor how and when your key vaults are accessed, and by whom. You can do this by enabling logging for Azure Key Vault, which saves information in an Azure storage account that you provide. For step by step guidance on setting this up, see How to enable Key Vault logging.

What is logged:

All authenticated REST API requests, including failed requests as a result of access permissions, system errors, or bad requests.

Operations on the key vault itself, including creation, deletion, setting key vault access policies, and updating key vault attributes such as tags.

Operations on keys and secrets in the key vault, including: Creating, modifying, or deleting these keys or secrets. Signing, verifying, encrypting, decrypting, wrapping and unwrapping keys, getting secrets, and listing keys and secrets (and their versions).

Unauthenticated requests that result in a 401 response. Examples are requests that don't have a bearer token, that are malformed or expired, or that have an invalid token.

Azure Event Grid notification events for the following conditions: expired, near expiration, and changed vault access policy (the new version event isn't logged). Events are logged even if there's an event subscription created on the key vault.

You can access your logging information 10 minutes (at most) after the key vault operation. In most cases, it will be quicker than this. It's up to you to manage your logs in your storage account:

Use standard Azure access control methods in your storage account to secure your logs by restricting who can access them.

Delete logs that you no longer want to keep in your storage account.

https://docs.microsoft.com/en-us/azure/key-vault/general/logging?tabs=Vault

Question 17: **Skipped**

Scenario: While working on a project using Azure Data Factory, you are routing data rows to different streams based on matching conditions. Which transformation in Mapping Data Flow is used to do this?

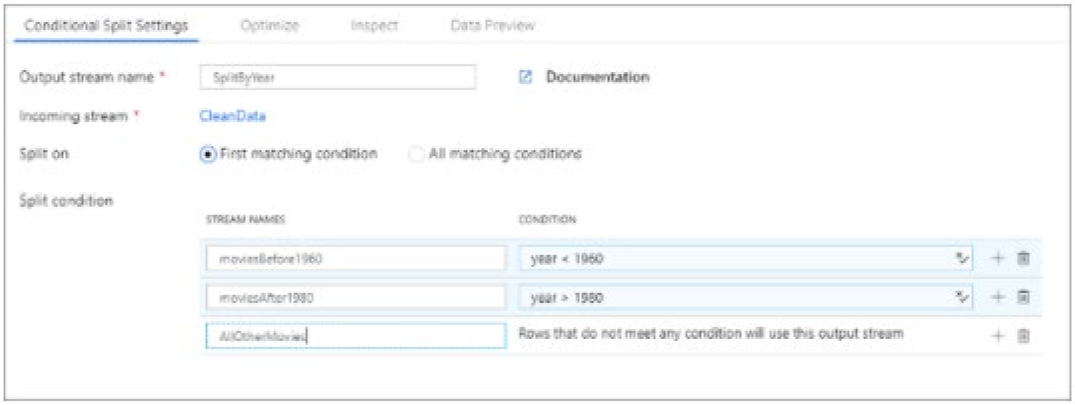
- ○ Select
- ○ Optimize
- ○ Conditional Split **(Correct)**
- ○ Lookup
- ○ Inspect

Explanation

Conditional Split transformation routes data rows to different streams based on matching conditions. The conditional split transformation is similar to a CASE decision structure in a programming language.

The **Split on** setting determines whether the row of data flows to the first matching stream or every stream it matches to.

Use the data flow expression builder to enter an expression for the split condition. To add a new condition, click on the plus icon in an existing row. A default stream can be added as well for rows that don't match any condition.



Output stream name * SplitByYear

Incoming stream * CleanData

Split on ● First matching condition ○ All matching conditions

Split condition

STREAM NAMES	CONDITION
moviesBefore1960	year < 1960
moviesAfter1980	year > 1980
AllOtherMovies	Rows that do not meet any condition will use this output stream

https://docs.microsoft.com/en-us/azure/data-factory/data-flow-conditional-split

Question 18: **Skipped**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

A(n) [?] schema must be defined before query time.

- ○ Unstructured data type
- ○ Structured data type **(Correct)**
- ○ Hybrid data type
- ○ Azure Cosmos DB data type

Explanation

Structured data

In relational database systems like Microsoft SQL Server, Azure SQL Database, and Azure SQL Data Warehouse, data structure is defined at design time. Data structure is designed in the form of tables. This means it's designed before any information is loaded into the system. The data structure includes the relational model, table structure, column width, and data types.

Relational systems react slowly to changes in data requirements because the structural database needs to change every time a data requirement changes. When new columns are added, you might need to bulk-update all existing records to populate the new column throughout the table.

Relational systems typically use a querying language such as Transact-SQL (T-SQL).

https://k21academy.com/microsoft-azure/dp-900/relational-and-non-relational-datastores/

Question 19: **Skipped**

Scenario: You are new on the job and are looking through the Azure knowledgebase to determine which Azure product is the right choice for an ingestion point for data streaming in an event processing solution that uses static data as a source.

You narrowed the choices down to the below list.

Which is the best choice?

- ○ Azure Sphere
- ○ Azure Blob Storage (Correct)
- ○ Azure IoT Hub
- ○ Azure Event Hubs
- ○ Azure IoT Central

Explanation

Azure Blob storage provides an ingestion point for data streaming in an event processing solution that uses static data as a source.

https://docs.microsoft.com/en-us/azure/data-explorer/ingest-data-overview

Question 20: **Skipped**

Scenario: You are working on a project with a 3rd party vendor to build a website for a customer. The image assets that will be used on the website are stored in an Azure Storage account that is held in your subscription. You want to give read access to this data for a limited period of time.

What security option would be the best option to use?

- ○ Shared Access Signatures **(Correct)**
- ○ Storage Account
- ○ Private Link
- ○ CORS Support

Explanation

A shared access signature is a string that contains a security token that can be attached to a URI. Use a shared access signature to delegate access to storage objects and specify constraints, such as the permissions and the time range of access.

Shared Access Signatures (SAS)

Access keys are the easiest approach to authenticating access to a storage account. However they provide full access to anything in the storage account, similar to a root password on a computer.

Storage accounts offer a separate authentication mechanism called *shared access signatures* that support expiration and limited permissions for scenarios where you need to grant limited access. You should use this approach when you are allowing other users to read and write data to your storage account. There are links to our documentation on this advanced topic at the end of the module.

https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

Question 21: Skipped

In Data Factory, an activity defines the action to be performed. A linked service defines a target data store or a compute service. An integration runtime provides the bridge between the activity and linked services.

An Azure integration runtime is capable of which of the following? (Select all that apply)

- □
  Triggering batch movement of ETL data on a dynamic schedule for most analytics solutions.
- □
  All the listed options.
- □
  None of the listed options.
- □
  Running Data Flows in Azure
  (Correct)
- □
  Dispatching transform activities in public network utilizing platforms such as Databricks Notebook/ Jar/ Python activity, HDInsight Hive activity and more.
  (Correct)
- □
  Running Copy Activity between cloud data stores
  (Correct)

Explanation

In Data Factory, an activity defines the action to be performed. A linked service defines a target data store or a compute service. An integration runtime provides the bridge between the activity and linked services.

Azure integration runtime

An Azure integration runtime is capable of:

• Running Data Flows in **Azure**

• Running Copy Activity **between cloud data stores**

• Dispatching the following transform activities in **public network**: Databricks Notebook/ Jar/ Python activity, HDInsight Hive activity, HDInsight Pig activity, HDInsight MapReduce activity, HDInsight Spark activity, HDInsight Streaming activity, Machine Learning Batch Execution activity, Machine Learning Update Resource activities, Stored Procedure activity, Data Lake Analytics U-SQL activity, .NET custom activity, Web activity, Lookup activity, and Get Metadata activity.

You can set a certain location of an Azure IR, in which case the data movement or activity dispatch will happen in that specific region. If you choose to use the auto-resolve Azure IR which is the default, ADF will make a best effort to automatically detect your sink and source data store to choose the best location either in the same region if available or the closest one in the same geography for the Copy Activity. For anything else, it will use the IR in the Data Factory region. Azure Integration Runtime also has support for virtual networks.

https://docs.microsoft.com/en-us/azure/data-factory/concepts-integration-runtime

Question 22: **Skipped**

Scenario: Pennyworth's Haberdashery is a clothing retailer based in London. The company has 2,000 retail stores across the EU and an emerging online presence. The network contains an Active Directory forest named pennyworths.com. The forest it integrated with an Azure Active Directory (Azure AD) tenant named pennyworths.com. Pennyworth's has an Azure subscription associated to the pennyworths.com Azure AD tenant.

Pennyworth's has three years of customer, transactional, operational, sourcing, and supplier data comprised of 10 billion records stored across multiple on-premises. Microsoft SQL Server servers. The SQL Server instances contain data from various operational systems. The data is loaded into the instances by using SQL Server Integration Services (SSIS) packages.

You have been hired as a consultant by Alfred Pennyworth to advise on very important projects within the company.

During your assessment of the IT environment, you estimate that combining all product sales transactions into a company-wide sales transactions dataset will result in a single table that contains 5 billion rows, with one row per transaction.

Most queries targeting the sales transactions data will be used to identify which products were sold in retail stores and which products were sold online during different time periods. Sales transaction data that is older than three years will be removed monthly.

The IT team plans to create a retail store table that will contain the address of each retail store. The table will be approximately 2 MB. Queries for retail store sales will include the retail store addresses.

They also plan to create a promotional table that will contain a promotion ID. The promotion ID will be associated to a specific product. The product will be identified by a product ID. The table will be approximately 5 GB.

The e-commerce department at Pennyworth's develops an Azure logic app that captures trending Twitter feeds referencing the company's products and pushes the products to Azure Event Hubs.

Planned Changes and Requirements

Pennyworth's plans to implement the following changes:

• Load the sales transaction dataset to Azure Synapse Analytics.

• Integrate on-premises data stores with Azure Synapse Analytics by using SSIS packages.

- Use Azure Synapse Analytics to analyze Twitter feeds to assess customer sentiments about products.

Sales Transaction Dataset Requirements

Pennyworth's identifies the following requirements for the sales transaction dataset:

- Partition data that contains sales transaction records. Partitions must be designed to provide efficient loads by month. Boundary values must belong to the partition on the right.

- Ensure that queries joining and filtering sales transaction records based on product ID complete as quickly as possible.

- Implement a surrogate key to account for changes to the retail store addresses.

- Ensure that data storage costs and performance are predictable.

- Minimize how long it takes to remove old records.

Customer Sentiment Analytics Requirements

Pennyworth's identifies the following requirements for customer sentiment analytics:

- Allow Pennyworth's users to use PolyBase in an Azure Synapse Analytics dedicated SQL pool to query the content of the data records that host the Twitter feeds.

- Data must be protected by using row-level security (RLS). The users must be authenticated by using their own Azure AD credentials.

- Maximize the throughput of ingesting Twitter feeds from Event Hubs to Azure Storage without purchasing additional throughput or capacity units.

- Store Twitter feeds in Azure Storage by using Event Hubs Capture. The feeds will be converted into Parquet files.

- Ensure that the data store supports Azure AD-based access control down to the object level.

- Minimize administrative effort to maintain the Twitter feed data records.

- Purge Twitter feed data records that are older than two years.

Data Integration Requirements

Pennyworth's identifies the following requirements for data integration:

• Use an Azure service that leverages the existing SSIS packages to ingest on-premises data into datasets stored in a dedicated SQL pool of Azure Synapse Analytics and transform the data.

• Identify a process to ensure that changes to the ingestion and transformation activities can be version-controlled and developed independently by multiple data engineers.

The Ask:

Alfred places a great importance on this project and asks you to work closely with the team to implement the surrogate key for the retail store table. The solution must meet the sales transaction dataset requirements.

Which of the following should you advise the team to create?

- ○ A table that has a `FOREIGN KEY` constraint.
- ○ A system-versioned temporal table.
- ○ A table that has an `IDENTITY` property. **(Correct)**
- ○ A user-defined `SEQUENCE` object.

Explanation

The best way to implement a surrogate key to account for changes to the retail store addresses is to create a table that has an `IDENTITY` property.

A surrogate key on a table is a column with a unique identifier for each row. The key is not generated from the table data. Data modelers like to create surrogate keys on their tables when they design data warehouse models. You can use the `IDENTITY` property to achieve this goal simply and effectively without affecting load performance.

Using IDENTITY to create surrogate keys using dedicated SQL pool in Azure Synapse Analytics

What is a surrogate key?

