Question 1: Skipped

Init Scripts provide a way to configure cluster's nodes. It is recommended to favour Cluster Scoped Init Scripts over Global and Named scripts.

Which of the following is best described by:

"You can limit the init script to run only on for a specific cluster's creation and restarts by placing it in `/databricks/init/<cluster_name>` *folder."*

- Cluster Named
  (Correct)
- Cluster Scoped
- Interactive
- Global

Explanation
Favour cluster scoped init scripts over global and named scripts

Init Scripts provide a way to configure cluster's nodes and to perform custom installs. Init scripts can be used in the following modes:

• **Global:** by placing the Init script in `/databricks/init` folder, you force the script's execution every time any cluster is created or restarted by users of the workspace.

• **Cluster Named (deprecated):** you can limit the init script to run only on for a specific cluster's creation and restarts by placing it in `/databricks/init/<cluster_name>` folder.

• **Cluster Scoped**: in this mode, the Init script is not tied to any cluster by its name and its automatic execution is not a virtue of its dbfs location. Rather, you specify the script in cluster's configuration by either writing it directly in the cluster configuration UI or storing it on Databricks File System (DBFS) and specifying the path in Cluster Create API. Any location under `DBFS /databricks` folder except `/databricks/init` can be used for this purpose, such as: `/databricks/<my-directory>/set-env-var.sh`

You should treat Init scripts with *extreme* caution because they can easily lead to intractable cluster launch failures. If you really need them, please use the **Cluster Scoped execution mode** as much as possible because:

• ADB executes the script's body in each cluster node. Thus, a successful cluster launch and subsequent operation are predicated on all nodal Init scripts executing in a timely manner without any errors and reporting a zero exit code. This process is highly error prone, especially for scripts downloading artifacts from an external service over unreliable and/or misconfigured networks.

• Because Global and Cluster Named Init scripts execute automatically due to their placement in a special DBFS location, it is easy to overlook that they could be causing a cluster to not launch. By specifying the Init script in the Configuration, there's a higher chance that you'll consider them while debugging launch failures.

Use cluster log delivery feature to manage logs

By default, Cluster logs are sent to default DBFS but you should consider sending the logs to a blob store location under your control using the Cluster Log Delivery feature. The Cluster Logs contain logs emitted by user code, as well as Spark framework's Driver and Executor logs. Sending them to a blob store controlled by yourself is recommended over default DBFS location because:

• ADB's automatic 30-day default DBFS log purging policy might be too short for certain compliance scenarios. A blob store location in your subscription will be free from such policies.

• You can ship logs to other tools only if they are present in your storage account and a resource group governed by you. The root DBFS, although present in your subscription, is launched inside a Microsoft Azure managed resource group and is protected by a read lock. Because of this lock, the logs are only accessible by privileged Azure Databricks framework code. However, constructing a pipeline to ship the logs to downstream log analytics tools requires logs to be in a lock-free location first.

https://github.com/Azure/AzureDatabricksBestPractices/blob/master/toc.md

---

Question 2: Skipped

How do you infer the data types and column names when you read a JSON file?

- `spark.read.option.inferSchema("true").json(jsonFile)`
- `spark.read.inferSchema("true").json(jsonFile)`
- `spark.read.option("inferSchema", "true").json(jsonFile)`
  (Correct)
- `spark.read.option("inferData", "true").json(jsonFile)`

Explanation

The `spark.read.option("inferSchema", "true").json(jsonFile)` approach is the correct way to infer the file's schema.

https://bartoszgajda.com/2020/06/26/exploiting-schema-inference-in-apache-spark/

Question 3: Skipped

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

Synapse dedicated SQL Pools supports JSON format data to be stored using [?]. The JSON format enables representation of complex or hierarchical data structures in tables. It allows to transform arrays of JSON objects into table format. The performance of JSON data can be optimized by using columnstore indexes and memory optimized tables.

- Standard `VARCHAR` table columns
- Standard `CHAR` table columns
- Standard `NCHAR` table columns
- Standard `NVARCHAR` table columns
  (Correct)

Explanation
Synapse dedicated SQL Pools supports JSON format data to be stored using **standard** `NVARCHAR` **table columns.** The JSON format enables representation of complex or hierarchical data structures in tables. It allows to transform arrays of JSON objects into table format. The performance of JSON data can be optimized by using columnstore indexes and memory optimized tables.

Insert JSON data - JSON data can be inserted using the usual T-SQL INSERT statements.

Read JSON data - JSON data can be read using the following T-SQL functions and provides the ability to perform aggregation and filter on JSON values.

- `ISJSON` – verify if text is valid JSON
- `JSON_VALUE` – extract a scalar value from a JSON string
- `JSON_QUERY` – extract a JSON object or array from a JSON string

Modify JSON data - JSON data can be modified and queried using the following T-SQL functions providing ability to update JSON string using T-SQL and convert hierarchical data into flat tabular structure.

• `JSON_MODIFY` – modifies a value in a JSON string

• `OPENJSON` – convert JSON collection to a set of rows and columns

You can also query JSON files using SQL serverless. The query's objective is to read the following type of JSON files using `OPENROWSET`.

• Standard JSON files where multiple JSON documents are stored as a JSON array.

• Line-delimited JSON files, where JSON documents are separated with new-line character. Common extensions for these types of files are `jsonl`, `ldjson`, and `ndjson`.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/query-json-files

---

Question 4: Skipped

Scenario: You are working on a project and you have been tasked with starting up a data platform service to execute as Spark job. The objective on this job is to ingest and process data and then shut down the service after the job is complete.

Which of the following would be the best compute resource to use?

- On-demand HDInsight cluster
  (Correct)
- None of the listed options
- Azure-SSIS Runtime
- HDInsight
- Azure Databricks

Explanation
On-demand HDInsight cluster service to execute as Spark job to ingest and process data and then shut down the service after the job is complete.

https://www.red-gate.com/simple-talk/cloud/infrastructure-as-a-service/automating-azure-creating-demand-hdinsight-cluster/

Question 5: 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 IT team has come up with a list of commands they are considering to execute which is shown below:

a. `CREATE EXTERNAL DATA SOURCE`

b. `CREATE EXTERNAL FILE FORMAT`

c. `CREATE EXTERNAL TABLE`

d. `CREATE EXTERNAL TABLE AS SELECT`

e. `CREATE DATABASE SCOPED CREDENTIAL`

The Ask:

Alfred places a great importance on this project and asks you to work closely with the team to ensure that the Twitter feed data can be analyzed in the dedicated SQL pool. The solution must meet the customer sentiment analytic requirements.

As the Azure expert, the team looks to you for direction with regards to the proper path forward.

Which Transact-SQL DDL commands should you recommend to be run in sequence?

- a → b → d
  (Correct)
- d → a → b → e
-

- c → d → a → e
- e → d → e → a → c

Explanation
The correct commands in sequence are a → b → d.

The requirement is to 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.

Command 1: `CREATE EXTERNAL DATA SOURCE`

External data sources are used to connect to storage accounts.

Command 2: `CREATE EXTERNAL FILE FORMAT`

`CREATE EXTERNAL FILE FORMAT` creates an external file format object that defines external data stored in Azure Blob Storage or Azure Data Lake Storage.

Creating an external file format is a prerequisite for creating an external table.

Command 3: `CREATE EXTERNAL TABLE AS SELECT`

When used in conjunction with the `CREATE TABLE AS SELECT` statement, selecting from an external table imports data into a table within the SQL pool. In addition to the `COPY` statement, external tables are useful for loading data.

The `CREATE EXTERNAL TABLE` command is a wrong choice because it creates an external table for Synapse SQL to access data stored in Azure Blob Storage or Azure Data Lake Storage.

External tables with Synapse SQL

An external table points to data located in Hadoop, Azure Storage blob, or Azure Data Lake Storage. External tables are used to read data from files or write data to files in Azure Storage. With Synapse SQL, you can use external tables to read external data using dedicated SQL pool or serverless SQL pool.

Depending on the type of the external data source, you can use two types of external tables:

Hadoop external tables that you can use to read and export data in various data formats such as CSV, Parquet, and ORC. Hadoop external tables as available in dedicated Synapse SQL pools, but they are not available in serverless SQL pools.

Native external tables that you can use to read and export data in various data formats such as CSV as Parquet. Native external tables are available in serverless Synapse SQL pools, but they are not available in dedicated Synapse SQL pools.

The key differences between Hadoop and native external tables are presented in the following table:

External table type	Hadoop	Native
Dedicated SQL pool	Available	Not available
Serverless SQL pool	Not available	Available
Supported formats	Delimited/CSV, Parquet, ORC, Hive RC, and RC	Delimited/CSV and Parquet
Folder partition elimination	No	Only for the partitioned tables synchronized from Apache Spark pools in Synapse workspace
Custom format for location	No	Yes, using wildcards like `/year=*/month=*/day=*`
Recursive folder scan	Always	Only when specified `/**` in the location path
Storage authentication	Storage Access Key(SAK), AAD passthrough, Managed identity, Custom application Azure AD identity	Shared Access Signature(SAS), AAD passthrough, Managed identity

External tables in dedicated SQL pool and serverless SQL pool

You can use external tables to:

• Query Azure Blob Storage and Azure Data Lake Gen2 with Transact-SQL statements.

• Store query results to files in Azure Blob Storage or Azure Data Lake Storage using CETAS

• Import data from Azure Blob Storage and Azure Data Lake Storage and store it into dedicated SQL pool (only Hadoop tables in dedicated pool).

*Note: When used in conjunction with the CREATE TABLE AS SELECT statement, selecting from an external table imports data into a table within the **dedicated** SQL pool. In addition to the COPY statement, external tables are useful for loading data.*

You can create external tables in Synapse SQL pools via the following steps:

- `CREATE EXTERNAL DATA SOURCE`
- `CREATE EXTERNAL FILE FORMAT`
- `CREATE EXTERNAL TABLE`

Security

User must have `SELECT` permission on external table to read the data. External table access underlying Azure storage using the database scoped credential defined in data source using the following rules:

• Data source without credential enables external tables to access publicly available files on Azure storage.

• Data source can have credential that enables external tables to access only the files on Azure storage using SAS token or workspace Managed Identity

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-external-tables

---

Question 6: Skipped

What Transact-SQL function is used to perform a `HyperLogLog` function?

- `APPROX_COUNT_DISTINCT`
  (Correct)
- None of the listed options.
- `COUNT_DISTINCT_APPROX`
- `HYPER_LOG_LOG`
- `COUNT`

Explanation

The `APPROX_COUNT_DISTINCT` function is used to perform a `HyperLogLog` function.

It is not uncommon for data engineers, data analysts, and data scientists alike to perform exploratory data analysis to gain an understanding of the data that they are working with. Exploratory data analysis can involve querying metadata about the data that is stored within the database, to running queries to provide a statistics information about the data such as average values for a column, through to distinct counts. Some of the activities can be time consuming, especially on large data sets.

For example, performing a distinct count of values in a Billion plus row table can be an expensive operation that takes time to resolve. As exploratory data analysis sometime doesn't require accurate information, there is a solution.

Azure Synapse Analytics supports Approximate execution using Hyperlog accuracy to reduce latency when executing queries with large datasets. Approximate execution is used to speed up the execution of queries with a compromise for a small reduction in accuracy. So if it takes too long to get basic information about the data in a large data set as you are exploring data of a big data set, then you can use the `HyperLogLog` accuracy and will return a result with a 2% accuracy of true cardinality on average. This is done by using the `APPROX_COUNT_DISTINCT` Transact-SQL function.

https://www.slideshare.net/jamserra/azure-synapse-analytics-overview

Question 7: Skipped

Which T-SQL Statement loads data directly from Azure Storage?

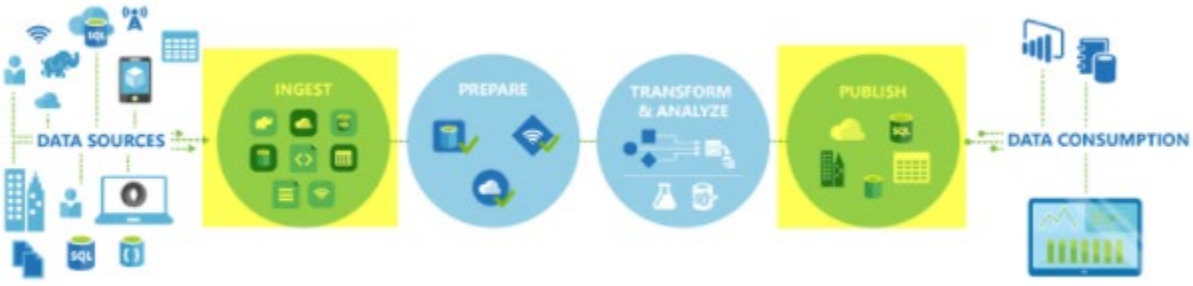
- `DUPLICATE`
- `GET`
- `LOAD DATA`
- `COPY`
  (Correct)
- `INSERT FROM FILE`
- `PULL`

Explanation

The T-SQL COPY Statement reads data from Azure Blob Storage or the Azure Data Lake and inserts it into a table within the SQL Pool.

The broad capabilities of the Copy Activity allow you to quickly and easily move data into SQL Pools from a variety of sources.

In Azure Data Factory, you can use the Copy activity to copy data among data stores located on-premises and in the cloud. After you copy the data, you can use other activities to further transform and analyze it. You can also use the Copy activity to publish transformation and analysis results for business intelligence (BI) and application consumption.



The Copy activity is executed on an integration runtime. You can use different types of integration runtimes for different data copy scenarios:

• When you're copying data between two data stores that are publicly accessible through the internet from any IP, you can use the Azure integration runtime for the copy activity. This integration runtime is secure, reliable, scalable, and globally available.

• When you're copying data to and from data stores that are located on-premises or in a network with access control (for example, an Azure virtual network), you need to set up a self-hosted integration runtime.

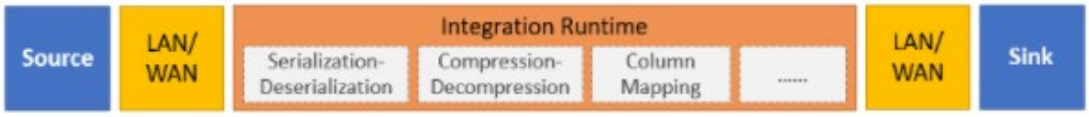
An integration runtime needs to be associated with each source and sink data store. For information about how the Copy activity determines which integration runtime to use, see Determining which IR to use.

To copy data from a source to a sink, the service that runs the Copy activity performs these steps:

1. Reads data from a source data store.

2. Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration of the input dataset, output dataset, and Copy activity.

3. Writes data to the sink/destination data store.



The Copy Activity supports a large range of data sources and sinks on-premises and in the cloud. It facilitates the efficient, yet flexible parsing and transfer of data or files between systems in an optimized fashion as well as giving you capability of easily converting datasets into other formats.

In the following example, you can load data from a public storage account. Here the `COPY` statement's defaults match the format of the line item csv file.

```
SQL

COPY INTO dbo.[lineitem] FROM 'https://unsecureaccount.blob.core.windows.net/customerdatasets/folder1/lineitem.csv'
```

The default values for csv files of the COPY command are:

- `DATEFORMAT = Session DATEFORMAT`
- `MAXERRORS = 0`
- `COMPRESSION` default is uncompressed
- `FIELDQUOTE = “”`
- `FIELDTERMINATOR = “,”`
- `ROWTERMINATOR = ‘\n’`
- `FIRSTROW = 1`
- `ENCODING = ‘UTF8’`
- `FILE_TYPE = ‘CSV’`
- `IDENTITY_INSERT = ‘OFF’`

https://docs.microsoft.com/en-us/azure/data-factory/copy-activity-overview

---

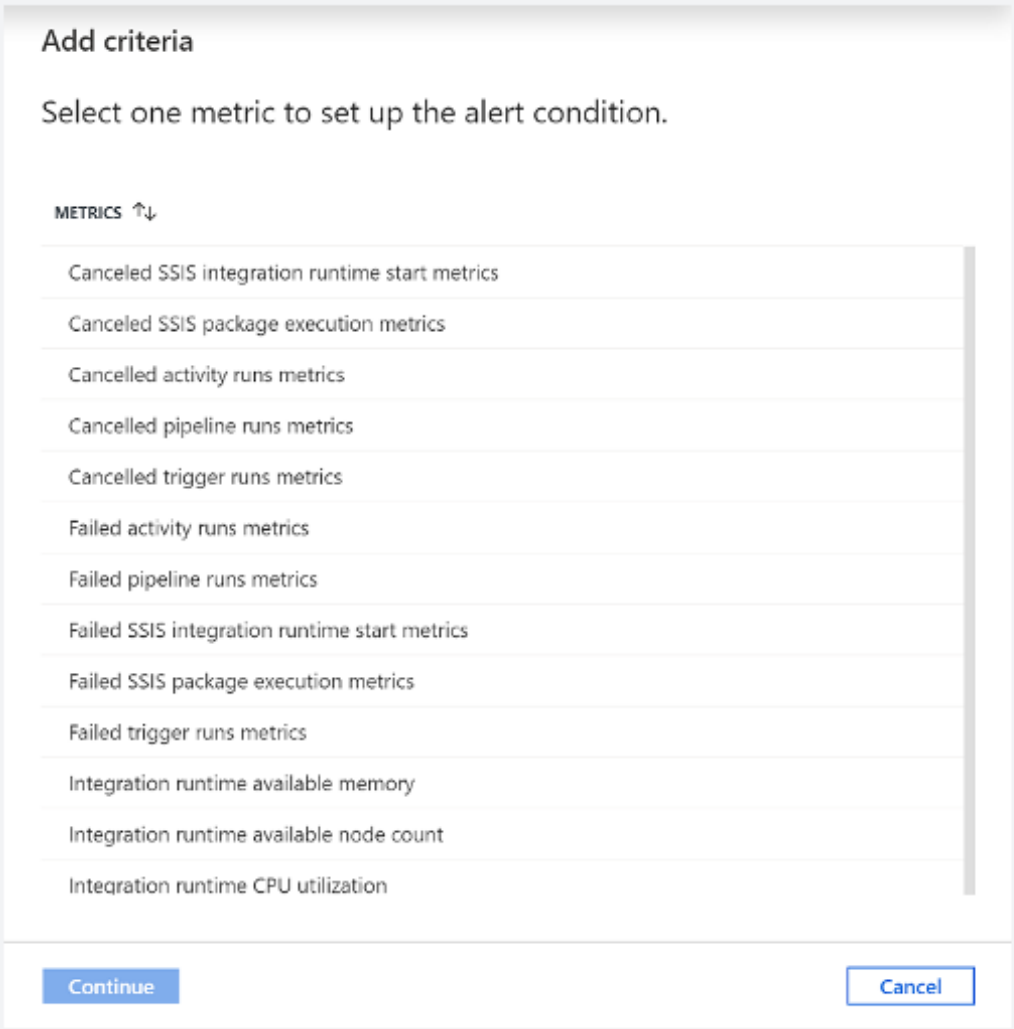
Question 8: Skipped

In Azure Data Factory, you can raise alerts based upon metrics outputted by the monitoring service. Alerts provide information on a variety of scenarios such as … (Select all that apply)

- Integration runtime CPU utilization
  (Correct)
- Integration runtime available memory
  (Correct)
- Failed activity run metrics
  (Correct)
- Failed pipelines
  (Correct)
- Failed trigger runs metrics
  (Correct)
- Cancelled SSIS integration runtime start metrics
  (Correct)
- Large factory sizes
  (Correct)
- Cancelled activity runs
  (Correct)
- Successful trigger runs metrics

Explanation

In Azure Data Factory, you can raise alerts based upon metrics outputted by the monitoring service. Alerts allow you to get alerted for a variety of scenarios such as, but not limited to, failed pipelines, large factory sizes, and integration runtime CPU utilization.



Alerts in the monitoring experience are based upon high-level metrics such as pipeline failures. For custom alerting on specific conditions that may occur within a pipeline or based upon data quality, it is recommended to configure these using a pipeline activity.