A surrogate key on a table is a column with a unique identifier for each row. The key is not generated from the table data. Data modelers like to create surrogate keys on their tables when they design data warehouse models. You can use the IDENTITY property to achieve this goal simply and effectively without affecting load performance.

Note: In Azure Synapse Analytics, the IDENTITY value increases on its own in each distribution and does not overlap with IDENTITY values in other distributions. The IDENTITY value in Synapse is not guaranteed to be unique if the user explicitly inserts a duplicate value with "SET IDENTITY_INSERT ON" or reseeds IDENTITY. For details, see CREATE TABLE (Transact-SQL) IDENTITY (Property).

Creating a table with an IDENTITY column

The IDENTITY property is designed to scale out across all the distributions in the dedicated SQL pool without affecting load performance. Therefore, the implementation of IDENTITY is oriented toward achieving these goals.

You can define a table as having the IDENTITY property when you first create the table by using syntax that is similar to the following statement:

```
SQL

CREATE TABLE dbo.T1

(    C1 INT IDENTITY(1,1) NOT NULL

,    C2 INT NULL

)

WITH

(   DISTRIBUTION = HASH(C2)

,   CLUSTERED COLUMNSTORE INDEX

)

;
```

In the preceding example, two rows landed in distribution 1. The first row has the surrogate value of 1 in column `C1`, and the second row has the surrogate value of 61. Both of these values were generated by the IDENTITY property. However, the allocation of the values is not contiguous. This behavior is by design.

Skewed data

The range of values for the data type are spread evenly across the distributions. If a distributed table suffers from skewed data, then the range of values available to the datatype can be exhausted prematurely. For example, if all the data ends up in a single distribution, then effectively the table has access to only one-sixtieth of the values of the data type. For this reason, the IDENTITY property is limited to `INT` and `BIGINT` data types only.

SELECT..INTO

When an existing IDENTITY column is selected into a new table, the new column inherits the IDENTITY property, unless one of the following conditions is true:

- The SELECT statement contains a join.

- Multiple SELECT statements are joined by using UNION.

- The IDENTITY column is listed more than one time in the SELECT list.

- The IDENTITY column is part of an expression.

If any one of these conditions is true, the column is created NOT NULL instead of inheriting the IDENTITY property.

CREATE TABLE AS SELECT

`CREATE TABLE AS SELECT` (CTAS) follows the same SQL Server behavior that's documented for SELECT..INTO. However, you can't specify an IDENTITY property in the column definition of the `CREATE TABLE` part of the statement. You also can't use the IDENTITY function in the `SELECT` part of the CTAS. To populate a table, you need to use `CREATE TABLE` to define the table followed by `INSERT..SELECT` to populate it.

Explicitly inserting values into an IDENTITY column

Dedicated SQL pool supports `SET IDENTITY_INSERT <your table> ON|OFF` syntax. You can use this syntax to explicitly insert values into the IDENTITY column.

Many data modelers like to use predefined negative values for certain rows in their dimensions. An example is the -1 or "unknown member" row.

The next script shows how to explicitly add this row by using SET IDENTITY_INSERT:

SQL

```
SET IDENTITY_INSERT dbo.T1 ON;


INSERT INTO dbo.T1
```

```
(   C1
,   C2
)
VALUES (-1,'UNKNOWN')
;

SET IDENTITY_INSERT dbo.T1 OFF;

SELECT     *
FROM    dbo.T1
;
```

Loading data

The presence of the IDENTITY property has some implications to your data-loading code. This section highlights some basic patterns for loading data into tables by using IDENTITY.

To load data into a table and generate a surrogate key by using IDENTITY, create the table and then use INSERT..SELECT or INSERT..VALUES to perform the load.

The following example highlights the basic pattern:

```
SQL
--CREATE TABLE with IDENTITY
CREATE TABLE dbo.T1
(    C1 INT IDENTITY(1,1)
,    C2 VARCHAR(30)
)
WITH
(   DISTRIBUTION = HASH(C2)
,   CLUSTERED COLUMNSTORE INDEX
)
;

--Use INSERT..SELECT to populate the table from an external table
```

```
INSERT INTO dbo.T1
(C2)
SELECT     C2
FROM    ext.T1
;

SELECT *
FROM   dbo.T1
;

DBCC PDW_SHOWSPACEUSED('dbo.T1');
```

Note: It's not possible to use `CREATE TABLE AS SELECT` currently when loading data into a table with an IDENTITY column.

System views

You can use the sys.identity_columns catalog view to identify a column that has the IDENTITY property.

To help you better understand the database schema, this example shows how to integrate sys.identity_column` with other system catalog views:

```
SQL
SELECT  sm.name
,       tb.name
,       co.name
,       CASE WHEN ic.column_id IS NOT NULL
             THEN 1
        ELSE 0
        END AS is_identity
FROM        sys.schemas AS sm
JOIN        sys.tables  AS tb           ON  sm.schema_id = tb.schema_id
JOIN        sys.columns AS co           ON  tb.object_id = co.object_id
LEFT JOIN   sys.identity_columns AS ic  ON  co.object_id = ic.object_id
                                        AND co.column_id = ic.column_id
```

```
WHERE   sm.name = 'dbo'
AND     tb.name = 'T1'
;
```

Limitations

The IDENTITY property can't be used:

- When the column data type is not INT or BIGINT
- When the column is also the distribution key
- When the table is an external table

The following related functions are not supported in dedicated SQL pool:

- IDENTITY()
- @@IDENTITY
- SCOPE_IDENTITY
- IDENT_CURRENT
- IDENT_INCR
- IDENT_SEED

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-tables-identity

Question 23: Skipped

By default, the Azure Data Factory user interface experience (UX) authors directly against the data factory service.

To provide a better authoring experience, Azure Data Factory allows you to configure a Git repository with either Azure Repos or GitHub. Git is a version control system that allows for easier change tracking and collaboration.

True or False: Configuring a git repository allows you to save changes, letting you only publish when you have tested your changes to your satisfaction.

- ○ True (Correct)
- ○ False

Explanation

By default, the Azure Data Factory user interface experience (UX) authors directly against the data factory service. This experience has the following limitations:

- The Data Factory service doesn't include a repository for storing the JSON entities for your changes. The only way to save changes is via the **Publish All** button and all changes are published directly to the data factory service.

- The Data Factory service isn't optimized for collaboration and version control.

- The Azure Resource Manager template required to deploy Data Factory itself is not included.

To provide a better authoring experience, Azure Data Factory allows you to configure a Git repository with either Azure Repos or GitHub. Git is a version control system that allows for easier change tracking and collaboration.

Advantages of Git integration

Below is a list of some of the advantages git integration provides to the authoring experience:

- **Source control:** As your data factory workloads become crucial, you would want to integrate your factory with Git to leverage several source control benefits like the following:

    - Ability to track/audit changes.

- Ability to revert changes that introduced bugs.

- **Partial saves:** When authoring against the data factory service, you can't save changes as a draft and all publishes must pass data factory validation. Whether your pipelines are not finished or you simply don't want to lose changes if your computer crashes, git integration allows for incremental changes of data factory resources regardless of what state they are in. **Configuring a git repository allows you to save changes, letting you only publish when you have tested your changes to your satisfaction.**

- **Collaboration and control:** If you have multiple team members contributing to the same factory, you may want to let your teammates collaborate with each other via a code review process. You can also set up your factory such that not every contributor has equal permissions. Some team members may only be allowed to make changes via Git and only certain people in the team are allowed to publish the changes to the factory.

- **Better CI/CD:** If you are deploying to multiple environments with a continuous delivery process, git integration makes certain actions easier. Some of these actions include:

  - Configure your release pipeline to trigger automatically as soon as there are any changes made to your 'dev' factory.

  - Customize the properties in your factory that are available as parameters in the Resource Manager template. It can be useful to keep only the required set of properties as parameters, and have everything else hard coded.

- **Better Performance:** An average factory with git integration loads 10 times faster than one authoring against the data factory service. This performance improvement is because resources are downloaded via Git.

Connect to a Git repository

There are different ways to connect a Git repository to your data factory for both Azure Repos and GitHub. After you connect to a Git repository, you can view and manage your configuration in the management hub under **Git configuration** in the **Source control** section.

Configuration method 1: Home page

In the Azure Data Factory home page, select **Set up Code Repository**.

Configuration method 2: Authoring canvas

In the Azure Data Factory UX authoring canvas, select the **Data Factory** drop-down menu, and then select **Set up Code Repository**.

Configuration method 3: Management hub

Go to the management hub in the Azure Data Factory UX. Select **Git configuration** in the **Source control** section. If you have no repository connected, click **Set up code repository**.

https://docs.microsoft.com/en-us/azure/data-factory/source-control

Question 24: **Skipped**

Scenario: You are working as a consultant at Avengers Security and the IT team has developed a data ingestion process to import data to a Microsoft Azure SQL Data Warehouse. They are using an Azure Data Lake Gen 2 storage account to store the data to be ingested. The data to be ingested resides in parquet files.

Required: Load the data from the Azure Data Lake Gen 2 storage account into the Azure SQL Data Warehouse.

The Avengers IT team has proposed the following solution:

1. Create an external data source pointing to the Azure storage account

2. Create an external file format and external table using the external data source

3. Load the data using the `INSERT ... SELECT` statement

Will the solution proposed by the Avengers IT team meet the requirement?

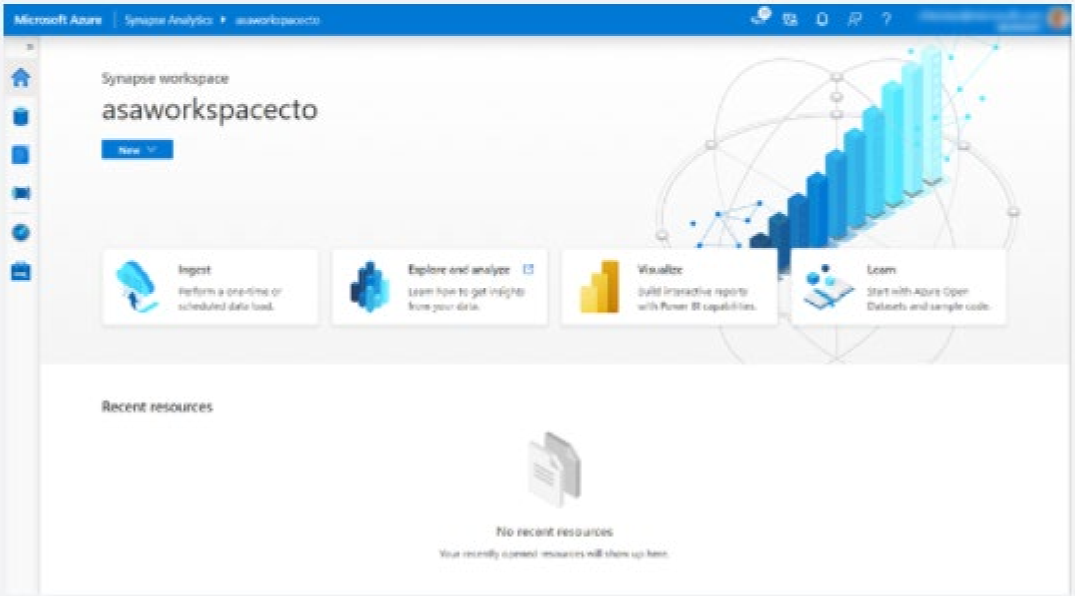
- ○ Yes
- ○ No (Correct)

Explanation

The proposed solution will not meet the requirement. They need to create an external file format and external table using the external data source. To load the data, use the `CREATE TABLE ... AS SELECT` statement.

Use polybase by defining external tables

Using Transact-SQL, you can use PolyBase to access files that are located directly on Azure Storage as if they were structured tables within your SQL Pool. You define an **external data source** pointing to the location of the file or the folder the files reside in, the external file format, which can be GZip compressed delimited text, ORC, Parquet or JSON, and then the external table with the column attributes that map to the structure from the external files.


Microsoft Azure
Synapse Analytics
asaworkspacecto
Synapse workspace
asaworkspacecto
New
Ingest
Perform a one-time or scheduled data load.
Explore and analyze
Learn how to get insights from your data.
Visualize
Build interactive reports with Power BI capabilities.
Learn
Start with Azure Open Datasets and sample code.
Recent resources
No recent resources
Your recently opened resources will show up here.

Create an import database

The first step in using PolyBase is to create a database-scoped credential that secures the credentials to the blob storage. Create a master key first, and then use this key to encrypt the database-scoped credential named **AzureStorageCredential**.

1. Paste the following code into the query window. Replace the `SECRET` value with the access key you retrieved in the previous exercise.

```
SQL
CREATE MASTER KEY;


CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
    IDENTITY = 'demodwStorage',
    SECRET = 'THE-VALUE-OF-THE-ACCESS-KEY' -- put key1's value here
;
```

2. Select **Run** to run the query. It should report `Query succeeded: Affected rows: 0.`

Create an external data source connection

Use the database-scoped credential to create an external data source named **AzureStorage**. Note the location URL point to the container named **data-files** that you created in the blob storage. The type **Hadoop** is used for both Hadoop-based and Azure Blob storage-based access.

1. Paste the following code into the query window. Replace the `LOCATION` value with your correct value from the previous exercise.

```
SQL
CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
    TYPE = HADOOP,
    LOCATION = 'wasbs://data-files@demodwstorage.blob.core.windows.net',
    CREDENTIAL = AzureStorageCredential
);
```

2. Select **Run** to run the query. It reports `Query succeeded: Affected rows: 0.`.

Define the import file format

Define the external file format named **TextFile**. This name indicates to PolyBase that the format of the text file is **DelimitedText** and the field terminator is a comma.

1. Paste the following code into the query window.

```
SQL
CREATE EXTERNAL FILE FORMAT TextFile
WITH (
    FORMAT_TYPE = DelimitedText,
    FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
);
```

2. Select **Run** to run the query. It reports `Query succeeded: Affected rows: 0.`.

Create a temporary table

Create an external table named `dbo.temp` with the column definition for your table. At the bottom of the query, use a `WITH` clause to call the data source definition named **AzureStorage**, as previously defined, and the file format named **TextFile**, as previously defined. The location denotes that the files for the load are in the root folder of the data source.

Note: External tables are in-memory tables that don't persist onto the physical disk. External tables can be queried like any other table.

The table definition must match the fields defined in the input file. There are 12 defined columns, with data types that match the input file data.

1. Add the following code into the Visual Studio window underneath the previous code.

```
SQL
-- Create a temp table to hold the imported data
CREATE EXTERNAL TABLE dbo.Temp (
    [Date] datetime2(3) NULL,
    [DateKey] decimal(38, 0) NULL,
    [MonthKey] decimal(38, 0) NULL,
    [Month] nvarchar(100) NULL,
    [Quarter] nvarchar(100) NULL,
    [Year] decimal(38, 0) NULL,
    [Year-Quarter] nvarchar(100) NULL,
    [Year-Month] nvarchar(100) NULL,
    [Year-MonthKey] nvarchar(100) NULL,
    [WeekDayKey] decimal(38, 0) NULL,
    [WeekDay] nvarchar(100) NULL,
    [Day Of Month] decimal(38, 0) NULL
)
WITH (
    LOCATION='../',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

2. Select **Run** to run the query. It takes a few seconds to complete and reports `Query succeeded: Affected rows: 0.`.

Create a destination table

Create a physical table in the Azure Synapse Analytics database. In the following example, you create a table named `dbo.StageDate`. The table has a clustered column store index defined on all the columns. It uses a table geometry of `round_robin` by design because `round_robin` is the best table geometry to use for loading data.

1. Paste the following code into the query window.

```
SQL

-- Load the data from Azure Blob storage to Azure Synapse Analytics

CREATE TABLE [dbo].[StageDate]

WITH (

    CLUSTERED COLUMNSTORE INDEX,

    DISTRIBUTION = ROUND_ROBIN

)

AS

SELECT * FROM [dbo].[Temp];
```

2. Select **Run** to run the query. It takes a few seconds to complete and reports `Query succeeded: Affected rows: 0.`.

Add statistics onto columns to improve query performance

As an optional step, create statistics on columns that feature in queries to improve the query performance against the table.

1. Paste the following code into the query window.

```
SQL

-- Create statistics on the new data

CREATE STATISTICS [DateKey] on [StageDate] ([DateKey]);

CREATE STATISTICS [Quarter] on [StageDate] ([Quarter]);