To get started, go to the **Monitor** tab and select **Alerts & metrics**.

https://azure.microsoft.com/en-us/blog/create-alerts-to-proactively-monitor-your-data-factory-pipelines/

Question 9: Skipped

Usually when you see 'df' in some code it refers to a [?].

```
1. Python
2. new_rows = <additional code here>
3. demo_df = <additional code here>
```

- Dateformat
- Dataformat
- Dataframe **(Correct)**
- Dataflow
- Datafeature

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

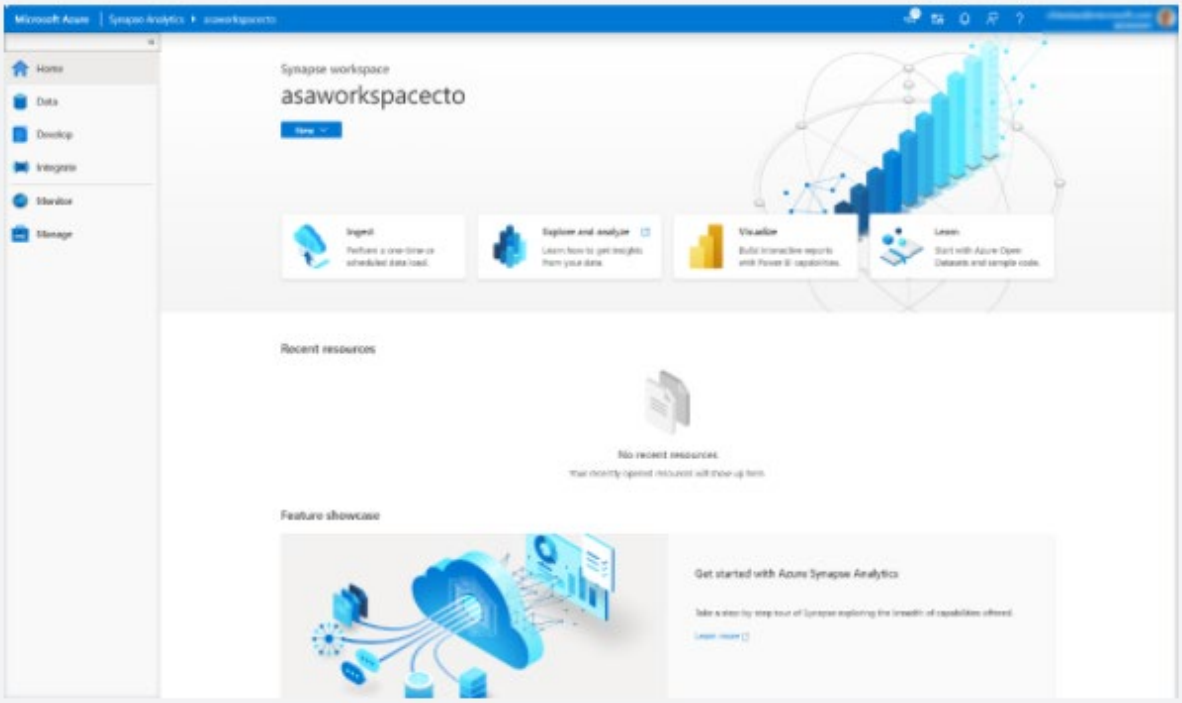
Azure Synapse Studio is the primary tool to use to interact with the many components that exist in the service. It organizes itself into hubs which allow you to perform a wide range of activities against your data.

Which of the following are the referenced hubs on Azure Synapse Studio? (Select six)

- ☐ Home **(Correct)**
- ☐ Integrate **(Correct)**
- ☐ Develop **(Correct)**
- ☐ Explore and analyze
- ☐ Ingest
- ☐ Data **(Correct)**
- ☐ Manage **(Correct)**
- ☐ Import
- ☐ Connect BI
- ☐ Monitor **(Correct)**

Explanation

Azure Synapse Studio is the primary tool to use to interact with the many components that exist in the service. It organizes itself into hubs, as seen on the left-hand side of the Azure Synapse Studio UI, which allow you to perform a wide range of activities against your data.



The following hubs are available within Azure Synapse Studio.

Home

The home hub contains short cuts that enable you to ingest, explore, analyze, and visualize your data. These provide shortcuts to tools such as the Copy Data Tool for ingesting data, to connecting to a Power BI workspace for visualization. You will also find links to resources that such as the documentation and pricing page. It will also list any resources you recently accessed, or pinned as favourite.

Data

The data hub can be accessed by either clicking on the Explore link in the home hub, or by selecting data on the left of the application. In this hub, you can access your provisioned SQL pool databases and SQL serverless databases in your workspace, as well as external data sources, such as storage accounts and other linked services. You also can preview data tables and data files.

Develop

The Develop hub is where you manage SQL scripts, Synapse notebooks, data flows, and Power BI reports. It can also be accessed by clicking on the Analyze icon in the home page.

Integrate

Manage data integration pipelines within the Integrate hub. If you are familiar with Azure Data Factory, then you will feel at home in this hub. The pipeline creation experience is the same as in Azure Data Factory, which gives you another powerful integration built into Synapse Analytics, removing the need to use Azure Data Factory separately for data movement and transformation pipelines.

Monitor

Use the Monitor hub to view pipeline and trigger runs, view the status of the various integration runtimes that are running, view Apache Spark jobs, SQL requests, and data flow debug activities. If you want to see the status of a job or activity, this is where you want to go.

The Monitor hub is your first stop for debugging issues and gaining insight on resource usage. You can see a history of all the activities taking place in the workspace and which ones are active now.

Manage

The Manage hub enables you to perform some of the same actions as 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.

https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-power-bi

---

Question 11: Skipped

In which modes does Azure Databricks provide data encryption?

- At-rest only
- None of the listed options.
- At-rest and in-transit
  (Correct)
- In-transit only

Explanation

Data stored in Azure Storage is encrypted using server-side encryption that is seamlessly accessed by Azure Databricks. All data transmitted between the Data Plane and the Control Plane is always encrypted in-flight via TLS.

https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption

---

Question 12: Skipped

Scenario: You are working as a consultant at Avengers Security. At the moment, you are consulting with Tony, the lead of the IT team and the topic of discussion is about a table in an enterprise data warehouse in Azure Synapse Analytics. See the subject table details below.

Table Characteristics:

• The file name is `SalesHistory`.

• The table contains sales data from the past 36 months.

• The file is partitioned by month.

• The file contains 1.5 billion rows.

• The file has clustered columnstore indexes.

Required:

At the beginning of each month, data older than 36 months must be promptly removed from the `SalesHistory` file.

The IT team has created a list of possible actions that should be performed to create a stored procedure, but there is debate about which are valid and the required sequence of the tabled actions. See the proposed action list below.

Proposed Actions:

a. Create an empty table named `SalesHistory_Current` that has a duplicate schema as the `SalesHistory` table.

b. Drop the `SalesHistory_Current` table.

c. Copy the data to a new table by using `CREATE TABLE AS SELECT`.

d. `TRUNCATE` the partition containing the stale data.

e. Switch the partition containing the stale data from `SalesHistory` to `SalesHistory_Current`.

f. Execute a `DELETE` statement where the value in the Date column is more than 36 months ago.

As the Azure SME, Tony and the team look to you to select the correct actions and put them in order in preparation for creation of the stored procedure. Which of the below contains the correct items in the correct sequence for the required stored procedure?

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

Explanation

Step 1: *Create an empty table named S* `alesHistory_Current` *that has a duplicate schema as the* `SalesHistory` *table.*

Step 2: *Switch the partition containing the stale data from* `SalesHistory` *to* `SalesHistory_Current`.

SQL Data Warehouse supports partition splitting, merging, and switching. To switch partitions between two tables, you must ensure that the partitions align on their respective boundaries and that the table definitions match.

Loading data into partitions with partition switching is a convenient way stage new data in a table that is not visible to users the switch in the new data.

Step 3: *Drop the* `SalesHistory_Current` *table.*

Partitioning tables in dedicated SQL pool

Table partitions enable you to divide your data into smaller groups of data. In most cases, table partitions are created on a date column. Partitioning is supported on all dedicated SQL pool table types; including clustered columnstore, clustered index, and heap. Partitioning is also supported on all distribution types, including both hash or round robin distributed.

Partitioning can benefit data maintenance and query performance. Whether it benefits both or just one is dependent on how data is loaded and whether the same column can be used for both purposes, since partitioning can only be done on one column.

Benefits to loads

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

Partition switching can be used to quickly remove or replace a section of a table. For example, a sales fact table might contain just data for the past 36 months. At the end of every month, the oldest month of sales data is deleted from the table. This data could be deleted by using a delete statement to delete the data for the oldest month.

However, deleting a large amount of data row-by-row with a delete statement can take too much time, as well as create the risk of large transactions that take a long time to rollback if something goes wrong. A more optimal approach is to drop the oldest partition of data. Where deleting the individual rows could take hours, deleting an entire partition could take seconds.

Benefits to queries

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

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

Sizing partitions

While partitioning can be used to improve performance some scenarios, creating a table with **too many** partitions can hurt performance under some circumstances. These concerns are especially true for clustered columnstore tables.

For partitioning to be helpful, it is important to understand when to use partitioning and the number of partitions to create. There is no hard fast rule as to how many partitions are too many, it depends on your data and how many partitions you loading simultaneously. A successful partitioning scheme usually has tens to hundreds of partitions, not thousands.

When creating partitions on **clustered columnstore** tables, it is important to consider how many rows belong to each partition. For optimal compression and performance of clustered columnstore tables, a minimum of 1 million rows per distribution and partition is needed. Before partitions are created, dedicated SQL pool already divides each table into 60 distributed databases.

Any partitioning added to a table is in addition to the distributions created behind the scenes. Using this example, if the sales fact table contained 36 monthly partitions, and given that a dedicated SQL pool has 60 distributions, then the sales fact table should contain 60 million rows per month, or 2.1 billion rows when all months are populated. If a table contains fewer than the recommended minimum number of rows per partition, consider using fewer partitions in order to increase the number of rows per partition.

How to split a partition that contains data

The most efficient method to split a partition that already contains data is to use a `CTAS` statement. If the partitioned table is a clustered columnstore, then the table partition must be empty before it can be split.

The following example creates a partitioned columnstore table. It inserts one row into each partition:

```
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])
,   PARTITION   (   [OrderDateKey] RANGE RIGHT FOR VALUES
                    (20000101
                    )
                )
)
;

INSERT INTO dbo.FactInternetSales
VALUES (1,19990101,1,1,1,1,1,1);
INSERT INTO dbo.FactInternetSales
VALUES (1,20000101,1,1,1,1,1,1);
```

The following query finds the row count by using the `sys.partitions` catalogue view:

```
SQL
SELECT  QUOTENAME(s.[name])+'.'+QUOTENAME(t.[name]) as Table_name
,       i.[name] as Index_name
,       p.partition_number as Partition_nmbr
,       p.[rows] as Row_count
,       p.[data_compression_desc] as Data_Compression_desc
FROM    sys.partitions p
JOIN    sys.tables     t    ON    p.[object_id]   = t.[object_id]
JOIN    sys.schemas    s    ON    t.[schema_id]   = s.[schema_id]
JOIN    sys.indexes    i    ON    p.[object_id]   = i.[object_Id]
                             AND   p.[index_Id]    = i.[index_Id]
WHERE t.[name] = 'FactInternetSales'
;
```

The following split command receives an error message:

```
SQL
ALTER TABLE FactInternetSales SPLIT RANGE (20010101);
```

Msg 35346, Level 15, State 1, Line 44 `SPLIT` clause of `ALTER PARTITION` statement failed because the partition is not empty. Only empty partitions can be split in when a columnstore index exists on the table. Consider disabling the columnstore index before issuing the `ALTER PARTITION` statement, then rebuilding the columnstore index after `ALTER PARTITION` is complete.

However, you can use `CTAS` to create a new table to hold the data.

```
SQL

CREATE TABLE dbo.FactInternetSales_20000101

    WITH    (   DISTRIBUTION = HASH(ProductKey)

            ,   CLUSTERED COLUMNSTORE INDEX

            ,   PARTITION   (   [OrderDateKey] RANGE RIGHT FOR VALUES

                                (20000101

                                )

                            )

            )

AS

SELECT *

FROM    FactInternetSales

WHERE   1=2

;
```

As the partition boundaries are aligned, a switch is permitted. This will leave the source table with an empty partition that you can subsequently split.

```
SQL

ALTER TABLE FactInternetSales SWITCH PARTITION 2 TO  FactInternetSales_20000101 P
ARTITION 2;


ALTER TABLE FactInternetSales SPLIT RANGE (20010101);
```

All that is left is to align the data to the new partition boundaries using `CTAS`, and then switch the data back into the main table.

```
SQL

CREATE TABLE [dbo].[FactInternetSales_20000101_20010101]

    WITH    (   DISTRIBUTION = HASH([ProductKey])
```

```
            ,   CLUSTERED COLUMNSTORE INDEX
            ,   PARTITION   (   [OrderDateKey] RANGE RIGHT FOR VALUES
                                (20000101,20010101
                                )
                            )
            )
AS
SELECT  *
FROM    [dbo].[FactInternetSales_20000101]
WHERE   [OrderDateKey] >= 20000101
AND     [OrderDateKey] <  20010101
;


ALTER TABLE dbo.FactInternetSales_20000101_20010101 SWITCH PARTITION 2 TO dbo.Fac
tInternetSales PARTITION 2;
```

Once you have completed the movement of the data, it is a good idea to refresh the statistics on the target table. Updating statistics ensures the statistics accurately reflect the new distribution of the data in their respective partitions.

```
SQL

UPDATE STATISTICS [dbo].[FactInternetSales];
```

Load new data into partitions that contain data in one step

Loading data into partitions with partition switching is a convenient way to stage new data in a table that is not visible to users. It can be challenging on busy systems to deal with the locking contention associated with partition switching.

To clear out the existing data in a partition, an `ALTER TABLE` used to be required to switch out the data. Then another `ALTER TABLE` was required to switch in the new data.

In dedicated SQL pool, the `TRUNCATE_TARGET` option is supported in the `ALTER TABLE` command. With `TRUNCATE_TARGET` the `ALTER TABLE` command overwrites existing data in the partition with new data. Below is an example that uses `CTAS` to create a new table with the existing data, inserts new data, then switches all the data back into the target table, overwriting the existing data.

```
SQL
```

```
CREATE TABLE [dbo].[FactInternetSales_NewSales]
    WITH    (   DISTRIBUTION = HASH([ProductKey])
            ,   CLUSTERED COLUMNSTORE INDEX
            ,   PARTITION   (   [OrderDateKey] RANGE RIGHT FOR VALUES
                                (20000101,20010101
                                )
                            )
            )
AS
SELECT  *
FROM    [dbo].[FactInternetSales]
WHERE   [OrderDateKey] >= 20000101
AND     [OrderDateKey] <  20010101
;

INSERT INTO dbo.FactInternetSales_NewSales
VALUES (1,20000101,2,2,2,2,2,2);

ALTER TABLE dbo.FactInternetSales_NewSales SWITCH PARTITION 2 TO dbo.FactInternet
Sales PARTITION 2 WITH (TRUNCATE_TARGET = ON);
```

Table partitioning source control

To avoid your table definition from **rusting** in your source control system, you may want to consider the following approach:

1. Create the table as a partitioned table but with no partition values

```
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])
,   PARTITION   (   [OrderDateKey] RANGE RIGHT FOR VALUES () )
)
;
```

2. `SPLIT` the table as part of the deployment process:

```
SQL
-- Create a table containing the partition boundaries


CREATE TABLE #partitions
WITH
(
    LOCATION = USER_DB
,   DISTRIBUTION = HASH(ptn_no)
)
AS
SELECT  ptn_no
,       ROW_NUMBER() OVER (ORDER BY (ptn_no)) as seq_no
FROM    (
    SELECT CAST(20000101 AS INT) ptn_no
    UNION ALL
    SELECT CAST(20010101 AS INT)
    UNION ALL
    SELECT CAST(20020101 AS INT)
    UNION ALL
    SELECT CAST(20030101 AS INT)
```

```
    UNION ALL
    SELECT CAST(20040101 AS INT)
) a
;

 -- Iterate over the partition boundaries and split the table

DECLARE @c INT = (SELECT COUNT(*) FROM #partitions)
,       @i INT = 1                                  --iterator for while loop
,       @q NVARCHAR(4000)                           --query
,       @p NVARCHAR(20)     = N''                   --partition_number
,       @s NVARCHAR(128)    = N'dbo'                --schema
,       @t NVARCHAR(128)    = N'FactInternetSales' --table
;

WHILE @i <= @c
BEGIN
    SET @p = (SELECT ptn_no FROM #partitions WHERE seq_no = @i);
    SET @q = (SELECT N'ALTER TABLE '+@s+N'.'+@t+N' SPLIT RANGE ('+@p+N');');

    -- PRINT @q;
    EXECUTE sp_executesql @q;
    SET @i+=1;
END

 -- Code clean-up

DROP TABLE #partitions;
```

With this approach, the code in source control remains static and the partitioning boundary values are allowed to be dynamic; evolving with the SQL pool over time.

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

Question 13: Skipped

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

[?] is a fully-managed, cloud-based Big Data and Machine Learning platform, which empowers developers to accelerate AI and innovation by simplifying the process of building enterprise-grade production data applications.

- Azure Databricks
  (Correct)
- Apache Kafka
- Azure Event Hub
- Apache Spark

Explanation
Azure Databricks is a fully-managed, cloud-based Big Data and Machine Learning platform, which empowers developers to accelerate AI and innovation by simplifying the process of building enterprise-grade production data applications. Built as a joint effort by the team that started Apache Spark and Microsoft, Azure Databricks provides data science and engineering teams with a single platform for Big Data processing and Machine Learning.

By combining the power of Databricks, an end-to-end, managed Apache Spark platform optimized for the cloud, with the enterprise scale and security of Microsoft's Azure platform, Azure Databricks makes it simple to run large-scale Spark workloads.

Optimized environment

To address the problems seen on other Big Data platforms, Azure Databricks was optimized from the ground up, with a focus on performance and cost-efficiency in the cloud. The Databricks Runtime adds several key capabilities to Apache Spark workloads that can increase performance and reduce costs by as much as 10-100x when running on Azure, including:

• High-speed connectors to Azure storage services, such as Azure Blob Store and Azure Data Lake

• Auto-scaling and auto-termination of Spark clusters to minimize costs

• Caching

• Indexing

• Advanced query optimization

By providing an optimized, easy to provision and configure environment, Azure Databricks gives developers a performant, cost-effective platform that enables them to spend more time building applications, and less time focused on managing clusters and infrastructure.

Who is Databricks?

Databricks was founded by the creators of Apache Spark, Delta Lake, and MLflow.

Over 2000 global companies use the Databricks platform across big data & machine learning lifecycle.

Databricks Vision: Accelerate innovation by unifying data science, data engineering and business.

Databricks Solution: Big Data Analytics Platform

What does Databricks offer that is not Open-Source Spark?

• Databricks Workspace - Interactive Data Science & Collaboration

• Databricks Workflows - Production Jobs & Workflow Automation

• Databricks Runtime

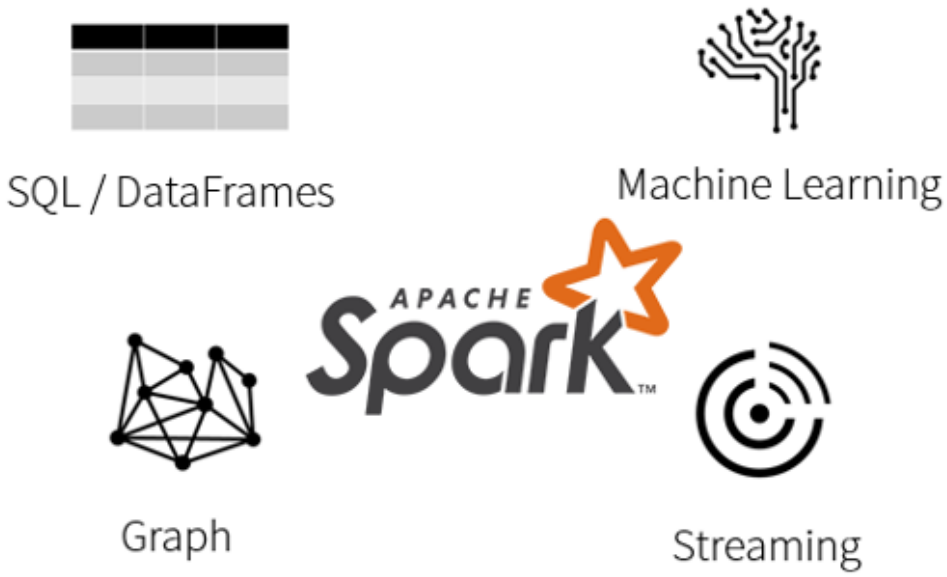
• Databricks I/O (DBIO) - Optimized Data Access Layer

• Databricks Serverless - Fully Managed Auto-Tuning Platform

• Databricks Enterprise Security (DBES) - End-To-End Security & Compliance

What is Apache Spark?

Spark is a unified processing engine that can analyze big data using SQL, machine learning, graph processing, or real-time stream analysis:



• At its core is the Spark Engine.

• The DataFrames API provides an abstraction above RDDs while simultaneously improving performance 5-20x over traditional RDDs with its Catalyst Optimizer.