CREATE STATISTICS [Month] on [StageDate] ([Month]);
```

2. Select **Run** to run the query. It reports `Query succeeded: Affected rows: 0.`.

You've loaded your first staging table in Azure Synapse Analytics. From here, you can write further Transact-SQL queries to perform transformations into dimension and fact tables. Try it out by querying the `StageDate` table in the query explorer or in another query tool. Refresh the view on the left to see the new table or tables that you created. Reuse the previous steps in a persistent SQL script to load additional data, as necessary.

https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-load-from-azure-data-lake-store

Question 25: **Skipped**

Microsoft Azure Storage is a managed service that provides durable, secure, and scalable storage in the cloud. You can create an Azure storage account using the Azure Portal, Azure PowerShell, or Azure CLI. Azure Storage provides three distinct account options with different pricing and features supported.

Which of the Azure Storage account options is best described by:

“Support all of the latest features for blobs, files, queues, and tables. Pricing has been designed to deliver the lowest per gigabyte prices.”

- ○ Blob storage accounts
- ○ Block
- ○ Queue
- ○ GPv2 **(Correct)**
- ○ Page
- ○ GPv1
- ○ Append

Explanation

Create a storage account

You can create an Azure storage account using the Azure portal, Azure PowerShell, or Azure CLI. Azure Storage provides three distinct account options with different pricing and features supported.

General-purpose v1 (GPv1)

General-purpose v1 (GPv1) accounts provide access to all Azure Storage services but may not have the latest features or the lowest per gigabyte pricing. For example, cool storage and archive storage are not supported in GPv1. Pricing is lower for GPv1 transactions, so workloads with high churn or high read rates may benefit from this account type.

General-purpose v2 (GPv2)

General-purpose v2 (GPv2) accounts are storage accounts that support all of the latest features for blobs, files, queues, and tables. Pricing for GPv2 accounts has been designed to deliver the lowest per gigabyte prices.

Blob storage accounts

A legacy account type, blob storage accounts support all the same block blob features as GPv2, but they are limited to supporting only block and append blobs. Pricing is broadly similar to pricing for general-purpose v2 accounts.

https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview

---

Question 26: Skipped

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Transactional databases are often called [?] systems. These systems commonly support lots of users, have quick response times, and handle large volumes of data.

- ○ OLTP (Online Transaction Processing) (Correct)
- ○ Extract, load, and transform (ELT)
- ○ Extract, transform, and load (ETL)
- ○ Automated Data Processing Structured (ADPS)
- ○ Atomicity, Consistency, Isolation, and Durability (ACID)
- ○ OLAP (Online Analytical Processing)

Explanation

A transaction is a logical group of database operations that execute together.

Here's the question to ask yourself regarding whether you need to use transactions in your application: Will a change to one piece of data in your dataset impact another? If the answer is yes, then you'll need support for transactions in your database service.

Transactions are often defined by a set of four requirements, referred to as ACID guarantees. ACID stands for **A**tomicity, **C**onsistency, **I**solation, and **D**urability:

• **Atomicity** means a transaction must execute exactly once and must be atomic; either all of the work is done, or none of it is. Operations within a transaction usually share a common intent and are interdependent.

• **Consistency** ensures that the data is consistent both before and after the transaction.

• **Isolation** ensures that one transaction is not impacted by another transaction.

• **Durability** means that the changes made due to the transaction are permanently saved in the system. Committed data is saved by the system so that even in the event of a failure and system restart, the data is available in its correct state.

When a database offers ACID guarantees, these principles are applied to any transactions in a consistent manner.

OLTP vs OLAP

Transactional databases are often called OLTP (Online Transaction Processing) systems. OLTP systems commonly support lots of users, have quick response times, and handle large volumes of data. They are also highly available (meaning they have very minimal downtime), and typically handle small or relatively simple transactions.

On the contrary, OLAP (Online Analytical Processing) systems commonly support fewer users, have longer response times, can be less available, and typically handle large and complex transactions.

The terms OLTP and OLAP aren't used as frequently as they used to be, but understanding them makes it easier to categorize the needs of your application.

Now that you're familiar with transactions, OLTP, and OLAP, let's walk through each of the data sets in the online retail scenario, and determine the need for transactions.

https://www.guru99.com/oltp-vs-olap.html

Question 27: **Skipped**

Because the Databricks API is declarative, a large number of optimizations are available to us. Among the most powerful components of Spark are Spark SQL. At its core lies the Catalyst optimizer.

When you execute code, Spark SQL uses Catalyst's general tree transformation framework in four phases, as shown below:

1. analyzing a logical plan to resolve references

2. logical plan optimization

3. physical planning

4. code generation to compile parts of the query to Java bytecode

In the physical planning phase, Catalyst may generate multiple plans and compare them based on [?].

- ○ Cost (Correct)
- ○ Region
- ○ Permissions
- ○ Rules

Explanation

Because the Databricks API is declarative, a large number of optimizations are available to us.

Some of the examples include:

• Optimizing data type for storage

• Rewriting queries for performance

• Predicate push downs

Among the most powerful components of Spark are Spark SQL. At its core lies the Catalyst optimizer. This extensible query optimizer supports both rule-based and cost-based optimization.

When you execute code, Spark SQL uses Catalyst's general tree transformation framework in four phases, as shown below:

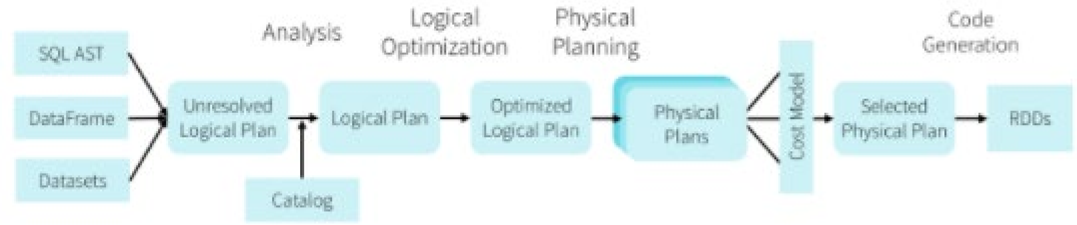
1. analyzing a logical plan to resolve references

2. logical plan optimization

3. physical planning

4. code generation to compile parts of the query to Java bytecode

In the physical planning phase, Catalyst may generate multiple plans and compare them based on cost. All other phases are purely rule-based.



Catalyst is based on functional programming constructs in Scala and designed with these key two purposes:

• Easily add new optimization techniques and features to Spark SQL

• Enable external developers to extend the optimizer (e.g. adding data source specific rules, support for new data types, etc.)

https://data-flair.training/blogs/spark-sql-optimization/

Question 28: Skipped

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

You can use *account-level* SAS to allow access to anything that a service-level SAS can allow, plus additional resources and abilities. You'd use this type of SAS, for example, … [?] (Select all that apply)

- ○ to allow the ability to create file systems.
  (Correct)
- ○ to allow an app to download a file.
- ○ None of the listed options.
- ○ to allow an app to retrieve a list of files in a file system.

Explanation

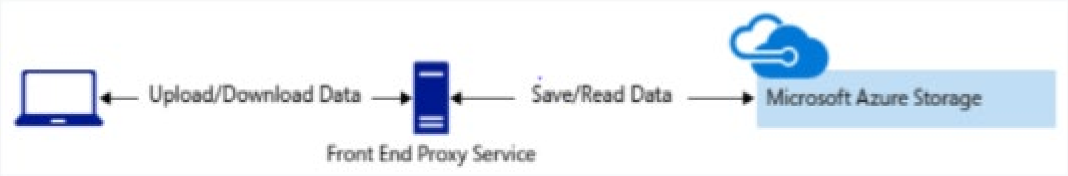
Types of shared access signatures

You can use a *service-level* SAS to allow access to specific resources in a storage account. You'd use this type of SAS, for example, to allow an app to retrieve a list of files in a file system, or to download a file.

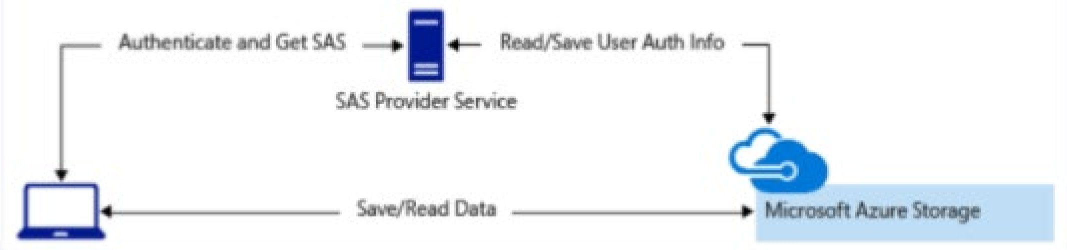
Use an *account-level* SAS to allow access to anything that a service-level SAS can allow, plus additional resources and abilities. For example, you can use an account-level SAS to allow the ability to create file systems.

You'd typically use a SAS for a service where users read and write their data to your storage account. Accounts that store user data have two typical designs:

• Clients upload and download data through a front-end proxy service, which performs authentication. This front-end proxy service has the advantage of allowing validation of business rules. But, if the service must handle large amounts of data or high-volume transactions, you might find it complicated or expensive to scale this service to match demand.



• A lightweight service authenticates the client, as needed. Next, it generates a SAS. After receiving the SAS, the client can access storage account resources directly. The SAS defines the client's permissions and access interval. It reduces the need to route all data through the front-end proxy service.



https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

Question 29: **Skipped**

Scenario: You have been assigned to a new project and your first task is to initialize the Blob Storage client library within an application.

Which of the following can be used to do this?

- ○

  A globally-unique identifier (GUID) that represents the application.
- ○

  The Azure Storage account connection string.

  (Correct)
- ○

  The Azure Storage account datacentre and location identifiers.
- ○

  An Azure username and password.

Explanation

A storage account connection string contains all the information needed to connect to Blob storage, most importantly the account name and the account key.

https://docs.microsoft.com/en-us/azure/storage/common/storage-configure-connection-string

---

Question 30: Skipped

What can cause a slower performance on join or shuffle jobs?

- ○ Bucketing
- ○ Data skew (Correct)
- ○ Enablement of autoscaling
- ○ Use the cache option

Explanation

The data skew is one of the most common reasons why your Apache Spark job is underperforming. Data skew can cause a slower performance on join or shuffle jobs due to asymmetry in your job data.

Spark is a distributed system, and as such, it divides the data into multiple pieces, called partitions, moves them into the different cluster nodes, and processes them in parallel. If one of these partitions happens to be much larger than others, the node processing it may experience the resource issues and slow down entire execution. This kind of data imbalance is called a data skew.

The size of the partitions depends on the factors, like partitioning configuration of the source files, the number of CPU cores and the nature of your query. The most common scenarios, involving the data skew problems, include the aggregation and join queries, where the grouping or joining field has unequally distributed keys (i.e. few keys have much more rows, than the remaining keys). In this scenario, Spark will send the rows with the same key to the same partition and cause data skew issues.

A traditional Apache Spark UI has some dashboards to determine data skew issues. In addition to that, Azure Synapse Analytics introduced nice data skew diagnosis tools.