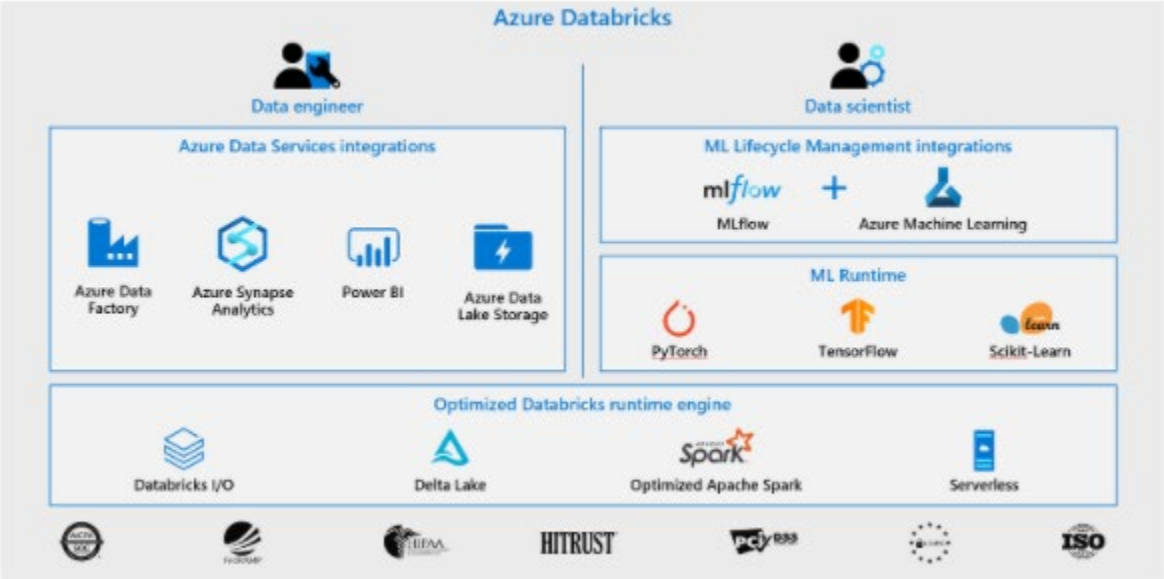
• Spark ML provides high quality and finely tuned machine learning algorithms for processing big data.

• The Graph processing API gives us an easily approachable API for modelling pairwise relationships between people, objects, or nodes in a network.

• The Streaming APIs give us End-to-End Fault Tolerance, with Exactly-Once semantics, and the possibility for sub-millisecond latency.

And it all works together seamlessly!

Azure Databricks



As a compute engine, Azure Databricks sits at the centre of your Azure-based software platform and provides native integration with Azure Active Directory (Azure AD) and other Azure services.

Scala, Python, Java, R & SQL

• Besides being able to run in many environments, Apache Spark makes the platform even more approachable by supporting multiple languages:

• Scala - Apache Spark's primary language

• Python - More commonly referred to as PySpark

• R - SparkR (R on Spark)

• Java

• SQL - Closer to ANSI SQL 2003 compliance

  • Now running all 99 TPC-DS queries

  • New standards-compliant parser (with good error messages!)

  • Subqueries (correlated & uncorrelated)

• Approximate aggregate stats

• With the DataFrames API, the performance differences between languages are nearly nonexistence (especially for Scala, Java & Python).

https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks

Question 14: Skipped

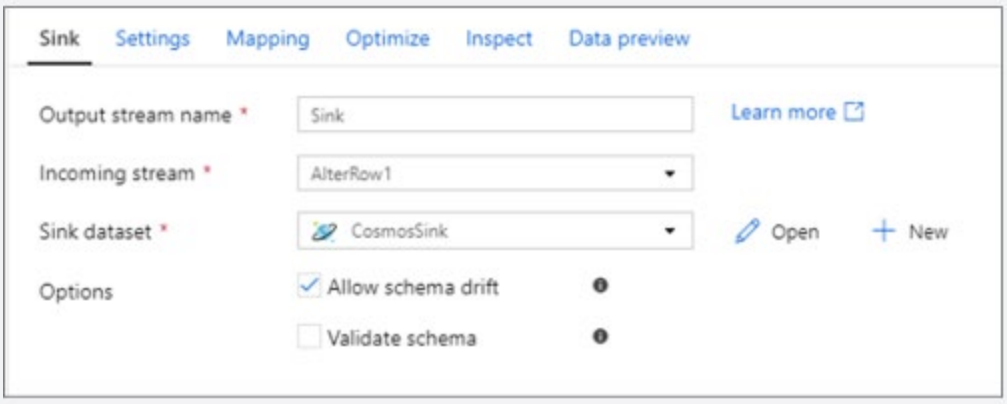
Which transformation is used to load data into a data store or compute resource?

- Source
- Cache
- Sink **(Correct)**
- Window
- Window
- Field
- Source

Explanation
A Sink transformation allows you to choose a dataset definition for the destination output data. You can have as many sink transformations as your data flow requires.

Every data flow requires at least one sink transformation, but you can write to as many sinks as necessary to complete your transformation flow. To write to additional sinks, create new streams via new branches and conditional splits.

Each sink transformation is associated with exactly one Azure Data Factory dataset object or linked service. The sink transformation determines the shape and location of the data you want to write to.



A Sink transformation allows you to choose a dataset definition for the destination output data. You can have as many sink transformations as your data flow requires.

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

---

Question 15: Skipped

Which of the below have the following characteristics?

• Permit us to store data in a format that more closely meets the original structure.

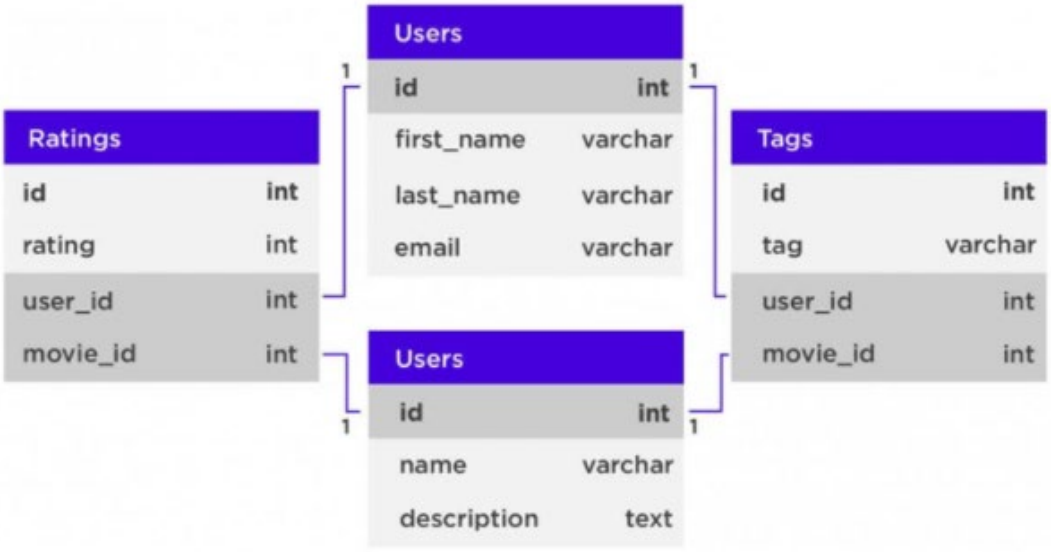
• Does not use the tabular schema of columns and rows found in most traditional database systems.

• Uses a storage model that is enhanced for the specific requirements of the type of data being stored.

- Relational
- Binary
- Structured
- Non-Relational **(Correct)**

Explanation
Non-Relational Data

• Non-relational databases permit us to store data in a format that more closely meets the original structure.



• A *non-relational database* is a database that does not use the tabular schema of columns and rows found in most traditional database systems.

• It uses a storage model that is enhanced for the specific requirements of the type of data being stored.

• In a non-relational database the data may be stored as **JSON documents**, as simple **key/value pairs**, or as a **graph** consisting of edges and vertices.

• Examples of relational databases:

• Redis

• JanusGraph

• MongoDB

• RabbitMQ

https://f5a395285c.nxcli.net/microsoft-azure/dp-900/structured-data-vs-unstructured-data-vs-semi-structured-data/

Question 16: Skipped

Scenario: O'Shaughnessy's is a fast food restaurant. The chain has stores nationwide and is rivalled by Big Belly Burgers. You have been hired by the company to advise on the creation and implementation of a dimension table in Azure Data Warehouse.

Specifications:

• The dimension table will be less than 1 GB.

Required:

- Fastest available query time
- Minimize data movement

As the Azure expert, the O'Shaughnessy IT team looks to you for direction. Which of the following should you advise them to utilize?

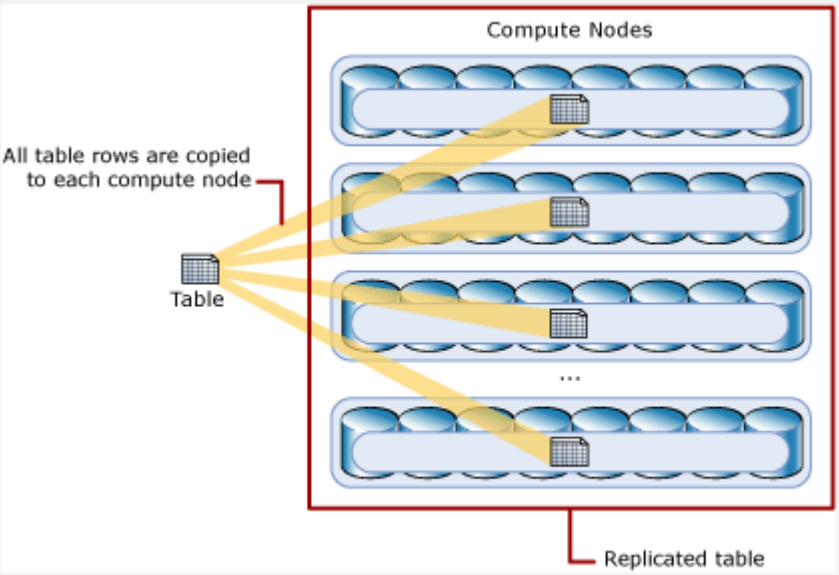
- Round-robin
- Hash distributed
- Heap
- Replicated
  (Correct)

Explanation
Replicated tables work well for dimension tables in a star schema. Dimension tables are typically joined to fact tables which are distributed differently than the dimension table. Dimensions are usually of a size that makes it feasible to store and maintain multiple copies. Dimensions store descriptive data that changes slowly, such as customer name and address, and product details. The slowly changing nature of the data leads to less maintenance of the replicated table.

What is a replicated table?

A replicated table has a full copy of the table accessible on each Compute node. Replicating a table removes the need to transfer data among Compute nodes before a join or aggregation. Since the table has multiple copies, replicated tables work best when the table size is less than 2 GB compressed. 2 GB is not a hard limit. If the data is static and does not change, you can replicate larger tables.

The following diagram shows a replicated table that is accessible on each Compute node. In SQL pool, the replicated table is fully copied to a distribution database on each compute node.



Replicated tables work well for dimension tables in a star schema. Dimension tables are typically joined to fact tables which are distributed differently than the dimension table. Dimensions are usually of a size that makes it feasible to store and maintain multiple copies. Dimensions store descriptive data that changes slowly, such as customer name and address, and product details. The slowly changing nature of the data leads to less maintenance of the replicated table.

Consider using a replicated table when:

• The table size on disk is less than 2 GB, regardless of the number of rows. To find the size of a table, you can use the `DBCC PDW_SHOWSPACEUSED` command: `DBCC PDW_SHOWSPACEUSED('ReplTableCandidate')`.

• The table is used in joins that would otherwise require data movement. When joining tables that are not distributed on the same column, such as a hash-distributed table to a round-robin table, data movement is required to complete the query. If one of the tables is small, consider a replicated table. We recommend using replicated tables instead of round-robin tables in most cases. To view data movement operations in query plans,

use `sys.dm_pdw_request_steps`. The `BroadcastMoveOperation` is the typical data movement operation that can be eliminated by using a replicated table.

Replicated tables may not yield the best query performance when:

• The table has frequent insert, update, and delete operations. The data manipulation language (DML) operations require a rebuild of the replicated table. Rebuilding frequently can cause slower performance.

• The SQL pool is scaled frequently. Scaling a SQL pool changes the number of Compute nodes, which incurs rebuilding the replicated table.

• The table has a large number of columns, but data operations typically access only a small number of columns. In this scenario, instead of replicating the entire table, it might be more effective to distribute the table, and then create an index on the frequently accessed columns. When a query requires data movement, SQL pool only moves data for the requested columns.

Use replicated tables with simple query predicates

Before you choose to distribute or replicate a table, think about the types of queries you plan to run against the table. Whenever possible,

• Use replicated tables for queries with simple query predicates, such as equality or inequality.

• Use distributed tables for queries with complex query predicates, such as LIKE or NOT LIKE.

CPU-intensive queries perform best when the work is distributed across all of the Compute nodes. For example, queries that run computations on each row of a table perform better on distributed tables than replicated tables. Since a replicated table is stored in full on each Compute node, a CPU-intensive query against a replicated table runs against the entire table on every Compute node. The extra computation can slow query performance.