https://www.mssqltips.com/sqlservertip/6747/azure-synapse-analytics-analyze-data-skew-issues/

Question 31: **Skipped**

Azure Data Factory provides a variety of methods for ingesting data, and also provides a range of methods to perform transformations.

These methods are:

• Mapping Data Flows

• Compute Resources

• SSIS Packages

Mapping Data Flows provides a number of different transformations types that enable you to modify data. They are broken down into the following categories:

• Schema modifier transformations

• Row modifier transformations

• Multiple inputs/outputs transformations

Which transformations type is best described by:

"A Sort transformation that orders the data."

- ○ Schema modifier transformations
- ○ Multiple inputs/outputs transformations
- ○ None of the listed options.
- ○ Row modifier transformations
  (Correct)

Explanation

Just as Azure Data Factory provides a variety of methods for ingesting data, it also provides a range of methods to perform transformations. You can pick a method that matches the skillsets of your team or takes advantage of existing technologies that you already have in your data estate. There is also the opportunity to perform transformations without writing code at all using the Mapping Data Flow.

Transforming data using Mapping Data Flow

Mapping Data Flows provide an environment for building a wide range of data transformations visually without the need to use code. The resulting data flows that are created are subsequently executed on scaled-out Apache Spark clusters that are automatically provisioned when you execute the Mapping Data Flow. Mapping Data Flows also provides the capability to monitor the execution of the transformations so that you can view how the transformations are progressing, or to understand any errors that may occur

Mapping Data Flows provides a number of different transformations types that enable you to modify data. They are broken down into the following categories:

Category Name: Schema modifier transformations

Description: These types of transformations will make a modification to a sink destination by creating new columns based on the action of the transformation. An example of this is the Derived Column transformation that will create a new column based on the operations performed on existing column.

Category Name: Row modifier transformations

Description: These types of transformations impact how the rows are presented in the destination. An example of this is a Sort transformation that orders the data.

Category Name: Multiple inputs/outputs transformations

Description: These types of transformations will generate new data pipelines or merge pipelines into one. An example of this is the Union transformation that combines multiple data streams.

https://docs.microsoft.com/en-us/azure/data-factory/transform-data

Question 32: **Skipped**

Which statement about the Azure Databricks Data Plane is true?

- ○ The Data Plane is hosted within a Microsoft-managed subscription.
- ○ The Data Plane is where you manage Key Vault itself and it is the interface used to create and delete vaults.
- ○ The Data Plane is hosted within the client subscription and is where all data is processed and stored.
  (Correct)
- ○ The Data Plane contains the Cluster Manager and coordinates data processing jobs.

Explanation

All data is processed by clusters hosted within the client Azure subscription and data is stored within Azure Blob storage and any connected Azure services within this portion of the platform architecture.

https://docs.microsoft.com/en-us/azure/key-vault/general/security-overview

---

Question 33: **Skipped**

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

Many business application architectures separate transactional and analytical processing into separate systems with data stored and processed on separate infrastructures. [?] systems are optimized for dealing with discrete system or user requests immediately and responding as quickly as possible.

- ○ OLTP (Correct)
- ○ ETL
- ○ OLAP
- ○ ELT
- ○ ADPS

Explanation

Many business application architectures separate transactional and analytical processing into separate systems with data stored and processed on separate infrastructures. These infrastructures are commonly referred to as OLTP (online transaction processing) systems working with operational data, and OLAP (online analytical processing) systems working with historical data, with each system is optimized for their specific task.

OLTP systems are optimized for dealing with discrete system or user requests immediately and responding as quickly as possible.

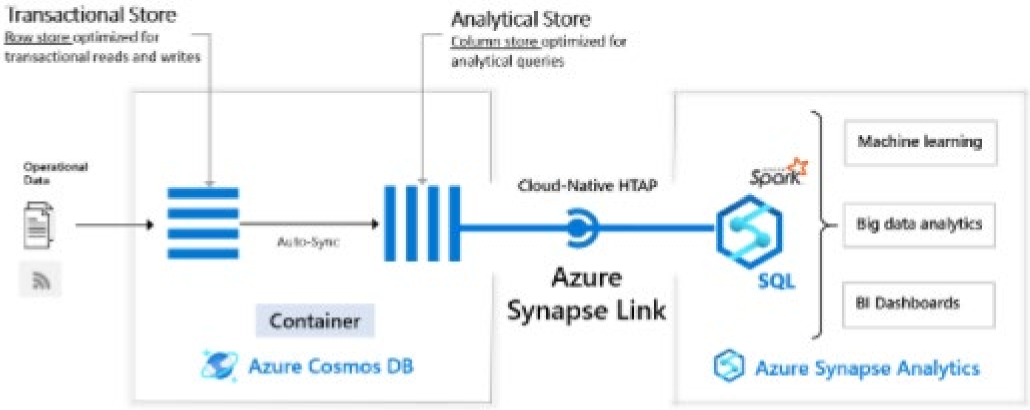
OLAP systems are optimized for the analytical processing, ingesting, synthesizing, and managing large sets of historical data. The data processed by OLAP systems largely originates from OLTP systems and needs to be loaded into the OLTP systems by means of batch processes commonly referred to as ETL (Extract, Transform, and Load) jobs.

Due to their complexity and the need to physically copy large amounts of data, this creates a delay in data being available to provide insights by way of the OLAP systems.

As more and more businesses move to digital processes, they increasingly recognize the value of being able to respond to opportunities by making faster and well-informed decisions. HTAP (Hybrid Transactional/Analytical processing) enables business to run advanced analytics in near-real-time on data stored and processed by OLTP systems.

Azure Synapse Link for Azure Cosmos DB

Azure Synapse Link for Azure Cosmos DB is a cloud-native HTAP capability that enables you to run near-real-time analytics over operational data stored in Azure Cosmos DB. Azure Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.



Azure Cosmos DB provides both a transactional store optimized for transactional workloads and an analytical store optimized for analytical workloads and a fully managed autosync process to keep the data within these stores in sync.

Azure Synapse Analytics provides both a SQL Serverless query engine for querying the analytical store using familiar T-SQL and an Apache Spark query engine for leveraging the analytical store using your choice of Scala, Java, Python or SQL and provides a user-friendly notebook experience.

Together Azure Cosmos DB and Synapse Analytics enable organizations to generate and consume insights from their operational data in near-real time, using the query and analytics tools of their choice. All of this is achieved without the need for complex ETL pipelines and without affecting the performance of their OLTP systems using Azure Cosmos DB.

https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link

Question 34: **Skipped**

True or False: Access keys are the easiest approach to authenticating access to a storage account which provide full access to anything in the storage account, similar to a root password on a computer.

- ○ False
- ○ True **(Correct)**

Explanation

Shared Access Signatures (SAS)

Access keys are the easiest approach to authenticating access to a storage account. However they provide full access to anything in the storage account, similar to a root password on a computer.

Storage accounts offer a separate authentication mechanism called *shared access signatures* that support expiration and limited permissions for scenarios where you need to grant limited access. You should use this approach when you are allowing other users to read and write data to your storage account. There are links to our documentation on this advanced topic at the end of the module.

https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

Question 35: Skipped

Which Azure Synapse Analytics component enables you to perform Hybrid Transactional and Analytical Processing?

- ○ Azure Data Warehouse
- ○ Azure Synapse Pipeline
- ○ Azure Synapse Spark pools
- ○ Azure Stream Analytics
- ○ None of the listed options
- ○ Azure Data Explorer
- ○ Azure Synapse Link (Correct)
- ○ Azure Synapse Studio

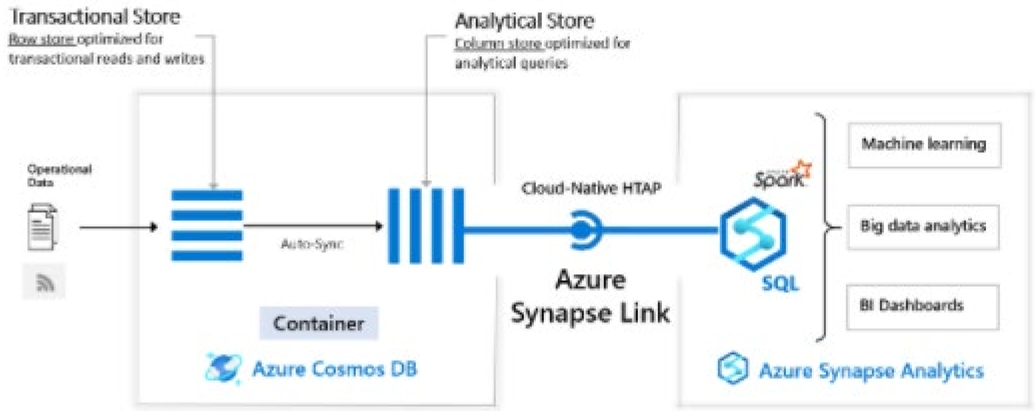
Explanation

Azure Synapse Link is the component that enables Hybrid Transactional and Analytical Processing.

Azure Synapse Link for Azure Cosmos DB is a cloud-native hybrid transactional and analytical processing (HTAP) capability that enables you to run near real-time analytics over operational data in Azure Cosmos DB. Azure Synapse Link creates a tight seamless integration between Azure Cosmos DB and Azure Synapse Analytics.

Using Azure Cosmos DB analytical store, a fully isolated column store, Azure Synapse Link enables no Extract-Transform-Load (ETL) analytics in Azure Synapse Analytics against your operational data at scale. Business analysts, data engineers and data scientists can now use Synapse Spark or Synapse SQL interchangeably to run near real-time business intelligence, analytics, and machine learning pipelines. You can achieve this without impacting the performance of your transactional workloads on Azure Cosmos DB.

The following image shows the Azure Synapse Link integration with Azure Cosmos DB and Azure Synapse Analytics:



https://docs.microsoft.com/en-us/azure/cosmos-db/synapse-link

Question 36: Skipped

Within creating a notebook, you need to specify the pool that needs to be attached to the notebook that is, a SQL or Spark pool. When it comes to the languages, a notebook has to be set with a primary language.

True or False: It is possible to use multiple languages in one notebook.

- ○ True (Correct)
- ○ False

Explanation

Within creating a notebook, you need to specify the pool that needs to be attached to the notebook that is, a SQL or Spark pool. When it comes to the languages, a notebook has to be set with a primary language.

The primary languages available within the notebook environment are:

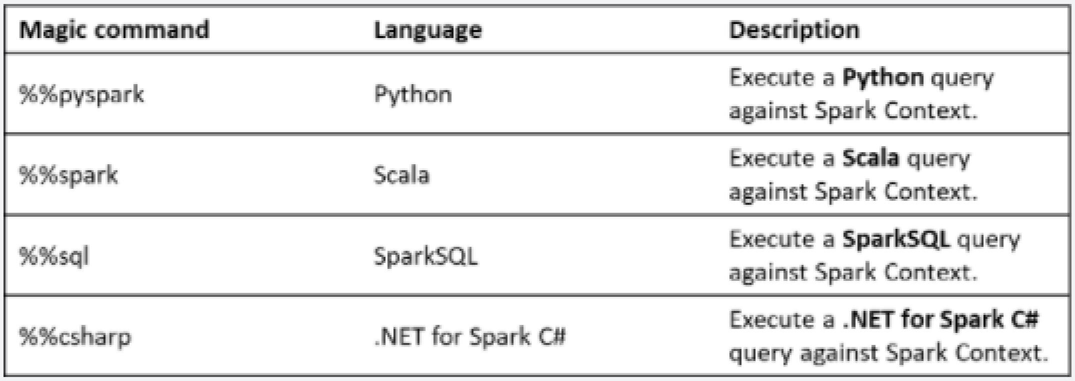
• PySpark (Python)

• Spark (Scala)

• .NET Spark (C#)

• Spark SQL

However, it is possible to use multiple languages in one notebook by specifying the language using a magic command at the beginning of a cell. The following table lists the magic commands to switch cell languages:



Magic command	Language	Description
%%pyspark	Python	Execute a **Python** query against Spark Context.
%%spark	Scala	Execute a **Scala** query against Spark Context.
%%sql	SparkSQL	Execute a **SparkSQL** query against Spark Context.
%%csharp	.NET for Spark C#	Execute a **.NET for Spark C#** query against Spark Context.

It is not possible to reference data or variables directly across different languages in a Synapse Studio notebook. In Spark, it is possible to reference a temporary table across languages.

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-development-using-notebooks?tabs=classical

Question 37: Skipped

Activities within Azure Data Factory define the actions that will be performed on the data and there are three categories including:

- Data movement activities

- Data transformation activities

- Control activities

When using JSON notation, the activities section can have one or more activity defined within it.

They have the following top-level structure:

```
JSON
{
"name": "Execution Activity Name",
"description": "description",
"type": "<ActivityType>",
"typeProperties":
{
},
"linkedServiceName": "MyLinkedService",
"policy":
{
},
"dependsOn":
{
}
}
```

Which of the JSON properties are required for HDInsight? (Select all that apply)

- □
  dependsOn
- □
  typeProperties
- □
  description
  (Correct)
- □
  type
  (Correct)
- □
  policy
- □
  linkedServiceName
  (Correct)
- □
  name
  (Correct)

Explanation

Activities within Azure Data Factory define the actions that will be performed on the data and there are three categories including:

• Data movement activities

• Data transformation activities

• Control activities

Activities and pipelines

Defining activities

When using JSON notation, the activities section can have one or more activities defined within it. There are two main types of activities: Execution and Control Activities. Execution (also known as Compute) activities include data movement and data transformation activities. They have the following top-level structure:

```
JSON
{
"name": "Execution Activity Name",
"description": "description",
"type": "<ActivityType>",
"typeProperties":
{
},
"linkedServiceName": "MyLinkedService",
"policy":
{
},
"dependsOn":
{
}
}
```

The following describes properties in the above JSON:

Property: name

Name of the activity.

Required: Yes

Property: description

Text describing what the activity or is used for.

Required: Yes

Property: type

Defines the type of the activity.

Required: Yes

Property: linkedServiceName

Name of the linked service used by the activity.

Required: Yes for HDInsight, Machine Learning Batch Scoring Activity and Stored Procedure Activity

Property: typeProperties

Properties in the typeProperties section depend on each type of activity.

Required: No

Property: policy

Policies that affect the run-time behaviour of the activity. This property includes timeout and retry behaviour.

Required: No

Property: dependsOn

This property is used to define activity dependencies, and how subsequent activities depend on previous activities.

Required: No

https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities

Question 38: Skipped

Identify the missing word(s) in the following sentence within the context of Microsoft Azure.

[?] is typically used to automate the process of extracting, transforming, and loading the data through a batch process against structured and unstructured data sources.

- ○ Azure Stored Procedure
- ○ Azure PowerShell
- ○ Azure Functions
- ○ Azure Orchestrator
- ○ Azure Conductor
- ○ Azure Designer
- ○ Azure Data Factory (Correct)

Explanation

Modern Data Warehouse workloads:

A Modern Data Warehouse is a centralized data store that provides descriptive analytics and decision support services across the whole enterprise using structured, unstructured, or streaming data sources. Data flows into the warehouse from multiple transactional systems, relational databases, and other data sources on a periodic basis. The stored data is used for historical and trend analysis reporting. The data warehouse acts as a central repository for many subject areas and contains the "single source of truth."

Azure Data factory is typically used to automate the process of extracting, transforming, and loading the data through a batch process against structured and unstructured data sources.

Advanced Analytical Workloads

You can perform advanced analytics in the form of predictive or preemptive analytics using a range of Azure data platform services. Azure Data Factory provides the integration from source systems into a Data Lake store, and can initiate compute resources such as Azure Databricks, or HDInsight to use the data to perform the advanced analytical work

https://cloudblogs.microsoft.com/industry-blog/en-gb/technetuk/2020/08/25/data-orchestration-with-azure-data-factory/

Question 39: Skipped

Azure Data Lake Storage combines a file system with a storage platform to help you quickly identify insights into your data. Data Lake Storage Gen2 builds on Azure Blob storage capabilities to optimize it specifically for analytics workloads.

You can set permissions at a directory level or file level for the data stored within the data lake. This security is configurable through technologies such as Hive and Spark, or utilities such as Azure Storage Explorer. All data that is stored is encrypted at rest by using either Microsoft or customer-managed keys.

Data Lake Storage Gen2 supports which of the following to enhance security?

- □ ACLs (Correct)
- □ AWS
- □ GRS
- □ HDFS
- □ POSIX (Correct)
- □ LRS

Explanation

A data lake is a repository of data that is stored in its natural format, usually as blobs or files. Azure Data Lake Storage is a comprehensive, scalable, and cost-effective data lake solution for big data analytics built into Azure.

Azure Data Lake Storage combines a file system with a storage platform to help you quickly identify insights into your data. Data Lake Storage Gen2 builds on Azure Blob storage capabilities to optimize it specifically for analytics workloads. This integration enables analytics performance, the tiering and data lifecycle management capabilities of Blob storage, and the high-availability, security, and durability capabilities of Azure Storage.

The variety and volume of data that is generated and analyzed today is increasing. Companies have multiple sources of data, from websites to Point of Sale (POS) systems, and more recently from social media sites to Internet of Things (IoT) devices. Each source provides an essential aspect of data that needs to be collected, analyzed, and potentially acted upon.

Benefits

Data Lake Storage Gen2 is designed to deal with this variety and volume of data at exabyte scale while securely handling hundreds of gigabytes of throughput. With this, you can use Data Lake Storage Gen2 as the basis for both real-time and batch solutions. Here is a list of additional benefits that Data Lake Storage Gen2 brings:

Hadoop compatible access

A benefit of Data Lake Storage Gen2 is that you can treat the data as if it's stored in a Hadoop Distributed File System. With this feature, you can store the data in one place and access it through compute technologies including Azure Databricks, Azure HDInsight, and Azure Synapse Analytics without moving the data between environments.

Security

Data Lake Storage Gen2 supports access control lists (ACLs) and Portable Operating System Interface (POSIX) permissions. You can set permissions at a directory level or file level for the data stored within the data lake. This security is configurable through technologies such as Hive and Spark, or utilities such as Azure Storage Explorer. All data that is stored is encrypted at rest by using either Microsoft or customer-managed keys.

Performance

Azure Data Lake Storage organizes the stored data into a hierarchy of directories and subdirectories, much like a file system, for easier navigation. As a result, data processing requires less computational resources, reducing both the time and cost.

Data redundancy

Data Lake Storage Gen2 takes advantage of the Azure Blob replication models that provide data redundancy in a single data centre with locally redundant storage (LRS), or to a secondary region by using the Geo-redundant storage (GRS) option. This feature ensures that your data is always available and protected if catastrophe strikes.

https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-overview

Question 40: **Skipped**

Which feature in alerts can be used to determine how an alert is fired?

- ○ Add severity
- ○ Add rule
- ○ Add specifications
- ○ Add criteria (Correct)

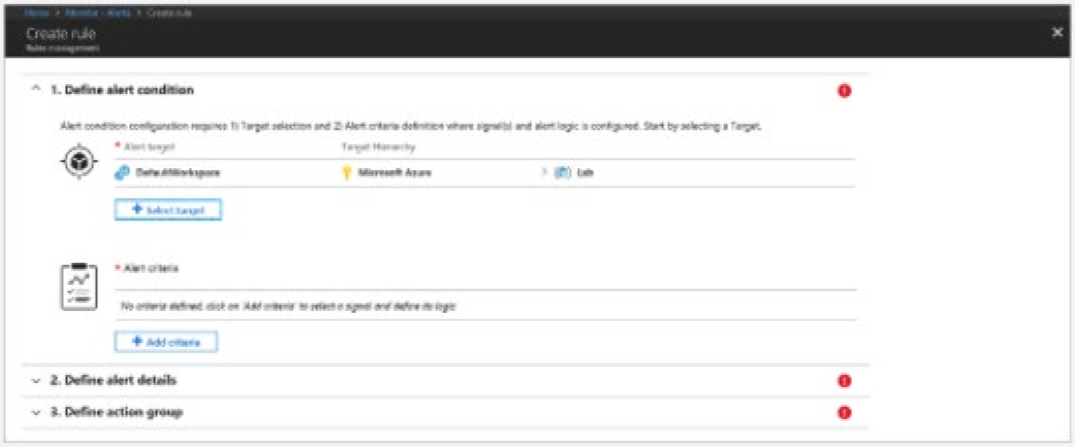
Explanation

Azure Data Factory Alerts provide an automated response that can be beneficial to monitor and audit Azure Data Factory activity. These alerts are very proactive and more efficient than manual monitoring operations. Alerts can be fired on both success and failure of a pipeline based on the rule configuration.

Alert Rule

Azure Data Factory Alerts use an alert rule which states the criteria upon which the alerts should trigger. We can enable or disable the alert rules.

• The **add criteria** feature enables you to determine how an alert is fired.



Home > Monitor - Alerts > Create rule

Create rule

Rules management

1. Define alert condition

Alert condition configuration requires 1) Target selection and 2) Alert criteria definition where signal(s) and alert logic is configured. Start by selecting a Target.

* Alert target	Target Hierarchy	
DefaultWorkspace	Microsoft Azure	> Lab

* Alert criteria

No criteria defined, click on 'Add criteria' to select a signal and define its logic

2. Define alert details

3. Define action group

https://docs.microsoft.com/en-us/azure/azure-monitor/alerts/tutorial-response

Question 41: Skipped
Azure provides many ways to store your data. A Storage account defines a policy that applies to all the storage services in the account. One of the settings within the Storage account is the Deployment Model which is the system Azure uses to organize the recourses.

Which of the following are valid deployment methods? (Select two)

- □
  Classic
  (Correct)
- □
  PowerShell
- □
  CLI
- □
  Resource Manager
  (Correct)
- □
  Boards
- □
  CloudShell

Explanation
Azure Storage Deployment Models

A *deployment model* is the system Azure uses to organize your resources. The model defines the API that you use to create, configure, and manage those resources. Azure provides two deployment models:

• **Resource Manager**: the current model that uses the Azure Resource Manager API

• **Classic**: a legacy offering that uses the Azure Service Management API

Most Azure resources only work with Resource Manager, and makes it easy to decide which model to choose. However, storage accounts, virtual machines, and virtual networks support both, so you must choose one or the other when you create your storage account.

The key feature difference between the two models is their support for grouping. The Resource Manager model adds the concept of a *resource group*, which is not available in the classic model. A resource group lets you deploy and manage a collection of resources as a single unit.

Microsoft recommends that you use **Resource Manager** for all new resources.

https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/deployment-models

Question 42: Skipped

Azure Data Factory is a cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale. Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from disparate data stores.

True or False: Each data factory has a single dedicated pipeline. When additional pipelines are needed for workloads, additional data factory deployments can be used to create an unlimited number of pipelines.

- ○ False (Correct)
- ○ True

Explanation

Azure Data Factory is a cloud-based ETL and data integration service that allows you to create data-driven workflows for orchestrating data movement and transforming data at scale. Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from disparate data stores.

A data factory can have one or more pipelines. A pipeline is a logical grouping of activities that together perform a task. For example, a pipeline could contain a set of activities that ingest and clean log data, and then kick off a mapping data flow to analyze the log data. The pipeline allows you to manage the activities as a set instead of each one individually. You deploy and schedule the pipeline instead of the activities independently.

https://docs.microsoft.com/en-us/azure/data-factory/introduction

Question 43: **Skipped**

Scenario: Queen Consolidated was overtaken by Raymond Carson Palmer and rebranded as Palmer Technologies. Now that Ray is overseeing the operations at Palmer, Ray has decided to move away from on-prem datacentres to Azure. Ray and the IT team are developing a new data engineering solutions for a company.

The current project is dealing with social media and has the following requirements.

Required:

• Real-time Twitter feed analysis of posts which contain specific keywords and must be stored as well as processed on MS Azure then displayed using MS Power BI.

Ray and the IT team have put together a list of actions they think need to be performed to meet the needs of the project, but they are not sure on the order to execute. Below is a list of the actions they are considering.

Proposed Actions:

a. Create an HDInsight cluster with the Hadoop cluster type.

b. Create a Jupyter Notebook.

c. Run a job that uses the Spark Streaming API to ingest data from Twitter.

d. Create a Runbook.

e. Create an HDInsight cluster with the Spark cluster type.

f. Create a HVAC table.

g. Load the HVAC table into Power BI Desktop

As you are the Azure SME, Ray and the team look to you for direction on selecting the required items and putting them in the proper order. Which of the below contains the correct items in the correct sequence to meet the requirements?

- ○
  e → a → c → g → d
- ○
  a → b → f → c → g
  (Correct)
- ○
  f → b → d → a → g → c

- b → a → e → f → c → g

Explanation

Step 1: Create an HDInisght cluster with the Spark cluster type.

Step 2: Create a Juputer Notebook.

Step 3: Create HVAC table.

The Jupyter Notebook that you created in the previous step includes code to create an HVAC table.

Step 4: Run a job that uses the Spark Streaming API to ingest data from Twitter.

Step 5: Load the HVAC table into Power BI Desktop.

You use Power BI to create visualizations, reports, and dashboards from the Spark cluster data.

https://www.youtube.com/watch?v=_RJ0VjZ2-og

https://docs.microsoft.com/en-us/azure/hdinsight/spark/apache-spark-use-with-data-lake-store

Question 44: Skipped

What steps are required to authorize Azure DevOps to connect to and deploy notebooks to a staging or production Azure Databricks workspace?