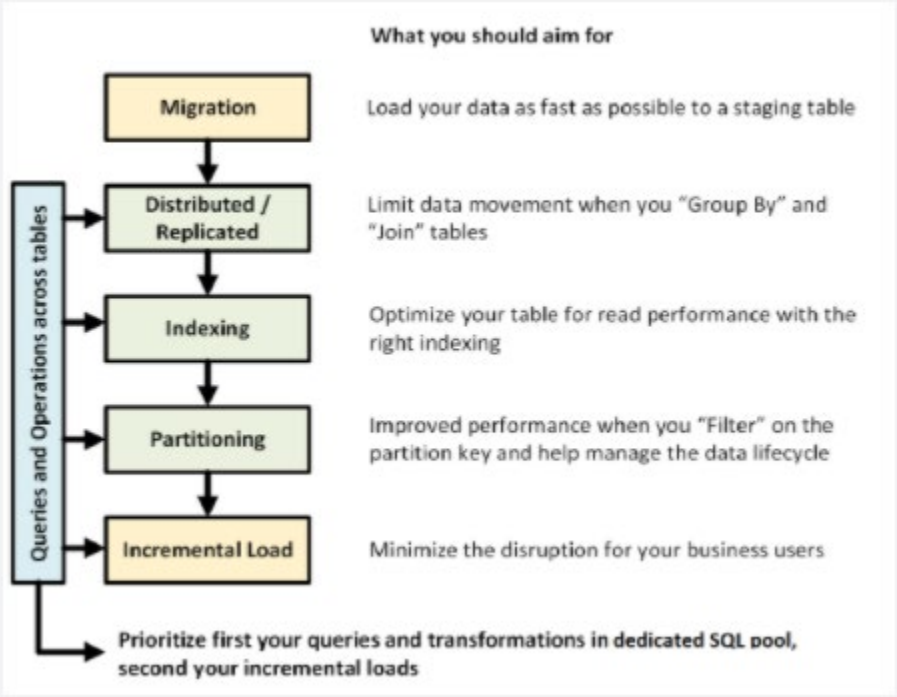
For example, this query has a complex predicate. It runs faster when the data is in a distributed table instead of a replicated table. In this example, the data can be round-robin distributed.

```
SQL

SELECT EnglishProductName

FROM DimProduct
```

```
WHERE EnglishDescription LIKE '%frame%comfortable%'
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/design-guidance-for-replicated-tables

The following graphic shows the process of designing a data warehouse with dedicated SQL pool



Data migration

First, load your data into Azure Data Lake Storage or Azure Blob Storage. Next, use the COPY statement to load your data into staging tables. Use the following configuration:

Design	Recommendation
Distribution	Round Robin
Indexing	Heap
Partitioning	None
Resource Class	largerc or xlargerc

Distributed or replicated tables

Use the following strategies, depending on the table properties:

Type	Great fit for...	Watch out if...
Replicated	* Small dimension tables in a star schema with less than 2 GB of storage after compression (~5x compression)	* Many write transactions are on table (such as insert, upsert, delete, update) * You change Data Warehouse Units (DWU) provisioning frequently * You only use 2-3 columns but your table has many columns * You index a replicated table
Round Robin (default)	* Temporary/staging table * No obvious joining key or good candidate column	* Performance is slow due to data movement
Hash	* Fact tables * Large dimension tables	* The distribution key cannot be updated

Tips:

• Start with Round Robin, but aspire to a hash distribution strategy to take advantage of a massively parallel architecture.

• Make sure that common hash keys have the same data format.

• Don't distribute on varchar format.

• Dimension tables with a common hash key to a fact table with frequent join operations can be hash distributed.

• Use *sys.dm_pdw_nodes_db_partition_stats* to analyze any skewness in the data.

• Use *sys.dm_pdw_request_steps* to analyze data movements behind queries, monitor the time broadcast, and shuffle operations take. This is helpful to review your distribution strategy.

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

---

Question 17: Skipped

There are two concepts within Apache Spark Pools in Azure Synapse Analytics, namely Spark Pools and Spark Instances.

Which of the following attributes belong to Spark Instances? (Select three)

- Exists as Metadata
- Permissions can be applied
- Created when connected to Spark Pool, Session, or Job **(Correct)**
- Multiple users can have access **(Correct)**
- Reusable **(Correct)**
- Creates a Spark Instance

Explanation
There are two concepts within Apache Spark Pools in Azure Synapse Analytics, namely Spark pools and Spark Instances. In short, they do the following:

Spark Pools:

• Exists as Metadata

• Creates a Spark Instance

• No costs associated with creating Pool

• Permissions can be applied

• Best practices

Spark Instances:

• Created when connected to Spark Pool, Session, or Job

• Multiple users can have access

• Reusable

A Spark pool is created in the Azure portal. It is 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; no resources are consumed, running, or charged for. A Spark pool has series of properties that control the characteristics of a Spark instance; these characteristics include but are not limited to name, size, scaling behaviour, time to live.

As there is no resource cost associated with creating Spark pools, any number of pools 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.

An example of Spark Pools:

• You create a Spark pool called SP1; it has a fixed cluster size of 20 nodes.

• You submit a notebook job, J1 that uses 10 nodes, a Spark instance, SI1 is created to process the job.

• You now submit another job, J2, that uses 10 nodes because there is still capacity in the pool and the instance, the J2, is processed by SI1.

• If J2 had asked for 11 nodes, there would not have been capacity in SP1 or SI1. In this case, if J2 comes from a notebook, then the job will be rejected; if J2 comes from a batch job, then it will be queued.

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, then if there is capacity in the pool, the existing Spark instance also has capacity then the existing instance will process the job; if not and there is capacity at the pool level, then a new Spark instance will be created.

An example of a Spark Instance:

• You create a Spark pool call SP2; it has an autoscale enabled 10 – 20 nodes

• You submit a notebook job, J1 that uses 10 nodes, a Spark instance, SI1, is created to process the job.

• You now submit another job, J2, that uses 10 nodes, because there is still capacity in the pool the instance auto grows to 20 nodes and processes J2.

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

---

Question 18: Skipped

Azure Data factory can accommodate organizations that are embarking on data integration projects from differing starting point. Typically, many data integration workflows must consider existing pipelines that have been created on previous projects, with different dependencies and using different technologies.

Which of the following are ingestion methods that can be used to extract data from a variety of sources?

- ☐ Self-hosted
- ☐ Copy Activity **(Correct)**
- ☐ SSIS packages **(Correct)**
- ☐ Datasets
- ☐ Linked Services
- ☐ Compute resources **(Correct)**
- ☐ Activities

Explanation

Azure Data factory can accommodate organizations that are embarking on data integration projects from differing starting point. It is rare for a data migration project to be a green field project. Typically, many data integration workflows must consider existing pipelines that have been created on previous projects, with different dependencies and using different technologies. To that end, there are a variety of ingestion methods that can be used to extract data from a variety of sources.

Ingesting data using the Copy Activity

Use this method to build code free data ingestion pipelines that don't require any transformation during the extraction of the data. The Copy Activity has support for over 100 native connectors. This method can suit green field projects that have a simple method of extraction to an intermediary data store. An example of ingesting data using the Copy Activity can include extracting data from multiple source database systems

and outputting the data to files in a data lake store. The benefit of this ingestion method is that they are simple to create, but they are not able to deal with sophisticated transformations or business logic.

Ingesting data using compute resources

Azure Data Factory can call on compute resources to process data by a data platform service that may be better suited to the job. A great example of this is that Azure Data Factory can create a pipeline to an analytical data platform such as Spark pools in an Azure Synapse Analytics instance to perform a complex calculation, which generates new data. This data is then ingested back into the pipeline for further downstream processing. There a wide range of compute resource, and the associated activities that they can perform as shown in the following:

Compute environment: On-demand HDInsight cluster or your own HDInsight cluster

Activities: Hive, Pig, Spark, MapReduce, Hadoop Streaming

Compute environment: Azure Batch

Activities: Custom activities

Compute environment: Azure Machine Learning Studio Machine

Activities: Learning activities: Batch Execution and Update Resource

Compute environment: Azure Machine Learning

Activities: Azure Machine Learning Execute Pipeline

Compute environment: Azure Data Lake Analytics

Activities: Data Lake Analytics U-SQL

Compute environment: Azure SQL, Azure SQL Data Warehouse, SQL Server

Activities: Stored Procedure

Compute environment: Azure Databricks

Activities: Notebook, Jar, Python

Compute environment: Azure Function

Activities: Azure Function activity

Ingesting data using SSIS packages

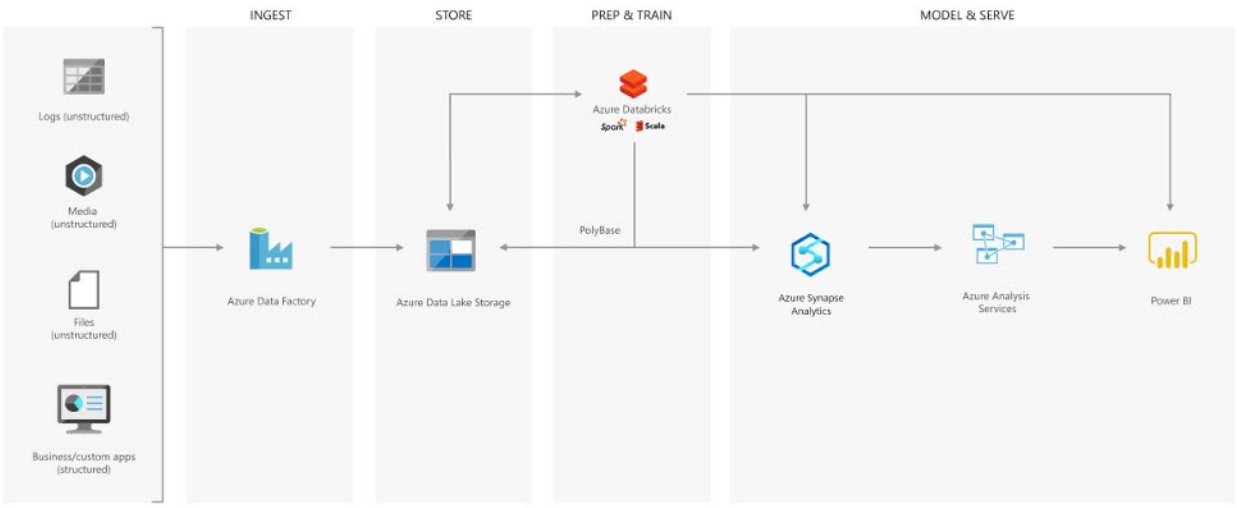
Many organizations have decades of development investment in SQL Server Integration Services (SSIS) packages that contain both ingestion and transformation logic from on-premises and cloud data stores. Azure Data Factory provides the ability to lift and shift existing SSIS workload, by creating an Azure-SSIS Integration Runtime to natively execute SSIS packages, and will enable you to deploy and manage your existing SSIS packages with little to no change using familiar tools such as SQL Server Data Tools (SSDT) and SQL Server Management Studio (SSMS), just like using SSIS on premises.

https://docs.microsoft.com/en-us/azure/machine-learning/how-to-data-ingest-adf

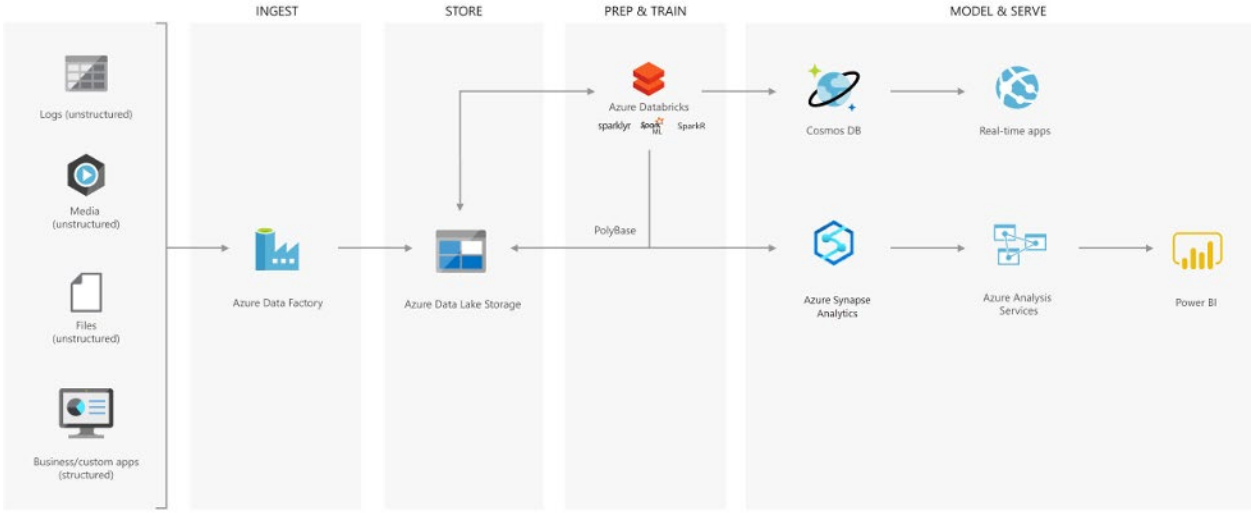
Question 19: Skipped

Scenario: Azure Data Lake Storage plays an important role in providing a large-scale data store. Your skills are needed by Hydra Corporation, which is a global seller of bicycles and cycling components through a chain of resellers and on the internet. As their customers browse the product catalogue on their websites and add items to their baskets, a recommendation engine that is built into Azure Databricks recommends other products. They need to make sure that the results of their recommendation engine can scale globally. The recommendations are based on the web log files that are stored on the web servers and transferred to the Azure Databricks model hourly. The response time for the recommendation should be less than 1 ms. Review the following architecture designs.

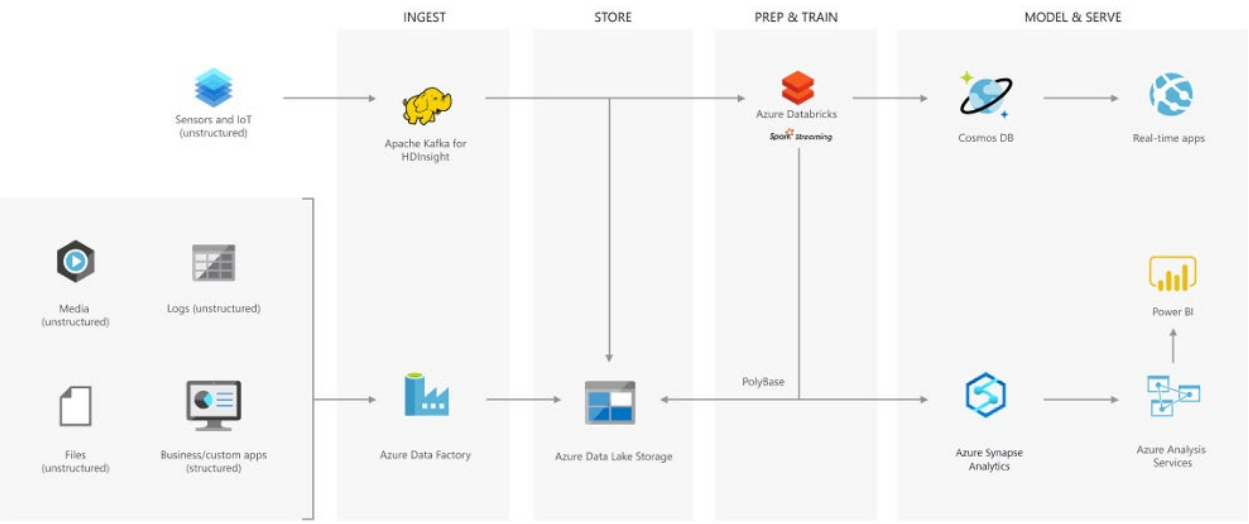
Design A:



Design B:



Design C:



Which architecture would be best suited for the need?

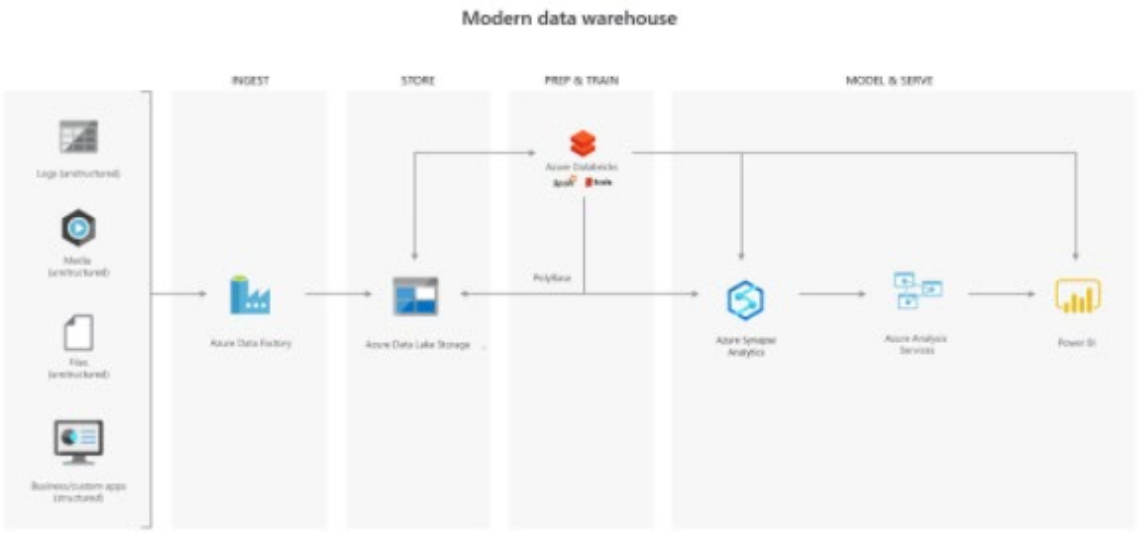
- None of the listed options
- Design C
- Design B
  (Correct)
- Design A

Explanation
Creating a modern data warehouse

Imagine you're a Data Engineering consultant for a Avengers Security. In the past, they've created an on-premises business intelligence solution that used a Microsoft SQL Server Database Engine, SQL Server Integration Services, SQL Server Analysis Services, and SQL Server Reporting Services to provide historical reports. They tried using the Analysis Services Data Mining component to create a predictive analytics solution to predict the buying behaviour of customers. While this approach worked well with low volumes of data, it couldn't scale after more than a gigabyte of data was collected. Furthermore, they were never able to deal with the JSON data that a third-party application generated when a customer used the feedback module of the point of sale (POS) application.

The company has turned to you for help with creating an architecture that can scale with the data needs that are required to create a predictive model and to handle the

JSON data so that it's integrated into the BI solution. You suggest the following architecture:



The architecture uses Azure Data Lake Storage at the centre of the solution for a modern data warehouse. Integration Services is replaced by Azure Data Factory to ingest data into the Data Lake from a business application. This is the source for the predictive model that is built into Azure Databricks. PolyBase is used to transfer the historical data into a big data relational format that is held in Azure Synapse Analytics, which also stores the results of the trained model from Databricks. Azure Analysis Services provides the caching capability for SQL Data Warehouse to service many users and to present the data through Power BI reports.

Advanced analytics for big data

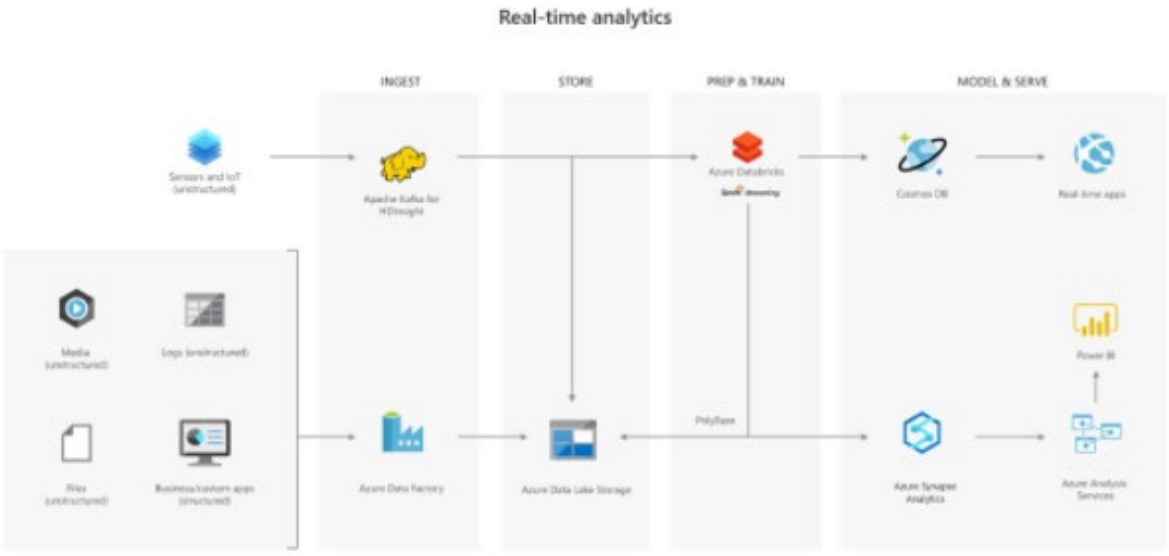
In this second use case, Azure Data Lake Storage plays an important role in providing a large-scale data store. Your skills are needed by Hydra Corporation, which is a global seller of bicycles and cycling components through a chain of resellers and on the internet. As their customers browse the product catalogue on their websites and add items to their baskets, a recommendation engine that is built into Azure Databricks recommends other products. They need to make sure that the results of their recommendation engine can scale globally. The recommendations are based on the web log files that are stored on the web servers and transferred to the Azure Databricks model hourly. The response time for the recommendation should be less than 1 ms. You propose the following architecture:

Real-time analytical solutions

To perform real-time analytical solutions, the ingestion phase of the architecture is changed for processing big data solutions. In this architecture, note the introduction of Apache Kafka for Azure HDInsight to ingest streaming data from an Internet of Things (IoT) device, although this could be replaced with Azure IoT Hub and Azure Stream Analytics. The key point is that the data is persisted in Data Lake Storage Gen2 to service other parts of the solution.

In this use case, you are a Data Engineer for HAMMER Industries, an organization that is working with a transport company to monitor the fleet of Heavy Goods Vehicles (HGV) that drive around Europe. Each HGV is equipped with sensor hardware that will continuously report metric data on the temperature, the speed, and the oil and brake solution levels of an HGV. When the engine is turned off, the sensor also outputs a file with summary information about a trip, including the mileage and elevation of a trip. A trip is a period in which the HGV engine is turned on and off.

Both the real-time data and batch data is processed in a machine learning model to predict a maintenance schedule for each of the HGVs. This data is made available to the downstream application that third-party garage companies can use if an HGV breaks down anywhere in Europe. In addition, historical reports about the HGV should be visually presented to users. As a result, the following architecture is proposed:



In this architecture, there are two ingestion streams. Azure Data Factory ingests the summary files that are generated when the HGV engine is turned off. Apache Kafka provides the real-time ingestion engine for the telemetry data. Both data streams are stored in Azure Data Lake Store for use in the future, but they are also passed on to other technologies to meet business needs. Both streaming and batch data are provided to the predictive model in Azure Databricks, and the results are published to Azure Cosmos DB to be used by the third-party garages. PolyBase transfers data from the Data Lake Store into SQL Data Warehouse where Azure Analysis Services creates the HGV reports by using Power BI.

https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

---

Question 20: Skipped

Which type of transactional database system would work best for product data?

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

Explanation

OLTP systems support a large set of users, have quick response times, handle large volumes of data, are highly available, and are great for small or relatively simple transactions.

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 21: Skipped

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

[?] is an encryption mechanism to help you protect Azure Synapse Analytics. It will protect Azure Synapse Analytics against threats of malicious offline activity. The [?] way will do so by is encrypting data at rest. [?] performs real-time encryption as well as decryption of the database, associated backups, and transaction log files at rest without you having to make changes to the application.

- Column-level security
- Database Encryption Key
- Dynamic Data Masking
- Table-level security
- Transparent Data Encryption **(Correct)**
- Row-level security

Explanation
Transparent Data Encryption

Transparent Data Encryption (TDE) is an encryption mechanism to help you protect Azure Synapse Analytics. It will protect Azure Synapse Analytics against threats of malicious offline activity. The way TDE will do so, is by encrypting data at rest. TDE performs real-time encryption as well as decryption of the database, associated backups, and transaction log files at rest without you having to make changes to the application. In order to use TDE for Azure Synapse Analytics, you will have to manually enable it.

What TDE does is performing I/O encryption and decryption of data at the page level in real time. When a page is read into memory, it is decrypted. It is encrypted before writing it to disk. TDE encrypts the entire data base storage, using a symmetric key called a Database Encryption Key (DEK). When you start up a database, the encrypted Database Encryption Key is decrypted when it then will be used for decryption and re-encryption of the database files in the SQL Server database engine. The DEK is protected by the Transparent Data Encryption Protector. This protector can be either a service-managed certificated, which is referred to as service-managed transparent data

encryption, or an asymmetric key that is stored in Azure Key Vault (customer-managed transparent data encryption).

What is important to understand is that for Azure Synapse Analytics, this TDE protector is set on the server level. There it is inherited by all the databases that are attached or aligned to that server. The term server refers both to server and instance.

Service-managed transparent data encryption

As stated above, the DEK that is protected by the Transparent Encryption protector can be service-managed certificated which we call service-managed TDE. When you look in Azure, that default setting means that the DEK is protected by a built-in certificate unique for each server with encryption algorithm AES256. When a database is in a geo-replicated relationship then primary and the geo-secondary database are protected by the primary database's parent server key. If the databases are connected to the same server, they will also have the same built-in AES 256 certificate. As Microsoft we automatically rotate the certificates in compliance with the internal security policy. The root key is protected by a Microsoft internal secret store. Microsoft also seamlessly moves and manages the keys as needed for geo-replication and restores.

Transparent data encryption with bring your own key for customer-managed transparent data encryption

As stated above, the DEK that is protected by the Transparent Data Encryption Protector can also be customer managed by bringing an asymmetric key that is stored in Azure Key Vault (customer-managed transparent data encryption). This is also referred to as Bring Your Own Key (BYOK) support for TDE. When this is the scenario that is applicable to you, the TDE Protector that encrypts the DEK is a customer-managed asymmetric key. This is stored in your own and managed Azure Key Vault. Azure Key Vault is Azure's cloud-based external key management system. This managed key never leaves the key vault. The TDE Protector can be generated by the key vault. Another option is to transfer the TDE Protector to the key vault from, for example, an on-premise hardware security module (HSM) device. Azure Synapse Analytics needs to be granted permissions to the customer-owned key vault in order to decrypt and encrypt the DEK. If permissions of the server to the key vault are revoked, a database will be inaccessible, and all data is encrypted.

By using Azure Key Vault integration for TDE, you have control over the key management tasks such as key rotations, key backups, and key permissions. It also enables you for auditing and reporting on all the TDE protectors when using the Azure Key Vault functionality. The reason for using Key Vault is that it provides you with a central key management system where tightly monitored HSMs are leveraged. It also enables you to separate duties of management of keys and data in order to meet compliance with security policies.

Moving a transparent data encryption protected database

In some use cases you need to move a database that is protected with TDE. Within Azure, there is no need to decrypt the databases. The TDE settings on the source database or primary database, will be inherited on the target. Some of the operations within Azure that inherited the TDE are:

• Geo-restore

• Self-service point-in-time restore

• Restoration of a deleted database

• Active geo-replication

• Creation of a database copy

• Restore of backup file to Azure SQL Managed Instance

If you export a TDE-protected database, the exported content is not encrypted. This will be stored in an unencrypted BACPAC file. You need to make sure that you protect this BACPAC file and enable TDE as soon as the import of the bacpac file in the new database is finished.

Securing your credentials through linked services with TokenLibrary for Apache Spark

It is quite a common pattern to access data from external sources. Unless the external data source allows anonymous access, it is highly likely that you need to secure your connection with a credential, secret, or connection string.

Within Azure Synapse Analytics, the integration process is simplified by providing linked services. Doing so, the connection details can be stored in the linked service or an Azure Key Vault. If the Linked Service is created, Apache spark can reference the linked service to apply the connection information in your code. When you want to access files from the Azure Data Lake Storage Gen 2 within your Azure Synapse Analytics Workspace, it uses AAD passthrough for the authentication. Therefore, there is no need to use the TokenLibrary. However, to connect to other linked services, you are enabled to make a direct call to the TokenLibrary.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-encryption-tde-tsql

---

Question 22: Skipped

Which method for renaming a `DataFrame` column is incorrect?

- All are correct.
- `df.select(col("timestamp").alias("dateCaptured"))`
- All are incorrect.
- `df.alias("timestamp", "dateCaptured")`
  (Correct)
- `df.toDF("dateCaptured")`

Explanation

The `DataFrame` does not contain an alias method for a column. `df.alias("timestamp", "dateCaptured")` is an incorrect method for renaming a `DataFrame` column.

https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-dataframes-python.html

---

Question 23: Skipped

Scenario: You need to provision a data store that will store but not query data.

Which is the least expensive option which will meet the requirement?

- Azure Queue Storage
- Azure File Storage
- Azure Blob Storage
  (Correct)
- Azure Database Server Storage
- Azure Table Storage
- Azure Cosmos DB Storage
- Azure Data Lake Storage

Explanation
When to use Blob Storage

If you need to provision a data store that will store but not query data, your cheapest option is to set up a storage account as a Blob store. Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data. Blob storage works well with images and unstructured data, and it's the cheapest way to store data in Azure.

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

---

Question 24: 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:

"The Union transformation that combines multiple data streams."

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

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 25: Skipped

Scenario: The Daily Bugle is a news organization led by J. Jonah Jameson which has a meager beginning and now has become a household name in news reporting. The company has grown well, despite some technology issues and now Jonah has hired you as an IT consultant to advise on several IT projects geared to improving the company's efficiencies.

One of the current projects is geared towards the design of an enterprise data warehouse in Azure Synapse Analytics.

Specifications:

• There will be millions of rows of data loaded to the data warehouse daily

• Staging tables must be optimized for data loading

Required:

• Design the staging tables

The team has come up with a list of options they are considering to use for the task at had but are not sure which to utilize. As you are the Azure expert, the team leader has asked for your opinion on which to use.

Which of the following should you recommend for the staging table design?

- Replicated table
- Hash-distributed table
- Round-robin distributed table **(Correct)**
- External table

Explanation

You should recommend the Round-robin for the staging table. The load with CTAS is fast. Once the data is in the staging table, use `INSERT...SELECT` *to move the data to production tables.*

Round-robin tables

A round-robin table distributes table rows evenly across all distributions. The rows are distributed randomly. Loading data into a round-robin table is fast. But, queries can require more data movement than the other distribution methods.

Common distribution methods for tables

The table category often determines the optimal option for table distribution.

Fact Table category

Use hash-distribution with clustered columnstore index. Performance improves when two hash tables are joined on the same distribution column.

Dimension Table category

Use replicated for smaller tables. If tables are too large to store on each Compute node, use hash-distributed.

Staging Table category

Use round-robin for the staging table. The load with CTAS is fast. Once the data is in the staging table, use `INSERT...SELECT` to move the data to production tables.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-overview

---

Question 26: Skipped

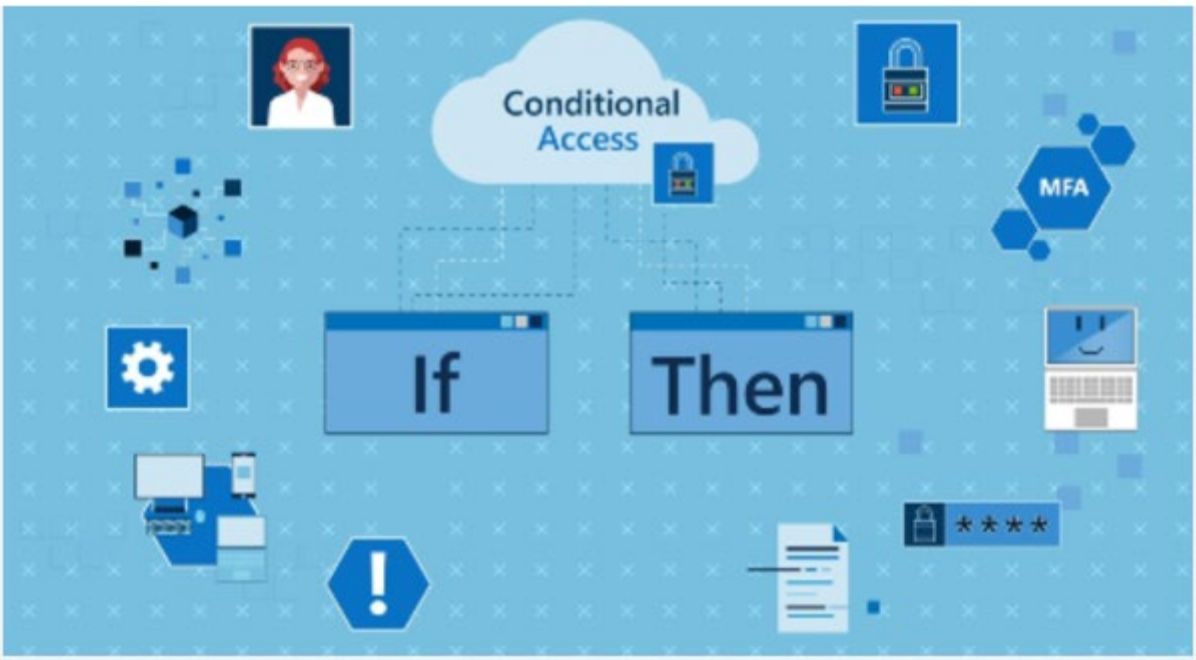
Conditional access is a feature that enables you to define the conditions under which a user can connect to your Azure subscription and access services. Conditional access provides an additional layer of security that can be used in combination with authentication to strengthen the security access to your network.

Conditional Access policies at their simplest are [?].

- Where-having statements
- While-if statements
- If-else statements
- If-then statements
  (Correct)

Explanation
Conditional access is a feature that enables you to define the conditions under which a user can connect to your Azure subscription and access services. Conditional access provides an additional layer of security that can be used in combination with authentication to strengthen the security access to your network.

Conditional Access policies at their simplest are if-then statements, if a user wants to access a resource, then they must complete an action. As an example, if a Data Engineer wishes to access services in Azure Synapse Analytics, they may be requested by the conditional access policy to perform an additional step of multi-factor authentication (MFA) to complete the authentication to get onto the service.



Conditional access policies use signals as a basis to determine if conditional access should first be applied. Common signals include:

• User or group membership names

• IP address information

• Device platforms or type

• Application access requests

• Real-time and calculated risk detection

• Microsoft Cloud App Security (MCAS)

Based on these signals, you can then choose to block access. The alternative is you can grant access, and at the same time request that the user perform an additional action including:

• Perform Multi-Factor authentication

• Use a specific device to connect

Given the amount of data that could potentially be stored, Azure Synapse Analytics dedicated SQL pools supports conditional access to provide protection for your data. It does require that Azure Synapse Analytics is configured to support Azure Active Directory, and that if you chose multi-factor authentication, that the tool you are using support it.

https://docs.microsoft.com/en-us/azure/azure-sql/database/conditional-access-configure

---

Question 27: Skipped

Scenario: ACME Books Inc uses Azure Cosmos DB to store user profile data from their eCommerce site. The NoSQL document store provided by the Azure Cosmos DB SQL API provides the familiarity of managing their data using SQL syntax, while being able to read and write the files at a massive, global scale.

While ACME is happy with the capabilities and performance of Azure Cosmos DB, they are concerned about the cost of executing a large volume of analytical queries over multiple partitions (cross-partition queries) from their data warehouse. They want to efficiently access all the data without needing to increase the Azure Cosmos DB request units (RUs).

They have looked at options for extracting data from their containers to the data lake as it changes, through the Azure Cosmos DB change feed mechanism. The problem with this approach is the extra service and code dependencies and long-term maintenance of the solution. They could perform bulk exports from a Synapse Pipeline, but then they won't have the most up-to-date information at any given moment.

Required: Pick a solution will ensure all transactional data is automatically stored in a fully isolated column store without impacting the transactional workloads or incurring resource unit (RU) costs.

- Enable Azure Synapse Link for Cosmos DB and enable the analytical store on their Azure Cosmos DB containers.
(Correct)
- Enable Azure Private Link for SQL Database and enable the analytical store on their SQL Database containers.
- Enable Spark Pools for SQL Datawarehouse and enable the analytical store on their Azure Cosmos DB containers.
- None of the listed options.

Explanation
The correct solution is to enable Azure Synapse Link for Cosmos DB and enable the analytical store on their Azure Cosmos DB containers. With this configuration, all transactional data is automatically stored in a fully isolated column store. This store enables large-scale analytics against the operational data in Azure Cosmos DB, without impacting the transactional workloads or incurring resource unit (RU) costs. Azure Synapse Link for Cosmos DB creates a tight integration between Azure Cosmos DB and Azure Synapse Analytics, which enables Tailwind Traders to run near real-time analytics

over their operational data with no-ETL and full performance isolation from their transactional workloads.

By combining the distributed scale of Cosmos DB's transactional processing with the built-in analytical store and the computing power of Azure Synapse Analytics, Azure Synapse Link enables a Hybrid Transactional/Analytical Processing (HTAP) architecture for optimizing Tailwind Trader's business processes. This integration eliminates ETL processes, enabling business analysts, data engineers & data scientists to self-serve and run near real-time BI, analytics, and Machine Learning pipelines over operational data.

Azure Synapse Link for Cosmos DB is a cloud native HTAP (Hybrid Transactional and Analytical Processing) capability that allows us to run near-real time analytics over our data in Azure Cosmos DB.

This is possible through the Azure Cosmos DB Analytical Store, which provides us a way to perform near real-time analytics on our data without have to engineer our own ETL pipelines to do so.

https://medium.com/swlh/building-near-real-time-analytics-with-azure-synapse-link-for-azure-cosmos-db-eba35e759e1c

Hybrid Transactional and Analytical Processing enables businesses to perform analytics over a database system that is seen to provide transactional capabilities without impacting the performance of the system. This enables organizations to use a database to fulfill both transactional and analytical needs to support near real-time analysis of operational data to make decisions about the information that is being analyzed.

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

---

Question 28: Skipped

As great as data lakes are at inexpensively storing our raw data, they also bring with them performance challenges:

• **Too many small or very big files** - more time opening & closing files rather than reading contents (worse with streaming).

• **Partitioning also known as "poor man's indexing"**- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

• **No caching** - cloud storage throughput is low (cloud object storage is 20-50MB/s/core vs 300MB/s/core for local SSDs).

As a solution to the challenges with Data Lakes noted above, Delta Lake is a file format that can help you build a data lake comprised of one or many tables in Delta Lake format. Delta Lake integrates tightly with Apache Spark, and uses an open format that is based on Parquet.

Two of the core features of Delta Lake are performing `UPSERT`s and Time Travel operations.

What does the `UPSERT` command do?

- The command will `INSERT` a column and if the column already exists, `UPDATE` the column.
- The command will `INSERT` a row and if the row already exists, `UPDATE` the row. **(Correct)**
- The command will `INSERT` a row and if the row already exists, append a new row in the table with an update notation.
- The command will `INSERT` a column and if the column already exists, add a new column in the table with an update notation.
- The command will `INSERT` a table and if the table already exists, `UPDATE` the table.

Explanation
Delta Lake is a transactional storage layer designed specifically to work with Apache Spark and Databricks File System (DBFS). At the core of Delta Lake is an optimized

Spark table. It stores your data as Apache Parquet files in DBFS and maintains a transaction log that efficiently tracks changes to the table.

Data lakes

A data lake is a storage repository that inexpensively stores a vast amount of raw data, both current and historical, in native formats such as XML, JSON, CSV, and Parquet. It may contain operational relational databases with live transactional data.

Enterprises have been spending millions of dollars getting data into data lakes with Apache Spark. The aspiration is to do data science and ML on all that data using Apache Spark.

But the data is not ready for data science & ML. The majority of these projects are failing due to unreliable data!

The challenge with data lakes

Why are these projects struggling with reliability and performance?

To extract meaningful information from a data lake, you must solve problems such as:

• Schema enforcement when new tables are introduced.

• Table repairs when any new data is inserted into the data lake.

• Frequent refreshes of metadata.

• Bottlenecks of small file sizes for distributed computations.

• Difficulty sorting data by an index if data is spread across many files and partitioned.

There are also data reliability challenges with data lakes:

• Failed production jobs leave data in corrupt state requiring tedious recovery.

• Lack of schema enforcement creates inconsistent and low quality data.

• Lack of consistency makes it almost impossible to mix appends and reads, batch and streaming.

As great as data lakes are at inexpensively storing our raw data, they also bring with them performance challenges:

• **Too many small or very big files** - more time opening & closing files rather than reading contents (worse with streaming).

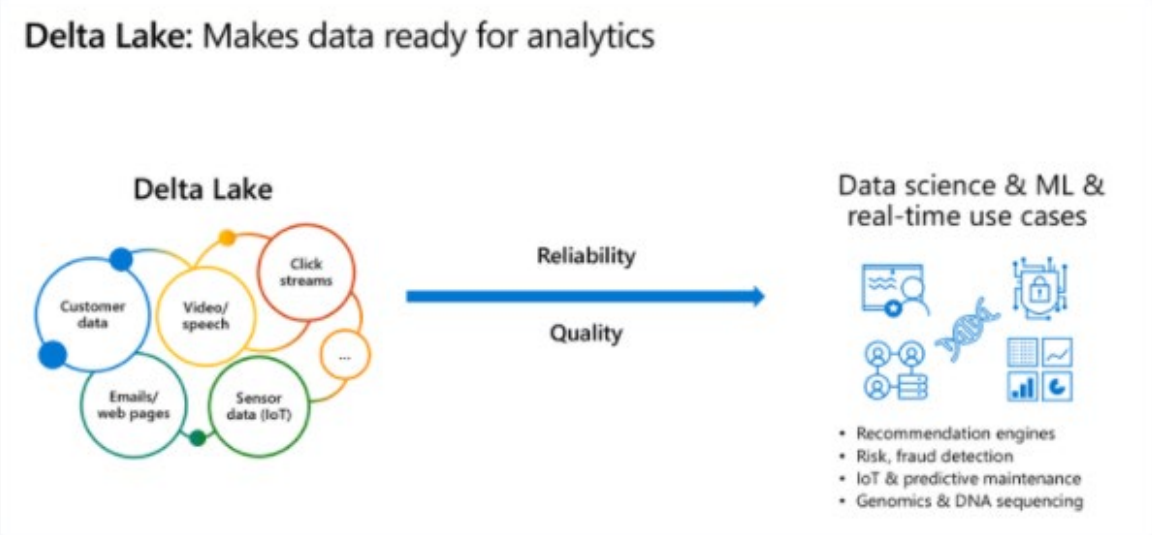
• **Partitioning also known as "poor man's indexing"**- breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

• **No caching** - cloud storage throughput is low (cloud object storage is 20-50MB/s/core vs 300MB/s/core for local SSDs).

The solution: Delta Lake

Delta Lake is a file format that can help you build a data lake comprised of one or many tables in Delta Lake format. Delta Lake integrates tightly with Apache Spark, and uses an open format that is based on Parquet. Because it is an open-source format, Delta Lake is also supported by other data platforms, including Azure Synapse Analytics.

Delta Lake makes data ready for analytics.



Delta Lake is an open-source storage layer that brings ACID transactions to Apache Spark™ and big data workloads.

You can read and write data that's stored in Delta Lake by using Apache Spark SQL batch and streaming APIs. These are the same familiar APIs that you use to work with Hive tables or DBFS directories. Delta Lake provides the following functionality:

ACID Transactions: Data lakes typically have multiple data pipelines reading and writing data concurrently, and data engineers have to go through a tedious process to ensure data integrity, due to the lack of transactions. Delta Lake brings ACID transactions to your data lakes. It provides serializability, the strongest level of isolation level.

Scalable Metadata Handling: In big data, even the metadata itself can be "big data". Delta Lake treats metadata just like data, leveraging Spark's distributed processing power to handle all its metadata. As a result, Delta Lake can handle petabyte-scale tables with billions of partitions and files at ease.

Time Travel (data versioning): Delta Lake provides snapshots of data enabling developers to access and revert to earlier versions of data for audits, rollbacks or to reproduce experiments.

Open Format: All data in Delta Lake is stored in Apache Parquet format enabling Delta Lake to leverage the efficient compression and encoding schemes that are native to Parquet.

Unified Batch and Streaming Source and Sink: A table in Delta Lake is both a batch table, as well as a streaming source and sink. Streaming data ingest, batch historic backfill, and interactive queries all just work out of the box.

Schema Enforcement: Delta Lake provides the ability to specify your schema and enforce it. This helps ensure that the data types are correct and required columns are present, preventing bad data from causing data corruption.

Schema Evolution: Big data is continuously changing. Delta Lake enables you to make changes to a table schema that can be applied automatically, without the need for cumbersome DDL.

100% Compatible with Apache Spark API: Developers can use Delta Lake with their existing data pipelines with minimal change as it is fully compatible with Spark, the commonly used big data processing engine.

Get started with Delta using Spark APIs

Delta Lake is included with Azure Databricks. You can start using it today. To quickly get started with Delta Lake, do the following:

Instead of parquet...