- ○ In the production or staging Azure Databricks workspace, enable Git integration to Azure DevOps, then link to the Azure DevOps source code repo.
- ○ Create an Azure Active Directory application, copy the application ID, then use that as the Databricks bearer token in the Databricks Notebooks Deployment step of the Release pipeline.
- ○ Create a new Access Token within the user settings in the production Azure Databricks workspace, then use the token as the Databricks bearer token in the Databricks Notebooks Deployment step of the Release pipeline. (Correct)
- ○ None of the listed options.

Explanation

To authorize Azure DevOps to connect to and deploy notebooks to a staging or production Azure Databricks workspace, create an Azure Active Directory application, copy the application ID, then use that as the Databricks bearer token in the Databricks Notebooks Deployment step of the Release pipeline.

The Access Token allows you to grant access to resources within an Azure Databricks workspace without passing in user credentials.

https://social.technet.microsoft.com/wiki/contents/articles/53094.azure-devops-integrate-with-an-azure-subscription-or-management-group.aspx

Question 45: **Skipped**

How many drivers does a Cluster have?

- ○ Configurable between one and ten
- ○ Only one (Correct)
- ○ Configurable between one and eight
- ○ Two, running in parallel

Explanation

A Cluster has one and only one driver.

Cluster node type

A cluster consists of one driver node and worker nodes.

You can pick separate cloud provider instance types for the driver and worker nodes, although by default the driver node uses the same instance type as the worker node. Different families of instance types fit different use cases, such as memory-intensive or compute-intensive workloads.

https://docs.microsoft.com/en-us/azure/databricks/clusters/configure

Question 46: **Skipped**

Spark pools in Azure Synapse Analytics is one of Microsoft's implementation of Apache Spark.

Which of the following are true about Spark pools in Azure Synapse Analytics? (Select all that apply)

- ☐
  The SparkContext connects to the Sparkle pool in Synapse Analytics. It is responsible for converting an application to an Excel file.
- ☐
  Spark applications act as independent sets of processes on a pool. It is coordinated by the SParkContext object in a main (driver) program
  (Correct)
- ☐
  Once connected, Sparkle gets the executors on nodes in the pool. Those processes run computations and store data on your local machine.
- ☐
  The SparkContext is able to connect to the cluster manager, which allocates resources across applications. The cluster manager is Adobe Hadoop WOOL.

Explanation

Apache Spark in Azure Synapse Analytics

Spark pools in Azure Synapse Analytics is one of Microsoft's implementation of Apache Spark, version Spark 2.4 for the Azure cloud.

Azure Synapse Analytics enables you to have a one-stop shop for your Analytics environment. With the addition of Spark Pools in Azure Synapse Analytics, it is now also possible to benefit from the features of Apache Spark in the same environment where you can set up your data warehousing solution. The spark pools within Azure Synapse Analytics are compatible with different Azure Storage solutions such as ADLS Gen2 and Blob Storage. It is imperative to know that currently providing Spark pools in an Azure Synapse Analytics workspace preview environment, is provided without a service level agreement and therefore not (yet) recommended for production workloads. In addition, some of the official Apache Spark documentation relies on using the spark console. At this moment, the spark console is not available on Azure Synapse Spark, so therefore it is highly recommended to use the notebook or IntelliJ experiences instead.

Spark Pools in Azure Synapse Analytics, a fully managed and integrated Spark service

Benefits of Spark Pools in Azure Synapse Analytics are listed below:

- Speed and Efficiency: Quick start-up time for nodes, automatic shut-down when instances are not used within 5 min after last job, unless there is a live notebook connection.

- Ease of creation: Creating a spark pool can be done through the Azure portal, PowerShell, or .NET SDK for Azure Synapse Analytics.

- Ease of use: Within the Azure Synapse Analytics workspace, you can connect directly to the Spark pool and interact with the integrated notebook experience, or use custom notebooks derived from Nteract. Notebook integration helps you in developing interactive data processing and visualization pipelines.

- REST APIs: In order to monitor and submit jobs remotely, you can use Apache Livy as Rest API Spark job server.

- Integration with third-party IDEs: Azure Synapse Analytics provides an IDE for IntelliJ to create and submit applications to the spark pool

- Pre-loaded Anaconda libraries: Over 200 Anaconda libraries pre-installed on the spark pool.

- Scalability: Possibility for autoscale, such that pools can be up/down scaled as required by adding or removing nodes.

Spark pools in Azure Synapse include the following components that are available on the pools by default.

- Spark Core. Includes Spark Core, Spark SQL, GraphX, and MLlib.

- Anaconda

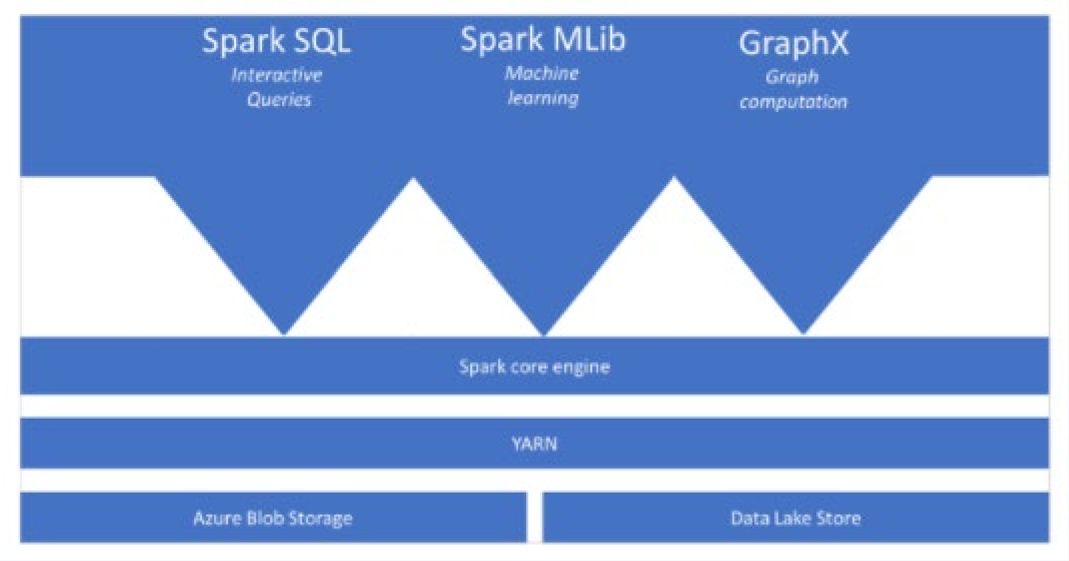
- Apache Livy

- Nteract notebook

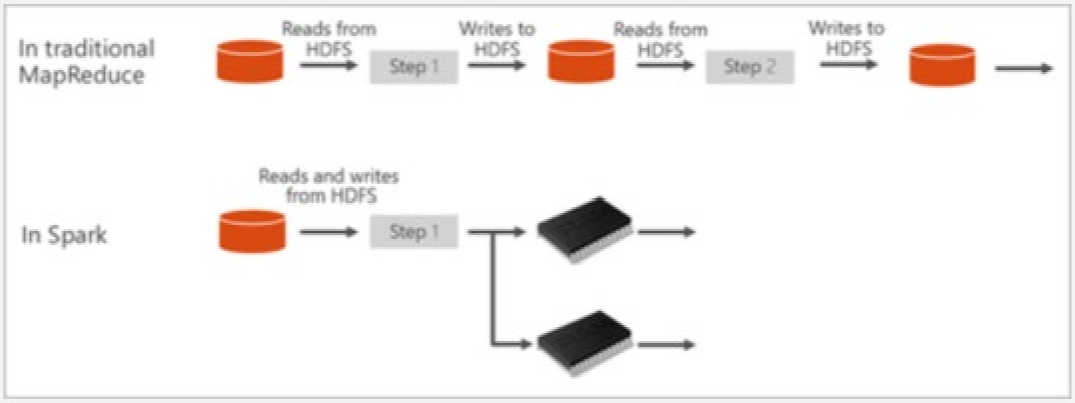
The supported languages and runtime versions for Apache spark and dependent components in Azure Synapse analytics can be found here:

- Apache Spark components in Azure Synapse Analytics

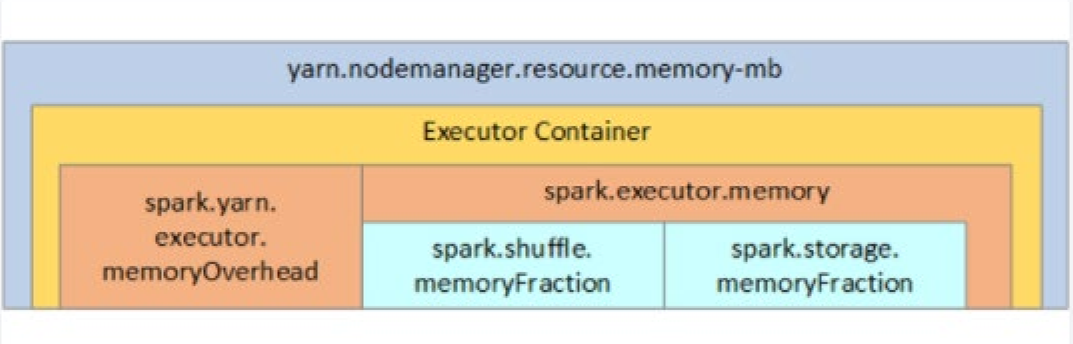
Spark pool architecture



It is imperative to understand the components of Spark by understanding how Spark runs on Synapse Analytics. **The different spark applications act as independent sets of processes on a pool. It is coordinated by the SParkContext object in a main (driver) program.**



The SparkContext is able to connect to the cluster manager, which allocates resources across applications. The cluster manager is Apache Hadoop YARN.



Once connected, Spark gets the executors on nodes in the pool. Those processes run computations and store data for your application. What follows is that your application code (defined by JAR or Python files passed to SparkContext) will be sent to the executors. Finally, SparkContext is able to send tasks to the executors to run.

The SparkContext runs the user's so your main function. What is then will do is execute the various parallel operations on the nodes. Then, the SparkContext will collect all the results of the operations that were sent to the nodes. The nodes are able to read and write data from and to the file system. Like mentioned in the introduction, the nodes caches the transformed data in-memory as Resilient Distributed Datasets (RDDs).

The SparkContext connects to the Spark pool in Synapse Analytics. It is responsible for converting an application to a directed acyclic graph (DAG). The graph consists of individual tasks that get executed within an executor process on the nodes. Each application gets its own executor processes, which stay up for the duration of the whole application and run tasks in multiple threads.

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-overview

Question 47: Skipped

What is a supported connector for built-in parameterization? (Select all that apply)

- ○

  Azure Data Lake Storage Gen1

- ○

  Azure Key Vault

- ○

  Azure Data Lake Storage Gen2

- ○

  Azure Synapse Analytics

  (Correct)

Explanation

Azure Synapse Analytics is a supported connector for built-in parameterization for Linked Services in Azure Data Factory.

Supported linked service types

You can parameterize any type of linked service. When authoring linked service on UI, Data Factory provides built-in parameterization experience for the following types of linked services. In linked service creation/edit blade, you can find options to new parameters and add dynamic content.

• Amazon Redshift

• Amazon S3

• Azure Cosmos DB (SQL API)

• Azure Database for MySQL

• Azure Databricks

• Azure Key Vault

• Azure SQL Database

• Azure SQL Managed Instance

• Azure Synapse Analytics

• MySQL

• Oracle

• SQL Server

• Generic HTTP

• Generic REST

For other linked service types that are not in above list, you can parameterize the linked service by editing the JSON on UI:

• In linked service creation/edit blade → expand "Advanced" at the bottom → check "Specify dynamic contents in JSON format" checkbox → specify the linked service JSON payload.

• Or, after you create a linked service without parameterization, in Management hub → Linked services → find the specific linked service → click "Code" `(button "{}")` to edit the JSON.

Refer to the JSON sample to add `parameters` section to define parameters and reference the parameter using `@{linkedService().paraName}`.

https://docs.microsoft.com/en-us/azure/data-factory/parameterize-linked-services

Question 48: Skipped

Scenario: Data loads at your company have increased the processing time for on-premises data warehousing descriptive analytic solutions. You have been tasked with looking into a cloud-based alternative to reduce processing time and release business intelligence reports faster. Your boss wants you to first consider scaling up on-premises servers but you discover this approach would reach its physical limits shortly.

The new solution must be on a petabyte scale that doesn't involve complex installations and configurations.

Which of the following would best suit the need?

- ○ Azure Synapse Analytics (Correct)
- ○ Azure DataNow
- ○ Azure Table Storage
- ○ Azure Stream Analytics
- ○ Azure On-prem Solution
- ○ Azure Cosmos DB

Explanation

Azure Synapse Analytics is a cloud-based data platform that brings together enterprise data warehousing and Big Data analytics. It can process massive amounts of data and answer complex business questions with limitless scale.

When to use Azure Synapse Analytics

The SQL Pools capability of Azure Synapse Analytics can meet the scenario needs.

The volume and variety of data that is being generated are providing opportunities to perform different types of analysis on the data. This can include techniques such as exploratory data analysis to identify initial patterns or meaning in the data. It can also include conducting predictive analytics for forecasting, or segmenting data. The Big Data Analytics capability of Azure Synapse Analytics will accommodate this.

Key features

SQL Pools uses massively parallel processing (MPP) to quickly run queries across petabytes of data. Because the storage is separated from the compute nodes, you can scale the compute nodes independently to meet any demand at any time.

In Azure Synapse Analytics, the Data Movement Service (DMS) coordinates and transports data between compute nodes as necessary. But you can use a replicated table to reduce data movement and improve performance. Azure Synapse Analytics supports three types of distributed tables: hash, round-robin and replicated. Use these tables to tune performance.

Importantly, Azure Synapse Analytics can also pause and resume the compute layer. This means you pay only for the computation you use. This capability is useful in data warehousing.

https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is

Question 49: **Skipped**

Scenario: You have been contracted by Wayne Enterprises, a company owned by Bruce Wayne with market value of over twenty seven million dollars. Bruce founded Wayne Enterprises shortly after he created the Wayne Foundation and he became the president and chairman of the company.

Bruce has come to you because his IT team needs advice on the use of Azure SQL Database to support a mission-critical application.

Required:

• The application must be highly available

• No performance loss during maintenance cycles

Which of the following applications should you recommend to Bruce and his team to adopt?

- □
  Ultra service tier
- □
  SQL Data Sync
- □
  Virtual machine Scale Sets
- □
  Premium service tier
  (Correct)
- □
  Always On availability groups
  (Correct)
- □
  Zone-redundant configuration
  (Correct)

Explanation

Premium service tier: Premium/business critical service tier model that is based on a cluster of database engine processes. This architectural model relies on a fact that there is always a quorum of available database engine nodes and has minimal performance impact on your workload even during maintenance activities.

Always On availability groups: In the premium model, Azure SQL database integrates compute and storage on the single node. High availability in this architectural model is achieved by replication of compute (SQL Server Database Engine process) and storage (locally attached SSD) deployed in 4-node cluster, using technology similar to SQL

Zone redundant configuration: By default, the quorum-set replicas for the local storage configurations are created in the same datacentre. With the introduction of Azure Availability Zones, you have the ability to place the different replicas in the quorum-sets to different availability zones in the same region. To eliminate a single point of failure, the control ring is also duplicated across multiple zones as three gateway rings (GW).

The goal of the high availability architecture in Azure SQL Database and SQL Managed Instance is to guarantee that your database is up and running minimum of 99.99% of time (For more information regarding specific SLA for different tiers, Please refer SLA for Azure SQL Database and SQL Managed Instance), without worrying about the impact of maintenance operations and outages. Azure automatically handles critical servicing tasks, such as patching, backups, Windows and Azure SQL upgrades, as well as unplanned events such as underlying hardware, software, or network failures. When the underlying database in Azure SQL Database is patched or fails over, the downtime is not noticeable if you employ retry logic in your app. SQL Database and SQL Managed Instance can quickly recover even in the most critical circumstances ensuring that your data is always available.