```
Python

CREATE TABLE ...

USING parquet
```

```
...

dataframe
.write
.format("parquet")
.save("/data")
... simply say delta
Python
CREATE TABLE ...
USING delta
...

dataframe
.write
.format("delta")
.save("/data")
```

Using Delta with your existing Parquet tables

Step 1: Convert Parquet to Delta tables:

```
Python
CONVERT TO DELTA parquet.`path/to/table` [NO STATISTICS]
[PARTITIONED BY (col_name1 col_type1, col_name2 col_type2, ...)]
Step 2: Optimize layout for fast queries:
Python
OPTIMIZE events
WHERE date >= current_timestamp() - INTERVAL 1 day
ZORDER BY (eventType)
```

Basic syntax

Two of the core features of Delta Lake are performing upserts (insert/updates) and Time Travel operations.

To `UPSERT` means to "UPdate" and "inSERT". In other words, `UPSERT` is literally TWO operations. It is not supported in traditional data lakes, as running an `UPDATE` could invalidate data that is accessed by the subsequent `INSERT` operation.

Using Delta Lake, however, we can do `UPSERTS`. Delta Lake combines these operations to guarantee atomicity to:

• `INSERT` a row

• if the row already exists, `UPDATE` the row.

Upsert syntax

Upserting, or merging, in Delta Lake provides fine-grained updates of your data. The following syntax shows how to perform an Upsert:

```
SQL
MERGE INTO customers -- Delta table
USING updates
ON customers.customerId = source.customerId
WHEN MATCHED THEN
UPDATE SET address = updates.address
WHEN NOT MATCHED
THEN INSERT (customerId, address) VALUES (updates.customerId, updates.address)
```

Time Travel syntax

Because Delta Lake is version controlled, you have the option to query past versions of the data. Using a single file storage system, you now have access to several versions your historical data, ensuring that your data analysts will be able to replicate their reports (and compare aggregate changes over time) and your data scientists will be able to replicate their experiments.

Other time travel use cases are:

• Re-creating analyses, reports, or outputs (for example, the output of a machine learning model). This could be useful for debugging or auditing, especially in regulated industries.

• Writing complex temporal queries.

• Fixing mistakes in your data.

• Providing snapshot isolation for a set of queries for fast changing tables.

Example of using time travel to reproduce experiments and reports:

```
SQL

SELECT count(*) FROM events

TIMESTAMP AS OF timestamp


SELECT count(*) FROM events

VERSION AS OF version

Python

spark.read.format("delta").option("timestampAsOf", timestamp_string).load("/event
s/")

If you need to rollback accidental or bad writes:

SQL

INSERT INTO my_table

SELECT * FROM my_table TIMESTAMP AS OF

date_sub( current_date(), 1)
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-what-is-delta-lake

Question 29: Skipped

When working with Azure Data Factory, before you create a dataset, you must create a Linked service to link your data store to the data factory.

You can programmatically define a linked service in the JSON format to be used via REST APIs or the SDK, using the following notation:

```
JSON
{
"name": "<Name of the linked service>",
"properties": {
"type": "<Type of the linked service>",
"typeProperties": {
"<data store or compute-specific type properties>"
},
"connectVia": {
"referenceName": "<name of Integration Runtime>",
"type": "IntegrationRuntimeReference"
}
}
}
```

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

- ☐ connectVia
- ☐ typeProperties **(Correct)**
- ☐ type **(Correct)**
- ☐ name **(Correct)**

Explanation
Linked Service

When working with Azure Data Factory, before you create a dataset, you must create a **Linked service** to link your data store to the data factory. Linked services are much like connection strings, which define the connection information needed for Data Factory to connect to external resources. There are over 100 connectors that can be used to define a linked service.

A linked service in Data Factory can be defined using the Copy Data Activity in the ADF designer, or you can create them independently to point to a data store or a compute

resources. The Copy Activity copies data between the source and destination, and when you run this activity you are asked to define a linked service as part of the copy activity definition

Alternatively you can programmatically define a linked service in the JSON format to be used via REST APIs or the SDK, using the following notation:

```
JSON
{
"name": "<Name of the linked service>",
"properties": {
"type": "<Type of the linked service>",
"typeProperties": {
"<data store or compute-specific type properties>"
},
"connectVia": {
"referenceName": "<name of Integration Runtime>",
"type": "IntegrationRuntimeReference"
}
}
}
```

The following describes properties in the above JSON:

Property: name

Name of the linked service.

Required: Yes

Property: type

Type of the linked service. For example: AzureStorage (data store) or AzureBatch (compute). See the description for typeProperties.

Required: Yes

Property: typeProperties

The type properties are different for each data store or compute. For the supported data store types and their type properties, see the dataset type table. Navigate to the data store connector article to learn about type properties specific to a data store.

Required: Yes

Property: connectVia

The Integration Runtime to be used to connect to the data store. You can use Azure Integration Runtime or Self-hosted Integration Runtime (if your data store is located in a private network). If not specified, it uses the default Azure Integration Runtime.

Required: No

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

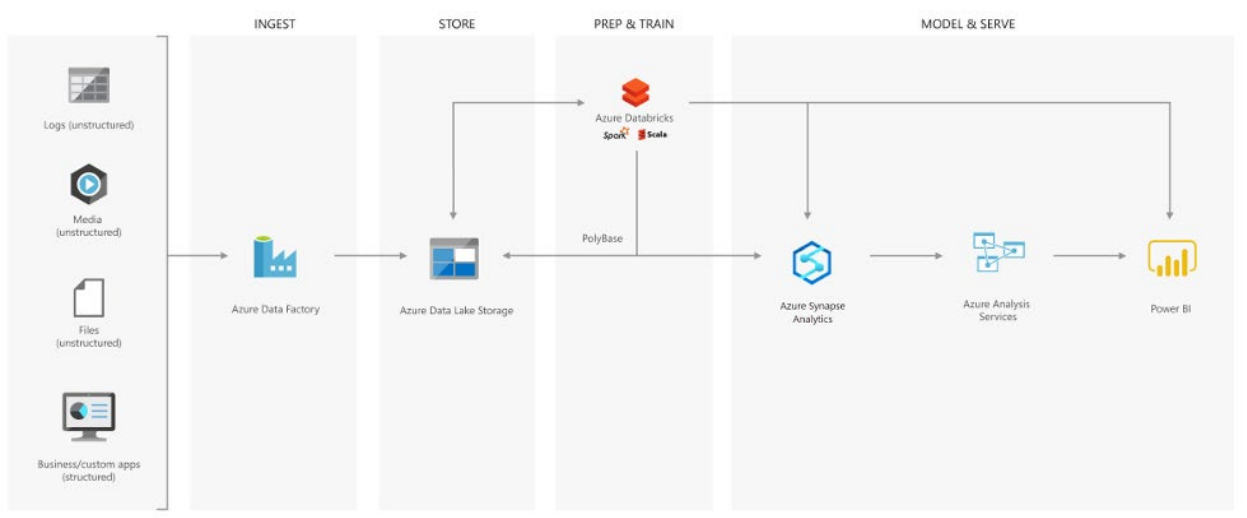
---

Question 30: Skipped

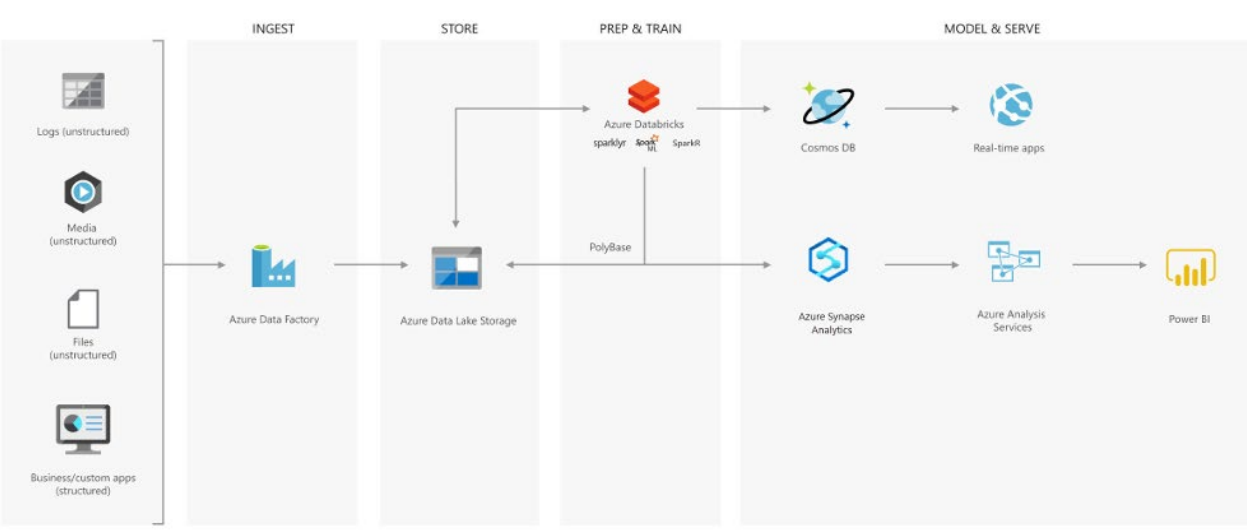
Scenario: You are working at LexCorp which is a household appliance manufacturer that wishes to implement predictive maintenance analytics for its appliances while onsite in its customers residences.

Review the following architecture designs.

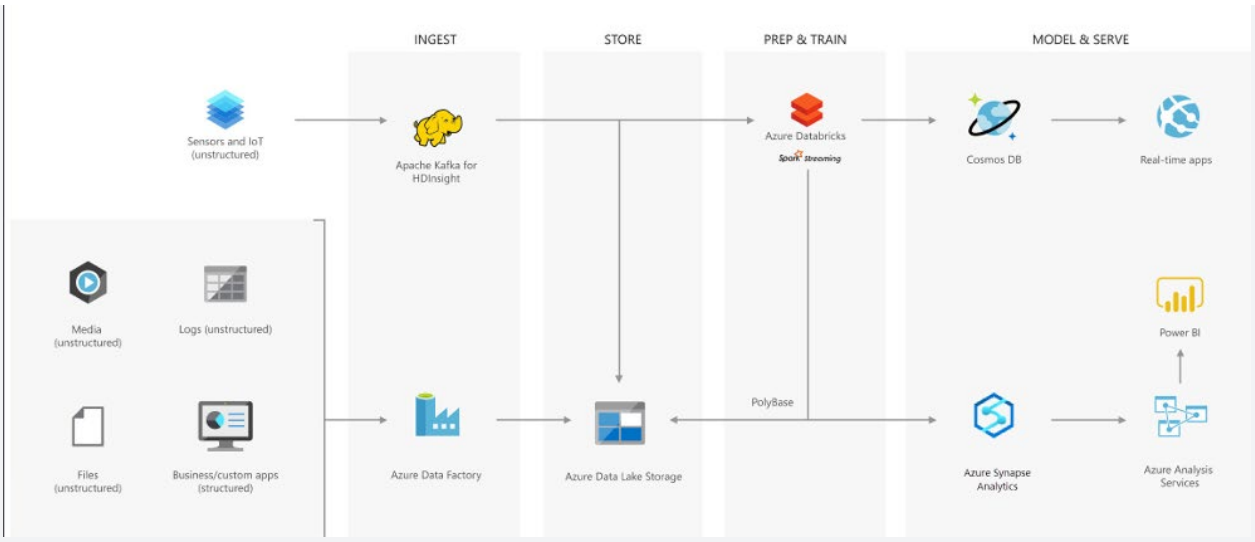
Design A:



Design B:



Design C:



Which architecture would be best suited for the need?

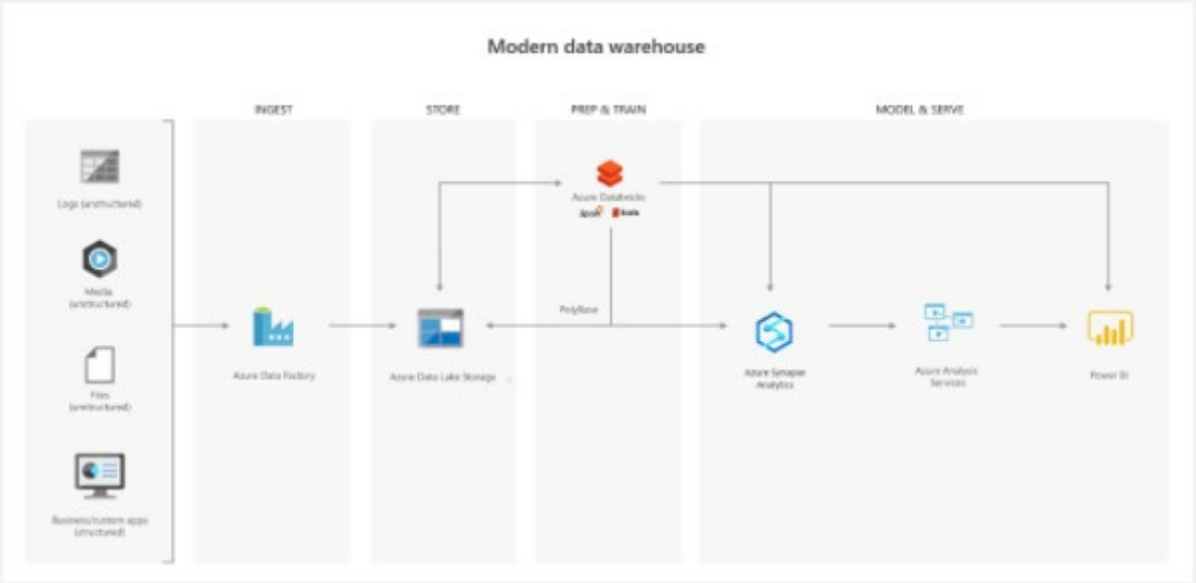
- Design C
  (Correct)
- None of the listed options
- Design A
- Design B

Explanation
Creating a modern data warehouse

Imagine you're a Data Engineering consultant for a Avengers Security. In the past, they've created an on-premises business intelligence solution that used a Microsoft SQL Server Database Engine, SQL Server Integration Services, SQL Server Analysis Services, and SQL Server Reporting Services to provide historical reports. They tried using the Analysis Services Data Mining component to create a predictive analytics solution to predict the buying behaviour of customers. While this approach worked well with low volumes of data, it couldn't scale after more than a gigabyte of data was collected. Furthermore, they were never able to deal with the JSON data that a third-party application generated when a customer used the feedback module of the point of sale (POS) application.

The company has turned to you for help with creating an architecture that can scale with the data needs that are required to create a predictive model and to handle the

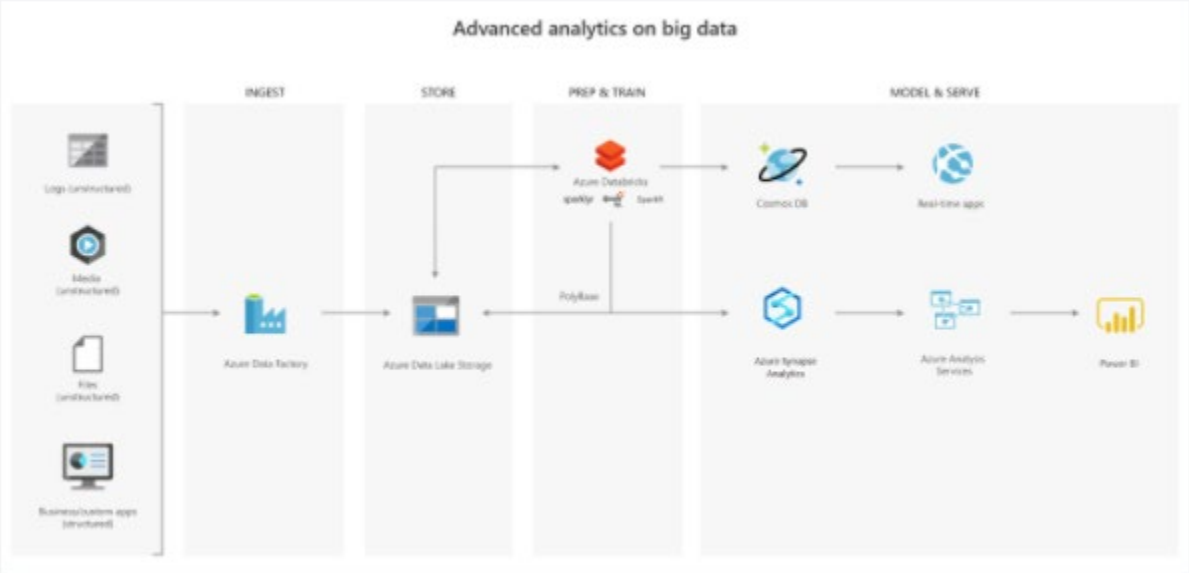
JSON data so that it's integrated into the BI solution. You suggest the following architecture:



The architecture uses Azure Data Lake Storage at the centre of the solution for a modern data warehouse. Integration Services is replaced by Azure Data Factory to ingest data into the Data Lake from a business application. This is the source for the predictive model that is built into Azure Databricks. PolyBase is used to transfer the historical data into a big data relational format that is held in Azure Synapse Analytics, which also stores the results of the trained model from Databricks. Azure Analysis Services provides the caching capability for SQL Data Warehouse to service many users and to present the data through Power BI reports.

Advanced analytics for big data

In this second use case, Azure Data Lake Storage plays an important role in providing a large-scale data store. Your skills are needed by Hydra Corporation, which is a global seller of bicycles and cycling components through a chain of resellers and on the internet. As their customers browse the product catalogue on their websites and add items to their baskets, a recommendation engine that is built into Azure Databricks recommends other products. They need to make sure that the results of their recommendation engine can scale globally. The recommendations are based on the web log files that are stored on the web servers and transferred to the Azure Databricks model hourly. The response time for the recommendation should be less than 1 ms. You propose the following architecture:

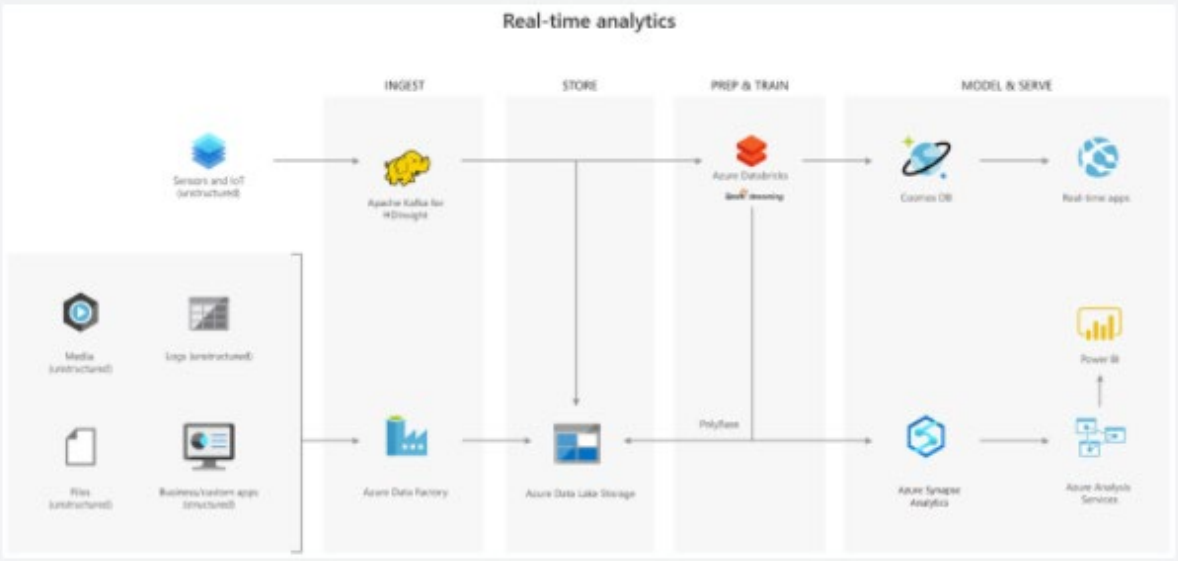


Real-time analytical solutions

To perform real-time analytical solutions, the ingestion phase of the architecture is changed for processing big data solutions. In this architecture, note the introduction of Apache Kafka for Azure HDInsight to ingest streaming data from an Internet of Things (IoT) device, although this could be replaced with Azure IoT Hub and Azure Stream Analytics. The key point is that the data is persisted in Data Lake Storage Gen2 to service other parts of the solution.

In this use case, you are a Data Engineer for HAMMER Industries, an organization that is working with a transport company to monitor the fleet of Heavy Goods Vehicles (HGV) that drive around Europe. Each HGV is equipped with sensor hardware that will continuously report metric data on the temperature, the speed, and the oil and brake solution levels of an HGV. When the engine is turned off, the sensor also outputs a file with summary information about a trip, including the mileage and elevation of a trip. A trip is a period in which the HGV engine is turned on and off.

Both the real-time data and batch data is processed in a machine learning model to predict a maintenance schedule for each of the HGVs. This data is made available to the downstream application that third-party garage companies can use if an HGV breaks down anywhere in Europe. In addition, historical reports about the HGV should be visually presented to users. As a result, the following architecture is proposed:



In this architecture, there are two ingestion streams. Azure Data Factory ingests the summary files that are generated when the HGV engine is turned off. Apache Kafka provides the real-time ingestion engine for the telemetry data. Both data streams are stored in Azure Data Lake Store for use in the future, but they are also passed on to other technologies to meet business needs. Both streaming and batch data are provided to the predictive model in Azure Databricks, and the results are published to Azure Cosmos DB to be used by the third-party garages. PolyBase transfers data from the Data Lake Store into SQL Data Warehouse where Azure Analysis Services creates the HGV reports by using Power BI.

https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction

Question 31: Skipped

Azure infrastructure is composed of geographies, regions, and Availability Zones, which limit the blast radius of a failure and therefore limit potential impact to customer applications and data. Duplicating customer content for redundancy and meeting service-level agreements (SLAs) in Azure meets which cloud technical requirement?

- High availability
  (Correct)
- All the listed options.
- Multilingual support
- None of the listed options.
- Maintainability
- Content distribution guarantees

Explanation

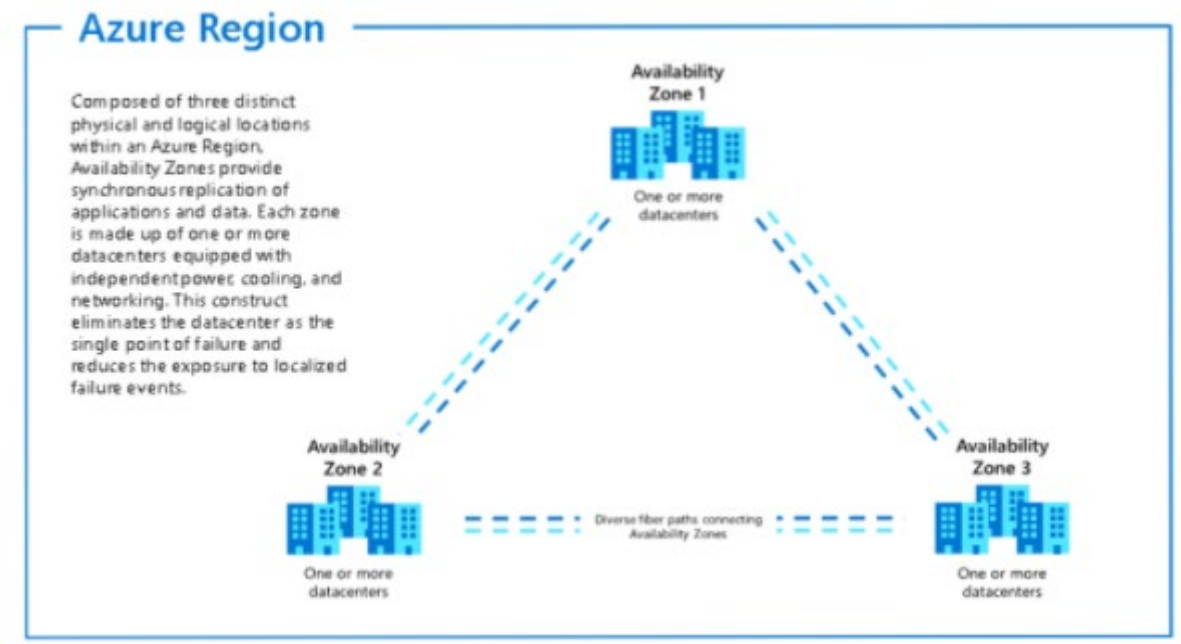
High availability duplicates customer content for redundancy and meets SLAs in Azure.