The high availability solution is designed to ensure that committed data is never lost due to failures, that maintenance operations do not affect your workload, and that the database will not be a single point of failure in your software architecture. There are no maintenance windows or downtimes that should require you to stop the workload while the database is upgraded or maintained.

There are two high availability architectural models:

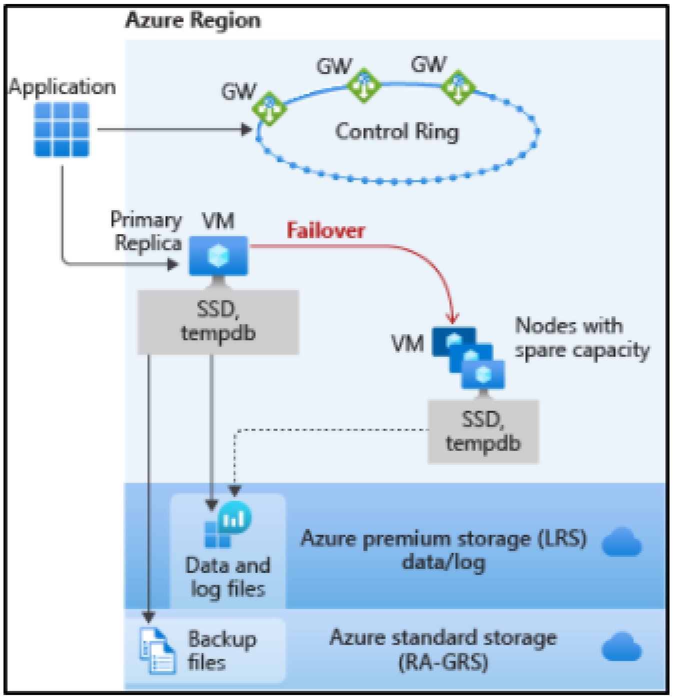
Standard availability model that is based on a separation of compute and storage. It relies on high availability and reliability of the remote storage tier. This architecture targets budget-oriented business applications that can tolerate some performance degradation during maintenance activities.

Premium availability model that is based on a cluster of database engine processes. It relies on the fact that there is always a quorum of available database engine nodes. This architecture targets mission critical applications with high IO performance, high transaction rate and guarantees minimal performance impact to your workload during maintenance activities.

SQL Database and SQL Managed Instance both run on the latest stable version of the SQL Server database engine and Windows operating system, and most users would not notice that upgrades are performed continuously.

Basic, Standard, and General Purpose service tier locally redundant availability

The Basic, Standard, and General Purpose service tiers leverage the standard availability architecture for both serverless and provisioned compute. The following figure shows four different nodes with the separated compute and storage layers.



The standard availability model includes two layers:

A stateless compute layer that runs the `sqlservr.exe` process and contains only transient and cached data, such as TempDB, model databases on the attached SSD, and plan cache, buffer pool, and columnstore pool in memory. This stateless node is operated by Azure Service Fabric that initializes `sqlservr.exe`, controls health of the node, and performs failover to another node if necessary.

A stateful data layer with the database files (.mdf/.ldf) that are stored in Azure Blob storage. Azure blob storage has built-in data availability and redundancy feature. It guarantees that every record in the log file or page in the data file will be preserved even if `sqlservr.exe` process crashes.

Whenever the database engine or the operating system is upgraded, or a failure is detected, Azure Service Fabric will move the stateless `sqlservr.exe` process to another stateless compute node with sufficient free capacity. Data in Azure Blob storage is not affected by the move, and the data/log files are attached to the newly initialized `sqlservr.exe` process. This process guarantees 99.99% availability, but a heavy workload may experience some performance degradation during the transition since the new `sqlservr.exe` process starts with cold cache.

General Purpose service tier zone redundant availability

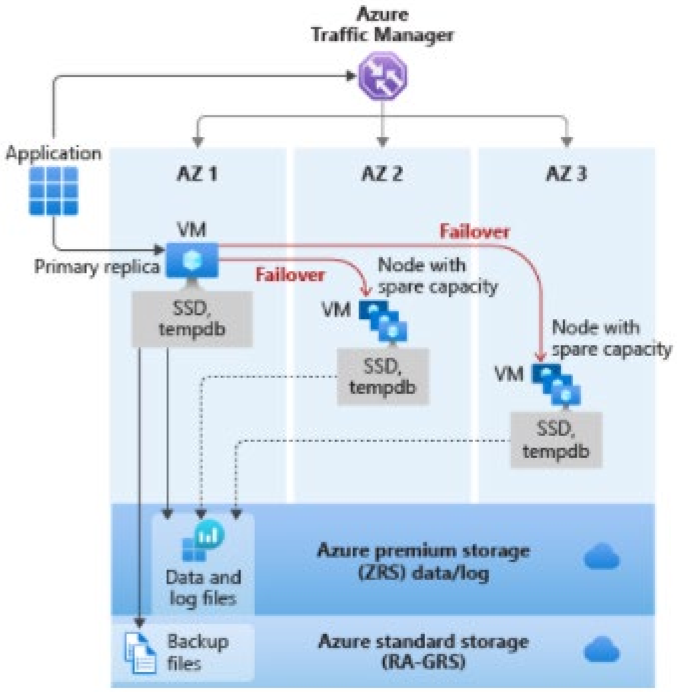
Zone redundant configuration for the general purpose service tier is offered for both serverless and provisioned compute. This configuration utilizes Azure Availability Zones to replicate databases across multiple physical locations within an Azure region. By selecting zone redundancy, you can make your new and existing serverlesss and provisioned general purpose single databases and elastic pools resilient to a much larger set of failures, including catastrophic datacenter outages, without any changes of the application logic.

Zone redundant configuration for the general purpose tier has two layers:

A stateful data layer with the database files (.mdf/.ldf) that are stored in ZRS(zone-redundant storage). Using ZRS the data and log files are synchronously copied across three physically-isolated Azure availability zones.

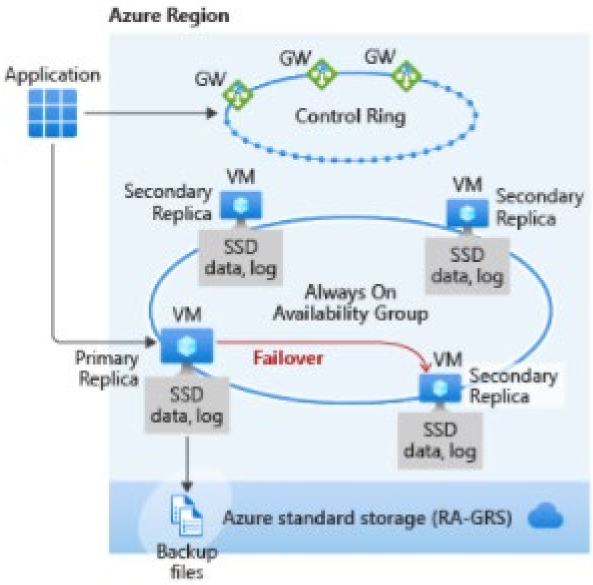
A stateless compute layer that runs the sqlservr.exe process and contains only transient and cached data, such as TempDB, model databases on the attached SSD, and plan cache, buffer pool, and columnstore pool in memory. This stateless node is operated by Azure Service Fabric that initializes sqlservr.exe, controls health of the node, and performs failover to another node if necessary. For zone redundant serverless and provisioned general purpose databases, nodes with spare capacity are readily available in other Availability Zones for failover.

The zone redundant version of the high availability architecture for the general purpose service tier is illustrated by the following diagram:



Premium and Business Critical service tier locally redundant availability

Premium and Business Critical service tiers leverage the Premium availability model, which integrates compute resources (`sqlservr.exe` process) and storage (locally attached SSD) on a single node. High availability is achieved by replicating both compute and storage to additional nodes creating a three to four-node cluster.



The underlying database files (.mdf/.ldf) are placed on the attached SSD storage to provide very low latency IO to your workload. High availability is implemented using a technology similar to SQL Server Always On availability groups. The cluster includes a single primary replica that is accessible for read-write customer workloads, and up to three secondary replicas (compute and storage) containing copies of data. The primary node constantly pushes changes to the secondary nodes in order and ensures that the data is synchronized to at least one secondary replica before committing each transaction. This process guarantees that if the primary node crashes for any reason, there is always a fully synchronized node to fail over to. The failover is initiated by the Azure Service Fabric. Once the secondary replica becomes the new primary node, another secondary replica is created to ensure the cluster has enough nodes (quorum set). Once failover is complete, Azure SQL connections are automatically redirected to the new primary node.

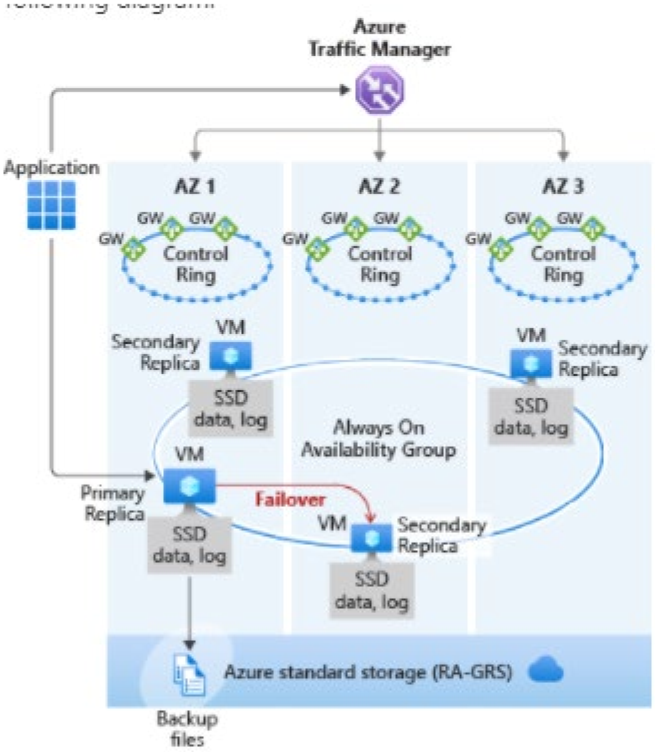
As an extra benefit, the premium availability model includes the ability to redirect read-only Azure SQL connections to one of the secondary replicas. This feature is called Read Scale-Out. It provides 100% additional compute capacity at no extra charge to off-load read-only operations, such as analytical workloads, from the primary replica.

Premium and Business Critical service tier zone redundant availability

By default, the cluster of nodes for the premium availability model is created in the same datacenter. With the introduction of Azure Availability Zones, SQL Database can place different replicas of the Business Critical database to different availability zones in the same region. To eliminate a single point of failure, the control ring is also duplicated across multiple zones as three gateway rings (GW). The routing to a specific gateway ring is controlled by Azure Traffic Manager (ATM). Because the zone redundant configuration in the Premium or Business Critical service tiers does not create additional database redundancy, you can enable it at no extra cost. By selecting a zone redundant configuration, you can make your Premium or Business Critical databases resilient to a much larger set of failures, including catastrophic datacenter outages, without any changes to the application logic. You can also convert any existing Premium or Business Critical databases or pools to the zone redundant configuration.

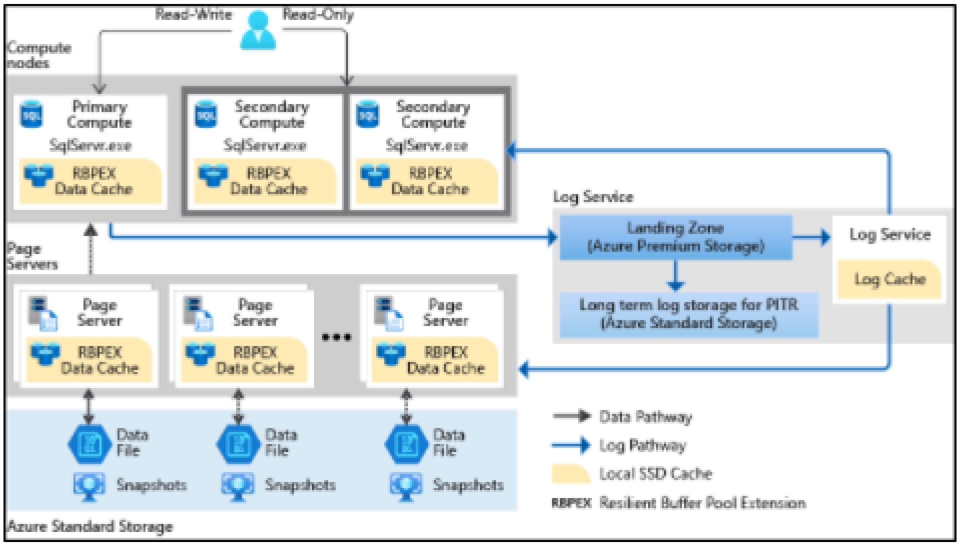
Because the zone redundant databases have replicas in different datacenters with some distance between them, the increased network latency may increase the commit time and thus impact the performance of some OLTP workloads. You can always return to the single-zone configuration by disabling the zone redundancy setting. This process is an online operation similar to the regular service tier upgrade. At the end of the process, the database or pool is migrated from a zone redundant ring to a single zone ring or vice versa.

The zone redundant version of the high availability architecture is illustrated by the following diagram:



Hyperscale service tier availability

The Hyperscale service tier architecture is described in Distributed functions architecture and is only currently available for SQL Database, not SQL Managed Instance.



The availability model in Hyperscale includes four layers:

A stateless compute layer that runs the `sqlservr.exe` processes and contains only transient and cached data, such as non-covering RBPEX cache, TempDB, model database, etc. on the attached SSD, and plan cache, buffer pool, and columnstore pool in memory. This stateless layer includes the primary compute replica and optionally a number of secondary compute replicas that can serve as failover targets.

A stateless storage layer formed by page servers. This layer is the distributed storage engine for the `sqlservr.exe` processes running on the compute replicas. Each page server contains only transient and cached data, such as covering RBPEX cache on the attached SSD, and data pages cached in memory. Each page server has a paired page server in an active-active configuration to provide load balancing, redundancy, and high availability.

A stateful transaction log storage layer formed by the compute node running the Log service process, the transaction log landing zone, and transaction log long term storage. Landing zone and long term storage use Azure Storage, which provides availability and redundancy for transaction log, ensuring data durability for committed transactions.

A stateful data storage layer with the database files (.mdf/.ndf) that are stored in Azure Storage and are updated by page servers. This layer uses data availability and redundancy features of Azure Storage. It guarantees that every page in a data file will be preserved even if processes in other layers of Hyperscale architecture crash, or if compute nodes fail.

Compute nodes in all Hyperscale layers run on Azure Service Fabric, which controls health of each node and performs failovers to available healthy nodes as necessary.

For more information on high availability in Hyperscale, see Database High Availability in Hyperscale.

Accelerated Database Recovery (ADR)

Accelerated Database Recovery (ADR) is a new database engine feature that greatly improves database availability, especially in the presence of long running transactions. ADR is currently available for Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics.

Testing application fault resiliency

High availability is a fundamental part of the SQL Database and SQL Managed Instance platform that works transparently for your database application. However, we recognize that you may want to test how the automatic failover operations initiated during planned or unplanned events would impact an application before you deploy it to production. You can manually trigger a failover by calling a special API to restart a database, an elastic pool, or a managed instance. In the case of a zone redundant serverless or provisioned General Purpose database or elastic pool, the API call would result in redirecting client connections to the new primary in an Availability Zone different from the Availability Zone of the old primary. So in addition to testing how failover impacts existing database sessions, you can also verify if it changes the end-to-end performance due to changes in network latency. Because the restart operation is intrusive and a large number of them could stress the platform, only one failover call is allowed every 15 minutes for each database, elastic pool, or managed instance.

Azure SQL Database and Azure SQL Managed Instance feature a built-in high availability solution, that is deeply integrated with the Azure platform. It is dependent on Service Fabric for failure detection and recovery, on Azure Blob storage for data protection, and on Availability Zones for higher fault tolerance (as mentioned earlier in document not applicable to Azure SQL Managed Instance yet). In addition, SQL Database and SQL Managed Instance leverage the Always On availability group technology from the SQL Server instance for replication and failover. The combination of these technologies enables applications to fully realize the benefits of a mixed storage model and support the most demanding SLAs.

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-high-availability

Question 50: **Skipped**

What happens if the command option ("checkpointLocation", pointer-to-checkpoint directory) is not specified in Structured Streaming?

- ○ It will not be possible to create more than one streaming query that uses the same streaming source since they will conflict.
- ○ The streaming job will function as expected since the checkpointLocation option does not exist.
- ○ When the streaming job stops, all state around the streaming job dumped to a default location, and upon restart, the job must start from aggregated data rather than tuned specific data.
- ○ When the streaming job stops, all state data around the streaming job is lost, and upon restart, the job must start from scratch.
  (Correct)

Explanation

Setting the checkpoint Location is required for many sinks used in Structured Streaming. For those sinks where this setting is optional, keep in mind that when you do not set this value, you risk losing your place in the stream.

https://www.waitingforcode.com/apache-spark-structured-streaming/checkpoint-storage-structured-streaming/read