Microsoft Azure global infrastructure is designed and constructed at every layer to deliver the highest levels of redundancy and resiliency to its customers. Azure infrastructure is composed of geographies, regions, and Availability Zones, which limit the blast radius of a failure and therefore limit potential impact to customer applications and data. The Azure Availability Zones construct was developed to provide a software and networking solution to protect against datacentre failures and to provide increased high availability (HA) to our customers.

Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacentres with independent power, cooling, and networking. The physical separation of Availability Zones within a region limits the impact to applications and data from zone failures, such as large-scale flooding, major storms and superstorms, and other events that could disrupt site access, safe passage, extended utilities uptime, and the availability of resources. Availability Zones and their associated datacentres are designed such that if one zone is compromised, the services, capacity, and availability are supported by the other Availability Zones in the region.

Availability Zones can be used to spread a solution across multiple zones within a region, allowing for an application to continue functioning when one zone fails. With Availability Zones, Azure offers industry best 99.99% Virtual Machine (VM) uptime

service-level agreement (SLA). Zone-redundant services replicate your services and data across Availability Zones to protect from single points of failure.



https://docs.microsoft.com/en-us/azure/architecture/high-availability/building-solutions-for-high-availability

Question 32: Skipped

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

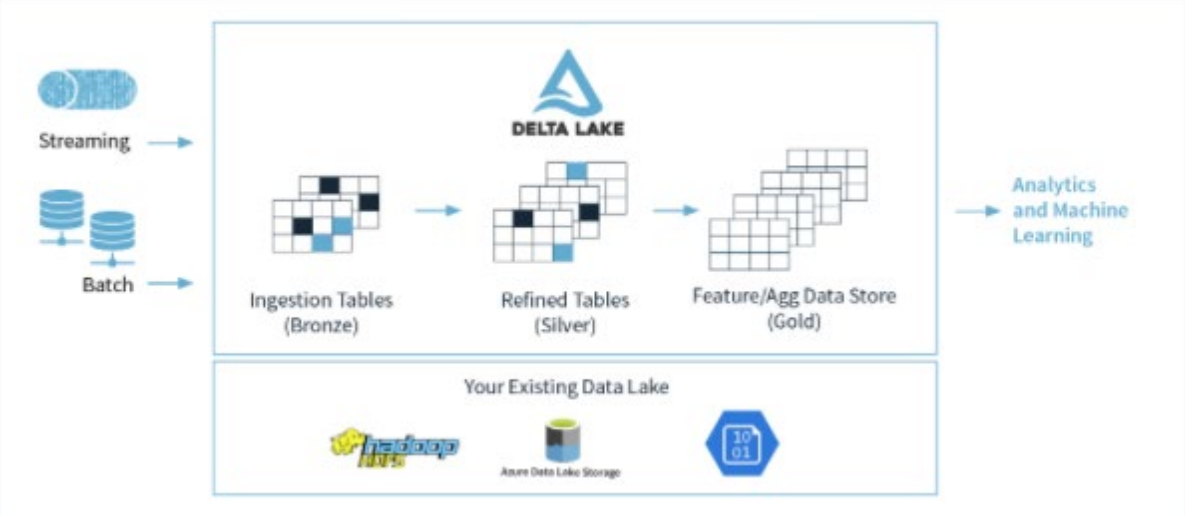
When working with large data sets, it can take a long time to run the sort of queries that clients need. These queries can't be performed in real time, and often require algorithms such as MapReduce that operate in parallel across the entire data set. The results are then stored separately from the raw data and used for querying.

One drawback to this approach is that it introduces latency. If processing takes a few hours, a query may return results that are several hours old. Ideally, you would like to get some results in real time (perhaps with some loss of accuracy), and combine these results with the results from the batch analytics.

The [?] is a big data processing architecture that addresses this problem by combining both batch- and real-time processing methods. It features an append-only immutable data source that serves as system of record. Timestamped events are appended to existing events (nothing is overwritten). Data is implicitly ordered by time of arrival.

- Anaconda architecture
- No-SQL architecture
- Lambda architecture
  (Correct)
- Serverless architecture

Explanation
An example of a Delta Lake Architecture might be as shown in the diagram below.



• Many **devices** generate data across different ingestion paths.

• Streaming data can be ingested from **IOT Hub** or **Event Hub**.

• Batch data can be ingested by **Azure Data Factory** or **Azure Databricks**.

• Extracted, Transformed data is loaded into a **Delta Lake**.

Lambda architecture

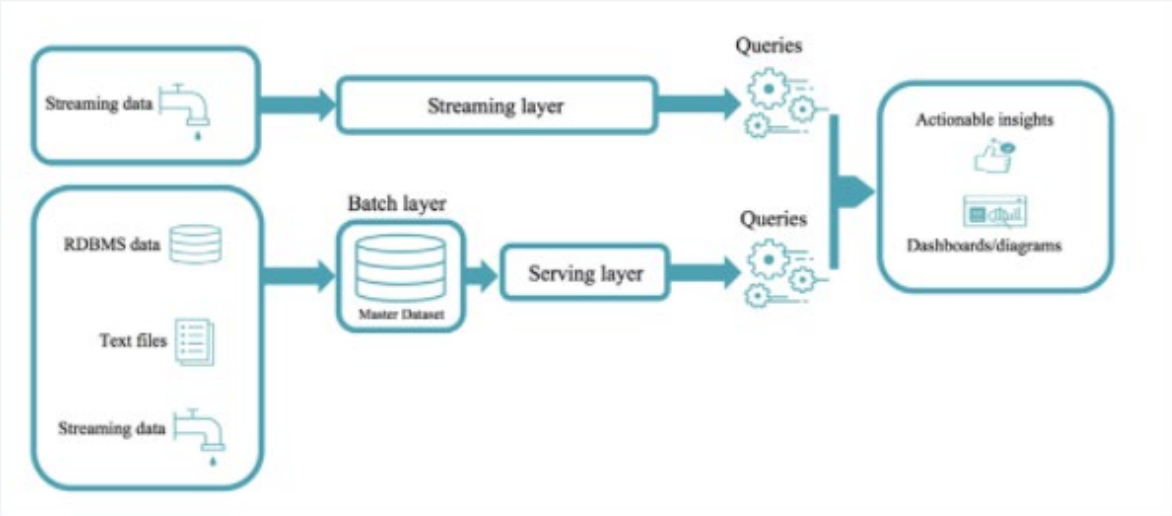
When working with large data sets, it can take a long time to run the sort of queries that clients need. These queries can't be performed in real time, and often require algorithms such as MapReduce that operate in parallel across the entire data set. The results are then stored separately from the raw data and used for querying.

One drawback to this approach is that it introduces latency. If processing takes a few hours, a query may return results that are several hours old. Ideally, you would like to get some results in real time (perhaps with some loss of accuracy), and combine these results with the results from the batch analytics.

The **lambda architecture** is a big data processing architecture that addresses this problem by combining both batch- and real-time processing methods. It features an append-only immutable data source that serves as system of record. Timestamped events are appended to existing events (nothing is overwritten). Data is implicitly ordered by time of arrival.

Notice how there are really two pipelines here, one batch and one streaming, hence the name *lambda* architecture.

It is difficult to combine processing of batch and real-time data as is evidenced by the diagram below:



Delta Lake architecture

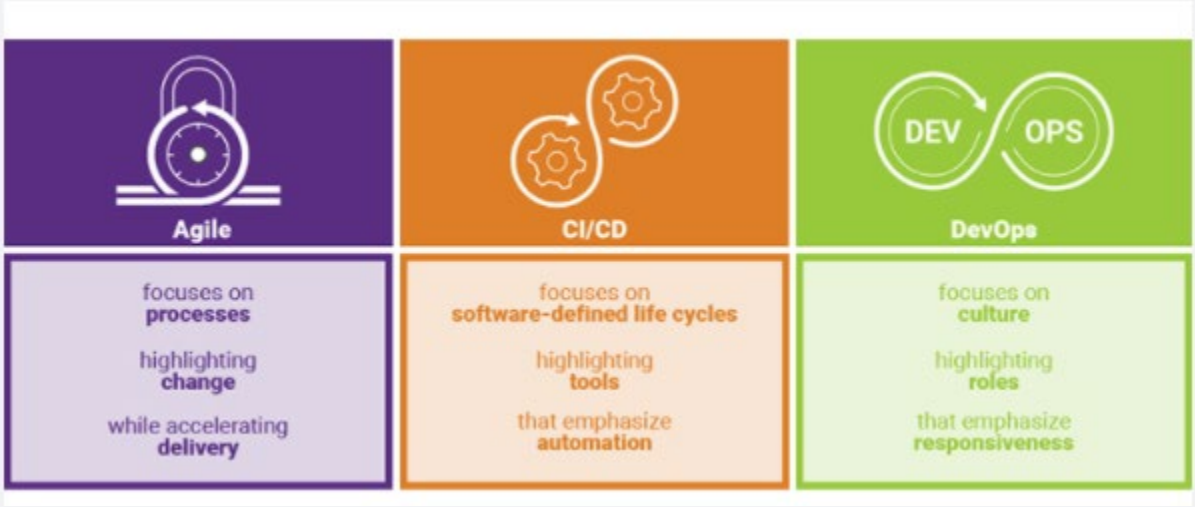
The Delta Lake Architecture is a vast improvement upon the traditional Lambda architecture. At each stage, we enrich our data through a unified pipeline that allows us to combine batch and streaming workflows through a shared filestore with ACID-compliant transactions.

Bronze tables contain raw data ingested from various sources (JSON files, RDBMS data, IoT data, etc.).

Silver tables will provide a more refined view of our data. We can join fields from various bronze tables to enrich streaming records, or update account statuses based on recent activity.

Gold tables provide business level aggregates often used for reporting and dashboarding. This would include aggregations such as daily active website users, weekly sales per store, or gross revenue per quarter by department.

The end outputs are actionable insights, dashboards, and reports of business metrics.



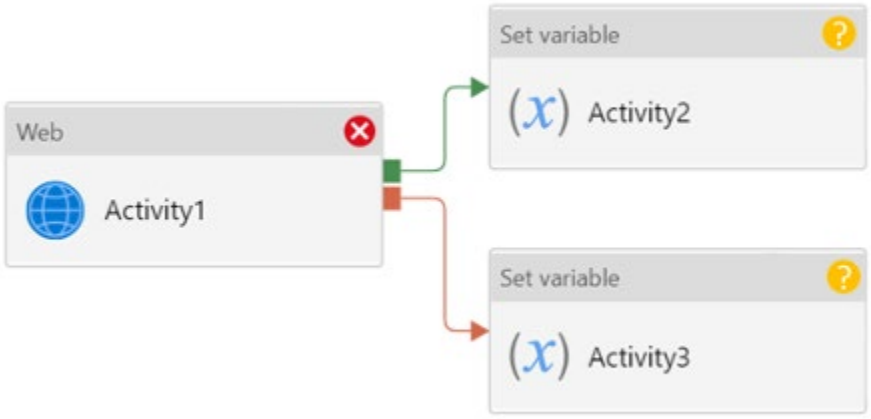
By considering our business logic at all steps of the extract-transform-load (ETL) pipeline, we can ensure that storage and compute costs are optimized by reducing unnecessary duplication of data and limiting ad hoc querying against full historic data.

Each stage can be configured as a batch or streaming job, and ACID transactions ensure that we succeed or fail completely.

https://www.jamesserra.com/archive/2019/10/databricks-delta-lake/

Question 33: Skipped

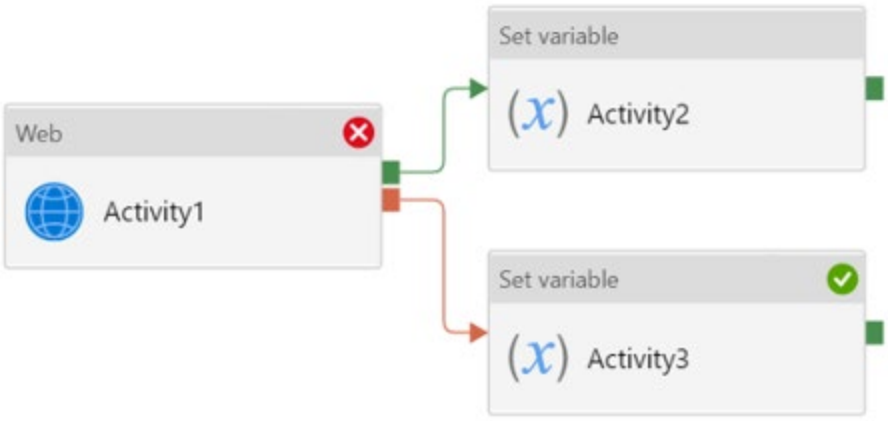
Scenario: We are working on a project which has a pipeline with 3 activities, where Activity1 has a success path to Activity2 and a failure path to Activity3.



What will the result be of pipeline?

- This pipeline reports completed.
- This pipeline reports skipped.
- This pipeline reports failure.
- This pipeline reports success.
  (Correct)

Explanation
If we have a pipeline with two activities where Activity 2 has a failure dependency on Activity1, the pipeline will not fail just because Activity1 failed. If Activity1 fails and Activity 2 succeeds, the pipeline will succeed. This scenario is treated as a try-catch block by Data Factory.



Azure Data Factory

In order to work with data factory pipelines, it is imperative to understand what a pipeline in Azure Data Factory is.

A pipeline in Azure Data Factory represents a logical grouping of activities where the activities together perform a certain task.

An example of a combination of activities in one pipeline can be, ingesting and cleaning log data in combination with a mapping data flow that analyzes the log data that has been cleaned.

A pipeline enables you to manage the separate individual activities as a set, which would otherwise be managed individually. It enables you to deploy and schedule the activities efficiently, through the use of a single pipeline, versus managing each activity independently.

Activities in a pipeline are referred to as actions that you perform on your data. An activity can take zero or more input datasets and produce one or more output datasets.

An example of an action can be the use of a copy activity, where you copy data from an Azure SQL Database to an Azure DataLake Storage Gen2. To build on this example, you can use a data flow activity or an Azure Databricks Notebook activity for processing and transforming the data that was copied to your Azure Data Lake Storage Gen2 account, in order to have the data ready for business intelligence reporting solutions like in Azure Synapse Analytics.

Since there are many activities that are possible in a pipeline in Azure Data Factory, we have grouped the activities in three categories:

• *Data movement activities*: the Copy Activity in Data Factory copies data from a source data store to a sink data store.

• *Data transformation activities*: Azure Data Factory supports transformation activities such as Data Flow, Azure Function, Spark, and others that can be added to pipelines either individually or chained with another activity.

• *Control activities*: Examples of control flow activities are 'get metadata', 'For Each', and 'Execute Pipeline'.

Activities can depend on each other. What we mean, is that the activity dependency defines how subsequent activities depend on previous activities. The dependency itself can be based on a condition of whether to continue in the execution of previous defined activities in order to complete a task. An activity that depends on one or more previous activities, can have different dependency conditions.

The four dependency conditions are:

• Succeeded

• Failed

• Skipped

• Completed

For example, if a pipeline has an Activity A, followed by an Activity B and Activity B has as a dependency condition on Activity A 'Succeeded', then Activity B will only run if Activity A has the status of succeeded.

If you have multiple activities in a pipeline and subsequent activities are not dependent on previous activities, the activities may run in parallel.

https://datasavvy.me/2021/02/18/azure-data-factory-activity-failures-and-pipeline-outcomes/

Question 34: Skipped

In order to create a Spark pool in Azure Synapse Analytics, what needs to be created to do so?

- Azure Databricks
- HDI
- Synapse Analytics Workspace
  (Correct)
- A Spark Instance

Explanation
In order to create a Spark pool in Azure Synapse Analytics, you would have to create a Synapse Analytics Workspace.

https://docs.microsoft.com/en-us/azure/synapse-analytics/quickstart-create-apache-spark-pool-portal

---

Question 35: Skipped

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

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.

Azure Synapse Analytics uses the [?] approach for bulk data.

- Extract, Transform, and Load (ETL)
- Atomicity, Consistency, Isolation, and Durability (ACID)
- Automated Data Processing Equipment (ADPE)
- Extract, Load, and Transform (ELT)
  (Correct)

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.

Ingesting and processing data

Azure Synapse Analytics uses the extract, load, and transform (ELT) approach for bulk data. SQL professionals are already familiar with bulk-copy tools such as bcp and the SQLBulkCopy API. Data engineers who work with Azure Synapse Analytics will soon learn how quickly PolyBase can load data.

PolyBase is a technology that removes complexity for data engineers. They take advantage of techniques for big-data ingestion and processing by offloading complex calculations to the cloud. Developers use PolyBase to apply stored procedures, labels, views, and SQL to their applications. You can also use Azure Data Factory to ingest and process data using PolyBase too.

Queries

As a data engineer, you can use the familiar Transact-SQL to query the contents of Azure Synapse Analytics. This method takes advantage of a wide range of features, including the WHERE, ORDER BY, and GROUP BY clauses. Load data fast by using PolyBase with additional Transact-SQL constructs such as `CREATE TABLE` and `SELECT`.

Data security

Azure Synapse Analytics supports both SQL Server authentication and Azure Active Directory. For high-security environments, set up multifactor authentication. From a data perspective, Azure Synapse Analytics supports security at the level of both columns and rows.

https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/data/enterprise-bi-synapse

---

Question 36: Skipped

Scenario: One of your teammates has just executed `GetBlockBlobReference` with the name of a blob.

What will happen?

- A new block blob is created in storage.
- The contents of the named blob are downloaded.
- A `CloudBlockBlob` object is created locally. No network calls are made. **(Correct)**
- An exception is thrown if the blob does not exist in storage.

Explanation
Getting a blob reference does not make any calls to Azure Storage, it simply creates an object locally that can work with a stored blob.

https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.storage.blob.cloudblobcontainer.getblockblobreference?view=azure-dotnet-legacy

---

Question 37: Skipped

Scenario: You are working as a consultant for Advanced Idea Mechanics (AIM) where the the IT team is working on an an Azure SQL database named AIM_Targets which contains a table named Targets_2021. This table has a field named Target_ID which is varchar(22).

Required: The team is to implement masking for the Target_ID field as per the following:

• Set the initial three prefix characters as "exposed".

• Set the final three suffix characters as "exposed".

• The remaining characters as "masked".

The team is planning to utilize data masking with a credit card function mask.

Will this solution meet the requirements?

- No
  (Correct)
- Yes

Explanation

Using data masking with a credit card function mask will not be successful. To meet the requirements, AIM must use Custom Text data masking, which exposes the first and last characters as specified and adds a custom padding string in the middle.

Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics support dynamic data masking. Dynamic data masking limits sensitive data exposure by masking it to non-privileged users.

Dynamic data masking helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal impact on the application layer. It's a policy-based security feature that hides the sensitive data in the result set of a query over designated database fields, while the data in the database is not changed.

For example, a service representative at a call centre might identify a caller by confirming several characters of their email address, but the complete email address shouldn't be revealed to the service representative. A masking rule can be defined that masks all the email address in the result set of any query. As another example, an

appropriate data mask can be defined to protect personal data, so that a developer can query production environments for troubleshooting purposes without violating compliance regulations.

Dynamic data masking basics

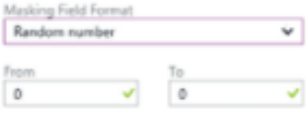
You set up a dynamic data masking policy in the Azure portal by selecting the **Dynamic Data Masking** blade under **Security** in your SQL Database configuration pane. This feature cannot be set using portal for SQL Managed Instance (use PowerShell or REST API). For more information, see Dynamic Data Masking.

Dynamic data masking policy

SQL users excluded from masking - A set of SQL users or Azure AD identities that get unmasked data in the SQL query results. Users with administrator privileges are always excluded from masking, and see the original data without any mask.

Masking rules - A set of rules that define the designated fields to be masked and the masking function that is used. The designated fields can be defined using a database schema name, table name, and column name.

Masking functions - A set of methods that control the exposure of data for different scenarios.

Masking function	Masking logic
Default	**Full masking according to the data types of the designated fields** • Use XXXX or fewer Xs if the size of the field is less than 4 characters for string data types (nchar, ntext, nvarchar). • Use a zero value for numeric data types (bigint, bit, decimal, int, money, numeric, smallint, smallmoney, tinyint, float, real). • Use 01-01-1900 for date/time data types (date, datetime2, datetime, datetimeoffset, smalldatetime, time). • For SQL variant, the default value of the current type is used. • For XML the document <masked/> is used. • Use an empty value for special data types (timestamp table, hierarchyid, GUID, binary, image, varbinary spatial types).
Credit card	**Masking method, which exposes the last four digits of the designated fields** and adds a constant string as a prefix in the form of a credit card. XXXX-XXXX-XXXX-1234
Email	**Masking method, which exposes the first letter and replaces the domain with XXX.com** using a constant string prefix in the form of an email address. aXX@XXXX.com
Random number	**Masking method, which generates a random number** according to the selected boundaries and actual data types. If the designated boundaries are equal, then the masking function is a constant number. 
Custom text	**Masking method, which exposes the first and last characters** and adds a custom padding string in the middle. If the original string is shorter than the exposed prefix and suffix, only the padding string is used. prefix[padding]suffix 

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dynamic-data-masking-get-started

Question 38: Skipped

Which `ALTER DATABASE` statement parameter allows a dedicated SQL pool to scale?

- `MODIFY`
  (Correct)
- `CHANGE`
- `SCALE`
- `OVER`

Explanation

`MODIFY` is used to scale a dedicated SQL pool.

In the following example, we use the ALTER DATABASE T-SQL statement to modify the service objective. Run the following query to change the service objective to DW300.

```
SQL

ALTER DATABASE mySampleDataWarehouse

MODIFY (SERVICE_OBJECTIVE = 'DW300c');
```

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/quickstart-scale-compute-tsql

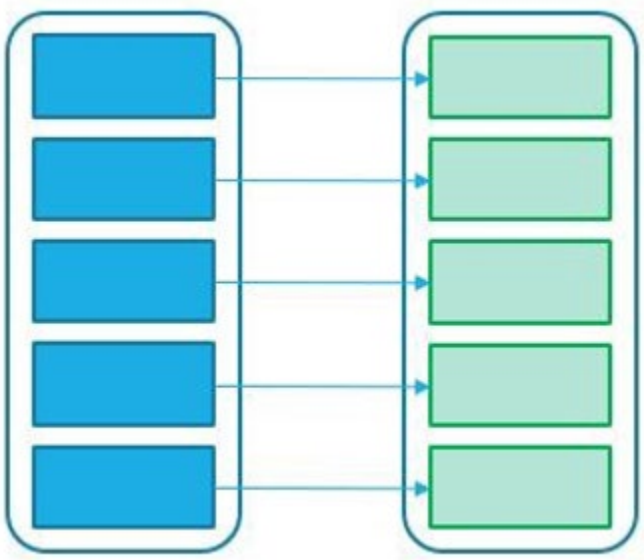
Question 39: Skipped

Which of the following statements describes a wide transformations?

- A wide transformation is where each input partition in the source data frame will contribute to sole output partition in the target data.
- A wide transformation applies data transformation over a large number of columns.
- A wide transformation requires sharing data across workers. It does so by shuffling data.
  (Correct)
- A wide transformation can be applied per partition/worker with no need to share or shuffle data to other workers.

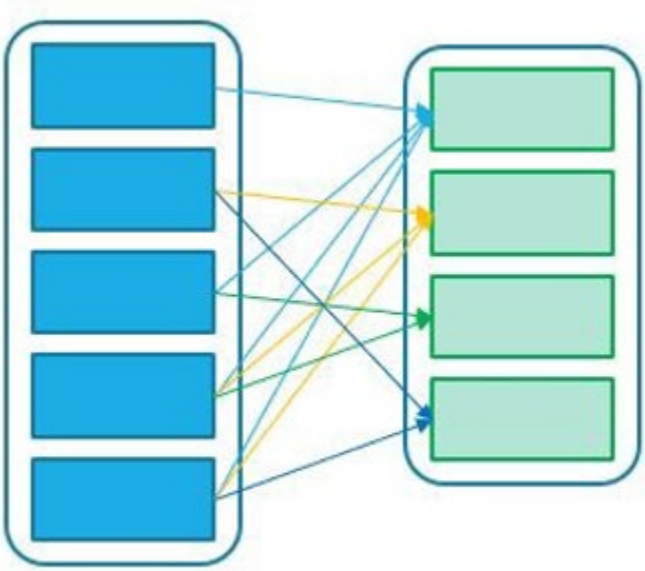
Explanation
Wide vs. Narrow Transformations

Transformations consisting of *narrow dependencie*s (narrow transformations) are those where each input partition in the source data frame will contribute to only one output partition in the target data.



Spark will automatically perform an operation called pipe-lining on narrow dependencies. If multiple filters are specified on a source data frame, they'll all be optimized for example performed in-memory.

W*ide dependencies* (or wide transformations) will have input partitions contributing to many output partitions. This will result in a *shuffle* where Spark will exchange partitions between executors.



Lazy Evaluation

The concept of *lazy evaluation* means that Spark will wait until required to execute the graph of computation instructions.

As opposed to narrow transformations, wide transformations cause data to shuffle between executors. This is because a wide transformation requires sharing data across workers. **Pipelining** helps us optimize our operations based on the differences between the two types of transformations.

Pipelining

• Pipelining is the idea of executing as many operations as possible on a single partition of data.

• Once a single partition of data is read into RAM, Spark will combine as many narrow operations as it can into a single **Task**

• Wide operations force a shuffle, conclude a stage, and end a pipeline.

Shuffles

A shuffle operation is triggered when data needs to move between executors.

To carry out the shuffle operation Spark needs to:

• Convert the data to the `UnsafeRow`, commonly referred to as **Tungsten Binary Format**.

• Write that data to disk on the local node - at this point the slot is free for the next task.

• Send that data across the wire to another executor

  • Technically the Driver decides which executor gets which piece of data.

  • Then the executor pulls the data it needs from the other executor's shuffle files.

• Copy the data back into RAM on the new executor

  • The concept, if not the action, is just like the initial read "every“ `DataFrame` starts with.

  • The main difference being it's the 2nd+ stage.

As we will see in a moment, this amounts to a free cache from what is effectively temp files.

Some actions induce in a shuffle. Good examples would include the operations `count()` and `reduce(..)`.

UnsafeRow (also known as Tungsten Binary Format)

Sharing data from one worker to another can be a costly operation.

Spark has optimized this operation by using a format called **Tungsten**.

Tungsten prevents the need for expensive serialization and de-serialization of objects in order to get data from one JVM to another.

The data that is "shuffled" is in a format known as `UnsafeRow`, or more commonly, the Tungsten Binary Format.

`UnsafeRow` is the in-memory storage format for Spark SQL, DataFrames & Datasets.

Advantages include:

• Compactness:

  • Column values are encoded using custom encoders, not as JVM objects (as with RDDs).

  • The benefit of using Spark 2.x's custom encoders is that you get almost the same compactness as Java serialization, but significantly faster encoding/decoding speeds.

  • Also, for custom data types, it is possible to write custom encoders from scratch.

• Efficiency: Spark can operate *directly out of Tungsten*, without first deserializing Tungsten data into JVM objects.

https://medium.com/@lackshub/notes-for-databricks-crt020-exam-prep-9fbc97a2147e

---

Question 40: Skipped

True or False: When using Azure Synapse Analytics, the preferred method for loading data is to use the service administrator account as this ensures the required permissions are available to write the content to the appropriate destinations. Lesser accounts run the risk of failure due to permission restrictions.

- True
- False **(Correct)**

Explanation
A mistake that many people make when first exploring dedicated SQL Pools are to use the service administrator account as the one used for loading data. This account is limited to using the smallrc dynamic resource class that can use between 3% and 25% of the resources depending on the performance level of the provisioned SQL Pools.

Instead, it's better to create specific accounts assigned to different resource classes dependent on the anticipated task. This will optimize load performance and maintain concurrency as required by managing the available resource slots available within the dedicated SQL Pool.

https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/move-resource-group-and-subscription

Question 41: Skipped

Scenario: You are working as a consultant at **Advanced Idea Mechanics** (**A.I.M.**) who is a privately funded think tank organized of a group of brilliant scientists whose sole dedication is to acquire and develop power through technological means. Their goal is to use this power to overthrow the governments of the world. They supply arms and technology to radicals and subversive organizations in order to foster a violent technological revolution of society while making a profit.

The company has 10,000 employees. Most employees are located in Europe. The company supports teams worldwide.

AIM has two main locations: a main office in London, England, and a manufacturing plant in Berlin, Germany.

During events, 100 engineers set up a remote portable office by using a VPN to connect the datacentre in the London office. The portable office is set up and torn down in approximately 20 different countries each year.

Chaos Central

During major events, AIM uses a primary application named Chaos Central. Each vehicle used in the activity has several sensors that send real-time telemetry data to the London datacentre. The data is used for real-time tracking of the vehicles. Chaos Central also sends batch updates to an application named Mechanical Workflow by using Microsoft SQL Server Integration Services (SSIS).

The telemetry data is sent to a MongoDB database. A custom application then moves the data to databases in SQL Server 2017. The telemetry data in MongoDB has more than 500 attributes. The application changes the attribute names when the data is moved to SQL Server 2017.

The database structure contains both OLAP and OLTP databases.

Mechanical Workflow

Mechanical Workflow is used to track changes and improvements made to the vehicles during their lifetime. Currently, Mechanical Workflow runs on SQL Server 2017 as an OLAP system. Mechanical Workflow has a named Table1 that is 1 TB. Large aggregations are performed on a single column of Table 1.

Requirements:

• Data collection for Chaos Central must be moved to Azure Cosmos DB and Azure SQL Database. The data must be written to the Azure datacentre closest to each race and must converge in the least amount of time.

• The query performance of Chaos Central must be stable, and the administrative time it takes to perform optimizations must be minimized.

• The datacentre for Mechanical Workflow must be moved to Azure SQL data Warehouse.

• Transparent data encryption (TDE) must be enabled on all data stores, whenever possible.

• An Azure Data Factory pipeline must be used to move data from Cosmos DB to SQL Database for Chaos Central. If the data load takes longer than 20 minutes, configuration changes must be made to Data Factory.

• The telemetry data must migrate toward a solution that is native to Azure.

• The telemetry data must be monitored for performance issues. You must adjust the Cosmos DB Request Units per second (RU/s) to maintain a performance SLA while minimizing the cost of the Ru/s.

Which of the following data stores should be configured to meet the TDE requirement?

- Cosmos DB
- SQL Database
- SQL Data Warehouse
  (Correct)
- All of the listed items

Explanation
Transparent data encryption (TDE) must be enabled on all data stores, whenever possible. The datacentre for Mechanical Workflow must be moved to Azure SQL data Warehouse. Cosmos DB does not support TDE.

Transparent data encryption (TDE) helps protect Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics against the threat of malicious offline activity by encrypting data at rest. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring

changes to the application. By default, TDE is enabled for all newly deployed SQL Databases and must be manually enabled for older databases of Azure SQL Database, Azure SQL Managed Instance. TDE must be manually enabled for Azure Synapse Analytics.

TDE performs real-time I/O encryption and decryption of the data at the page level. Each page is decrypted when it's read into memory and then encrypted before being written to disk. TDE encrypts the storage of an entire database by using a symmetric key called the Database Encryption Key (DEK). On database startup, the encrypted DEK is decrypted and then used for decryption and re-encryption of the database files in the SQL Server database engine process. DEK is protected by the TDE protector. TDE protector is either a service-managed certificate (service-managed transparent data encryption) or an asymmetric key stored in Azure Key Vault (customer-managed transparent data encryption).

For Azure SQL Database and Azure Synapse, the TDE protector is set at the server level and is inherited by all databases associated with that server. For Azure SQL Managed Instance, the TDE protector is set at the instance level and it is inherited by all encrypted databases on that instance. The term *server* refers both to server and instance throughout this document, unless stated differently.

https://docs.microsoft.com/en-us/azure/azure-sql/database/transparent-data-encryption-tde-overview?tabs=azure-portal

---

Question 42: 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 the following:

"A legacy account type which supports only block and append blobs."

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

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 43: Skipped

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

A common use with [?] is to take the data that is shared and use it as a source into Azure Data Factory pipelines to use with your own internal data.

- Azure Data Share **(Correct)**
- Azure SQL Database
- Azure Managed SQL Warehouse
- Azure Data Lake Storage
- Azure Databricks

Explanation

A common use with Azure Data Share is to take the data that is shared and use it as a source into Azure Data Factory pipelines to use with your own internal data.

Azure Data Factory will give you the opportunity to perform code-free ETL/ELT, which will result in a comprehensive overview of your data pipelines. As a data engineer, this gives you the confidence to work with more data.

In order to start creating a pipeline, we first need to set up linked services in Azure Data Factory. Linked services define the connection information for data factory to the external resources you want to connect with, for example an Azure SQL Database or Azure Data Lake Storage.

The connection to the data source and dataset that is linked to that linked service, represents the data structure. For example, an Azure Data Lake Storage linked service will specify the connection string to the Azure Data Lake Storage account.

The connection string can be passed through to Azure Data Factory by creating a linked service.

The purpose of linked services, is to represent and show data store as well as compute resources that need to be hosted for the execution of a pipeline or activity.

Using the code-free User Experience of Azure Data Factory from the Azure portal makes it easy for the non-coder to develop linked services.

Currently, Azure Data Factory supports over 85 of these connectors.

A pipeline in Azure Data Factory is a logical grouping of activities such as copy in order to perform a task. The activity defines the operation that you're performing on the data (therefore, a copy means copying the same data to another data store).

The dataset that you're using is pointing to the data that you're going to use from the linked service.

Therefore, if you have linked a SQL DB, which contains a database, which contains tables, you can select the table that you want to copy.

In doing so, the data from that table will be copied to an Azure Data Lake storage Account.

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

Question 44: Skipped

Azure HDInsight provides technologies to help you ingest, process, and analyze big data. It supports batch processing, data warehousing, IoT, and data science.

Data processing within Hadoop uses which of the following to process big data? (Select three)

- R
- Java
  (Correct)
- .NET
  (Correct)
- Python
  (Correct)
- C++
- C#
- JavaScript

Explanation

Azure HDInsight provides technologies to help you ingest, process, and analyze big data. It supports batch processing, data warehousing, IoT, and data science.

Data processing

In Hadoop, use Java, Python and .NET to process big data. Mapper consumes and analyzes input data. It then emits tuples that Reducer can analyze. Reducer runs summary operations to create a smaller combined result set.

Spark processes streams by using Spark Streaming. For machine learning, use the 200 preloaded Anaconda libraries with Python. Use GraphX for graph computations.

Developers can remotely submit and monitor jobs from Spark. Storm supports common programming languages like Java, C#, and Python.

Queries

Hadoop supports Pig and HiveQL languages. In Spark, data engineers use Spark SQL.

Data security

Hadoop supports encryption, Secure Shell (SSH), shared access signatures, and Azure Active Directory security.

https://azure.microsoft.com/en-us/blog/manage-azure-hdinsight-clusters-using-net-python-or-java/

Question 45: Skipped

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

Microsoft Azure Storage is a managed service that provides durable, secure, and scalable storage in the cloud. Azure Blob storage is an object storage solution optimized for storing massive amounts of unstructured data, such as text or binary data.

Azure Storage supports which kinds of blobs? (Select three)

- ☐ Block **(Correct)**
- ☐ GPv2
- ☐ GPv1
- ☐ Page **(Correct)**
- ☐ Queue
- ☐ Append **(Correct)**

Explanation
Microsoft Azure Storage is a managed service that provides durable, secure, and scalable storage in the cloud.

Durable	Redundancy ensures that your data is safe in the event of transient hardware failures. You can also replicate data across datacenters or geographical regions for extra protection from local catastrophe or natural disaster. Data replicated in this way remains highly available in the event of an unexpected outage.
Secure	All data written to Azure Storage is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.
Scalable	Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
Managed	Microsoft Azure handles maintenance and any critical problems for you.

A single Azure subscription can host up to 200 storage accounts, each of which can hold 500 TB of data.

Azure data services

Azure storage includes four types of data:

• Azure Blobs: A massively scalable object store for text and binary data. Can include support for Azure Data Lake Storage Gen2.

• **Files**: Managed file shares for cloud or on-premises deployments.

• Azure Queues: A messaging store for reliable messaging between application components.

• Azure Tables: A NoSQL store for schema-less storage of structured data. Table Storage is not covered in this module.

• Azure Disks: Block-level storage volumes for Azure VMs.

All of these data types in Azure Storage are accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides SDKs for Azure Storage in various languages, and a REST API. You can also visually explore your data right in the Azure portal.

Blob storage

Azure Blob storage is an object storage solution optimized for storing massive amounts of unstructured data, such as text or binary data. Blob storage is ideal for:

• Serving images or documents directly to a browser, including full static websites.

• Storing files for distributed access.

• Streaming video and audio.

• Storing data for backup and restoration, disaster recovery, and archiving.

• Storing data for analysis by an on-premises or Azure-hosted service.

Azure Storage supports three kinds of blobs:

Blob type	Description
Block blobs	Block blobs are used to hold text or binary files up to ~5 TB (50,000 blocks of 100 MB) in size. The primary use case for block blobs is the storage of files that are read from beginning to end, such as media files or image files for websites. They are named block blobs because files larger than 100 MB must be uploaded as small blocks. These blocks are then consolidated (or committed) into the final blob.
Page blobs	Page blobs are used to hold random-access files up to 8 TB in size. Page blobs are used primarily as the backing storage for the VHDs used to provide durable disks for Azure Virtual Machines (Azure VMs). They are named page blobs because they provide random read/write access to 512-byte pages.
Append blobs	Append blobs are made up of blocks like block blobs, but they are optimized for append operations. These blobs are frequently used for logging information from one or more sources into the same blob. For example, you might write all of your trace logging to the same append blob for an application running on multiple VMs. A single append blob can be up to 195 GB.

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

Question 46: 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. Notebook cells are individual blocks of code or text that can be ran independently or as a group.

True or False: It is possible to reference data or variables directly across different languages in a Synapse Studio notebook.

- False
(Correct)
- True

Explanation
In order to understand the development of notebooks you need to understand that it consists of cells. Cells are individual blocks of code or text that can be ran independently or as a group.

In order to develop notebooks, there are a couple of things to take in mind

• Adding cells to notebooks This will give you the opportunity to add code in a different cell. There are multiple ways to add a new cell to your notebook.

• Setting a primary language Azure Synapse Studio notebook has support for four Apache Spark languages to be set as primary languages in a notebook.

These are pySpark (Python), Spark (Scala), SparkSQL, and .NET for Apache Spark (C#)

• Using multiple languages Within a notebook you are enabled to use multiple languages. The one thing to take in mind is that you need to use magic commands, at the beginning of a cell.

• Using temp tables to reference data across languages. **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.

• IDE-Style IntelliSense When you use Azure Synapse Studio notebooks, you'll see the integration with the Monaco editor. It enables you to bring IDE-style IntelliSense to the cell editor. This helps you in cases of Syntax highlight, error marker, and automatic code completions to help you to write code and identify issues quicker. You do have to take in mind that The IntelliSense features are sometimes at different levels of maturity for different languages. So depending on the language you want to write your code in, in the

Azure Synapse Studio Notebooks environment you could check the following table that explains what is supported using the types of language at your convenience.

Languages	Syntax Highlight	Syntax Error Marker	Syntax Code Completion	Variable Code Completion	System Function Code Completion	User Function Code Completion	Smart Indent	Code Folding
PySpark (Python)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Spark (Scala)	Yes	Yes	Yes	Yes	-	-	-	Yes
SparkSQL	Yes	Yes	-	-	-	-	-	-
.NET for Spark (C#)	Yes	-	-	-	-	-	-	-

• Undo Cell operations. When you, for example, need to revoke a cell operation, you can do so within the Azure Synapse Studio notebook environment.

• Move a cell. If you want to align different cells of code and put them in a correct order, the notebook environment gives you the opportunity to do so.

• Delete a cell. If you have written a cell of code, but no longer need it or it needs to be deleted, the functionality can be used within the notebook environment.

• Collapse Cell in and output If you want to collapse a cell to check in or output, the functionality is available to you when using the notebook environment in Synapse Studio.

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

---

Question 47: Skipped

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

In Azure Stream Analytics, a(n) [?] is a unit of execution.

- Output
- Transformation query
- Input
- Job **(Correct)**

Explanation
In Azure Stream Analytics, a *job* is a unit of execution. A Stream Analytics job pipeline consists of three parts:

• An **input** that provides the source of the data stream.

• A **transformation query** that acts on the input. For example, a transformation query could aggregate the data.

• An **output** that identifies the destination of the transformed data.

The Stream Analytics pipeline provides a transformed data flow from input to output, as the following diagram shows.

https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-monitoring

Question 48: 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 Storage account *kind,* which is a set of policies that determine which data services you can include in the account and the pricing of those services.

Which of the following are valid *kinds* of Storage accounts? (Select three)

- ☐ Append blobs Storage
- ☐ Data Pool Storage
- ☐ Block blobs Storage
- ☐ General Purpose v1
  (Correct)
- ☐ General Purpose v2
  (Correct)
- ☐ Data Lake Storage
- ☐ Container Storage
- ☐ Page blobs Storage
- ☐ Classic Storage
- ☐ Blob Storage
  (Correct)

Explanation
Azure Storage Account kind

Storage account *kind* is a set of policies that determine which data services you can include in the account and the pricing of those services. There are three kinds of storage accounts:

• **StorageV2 (general purpose v2)**: the current offering that supports all storage types and all of the latest features

• **Storage (general purpose v1)**: a legacy kind that supports all storage types but may not support all features

• **Blob storage**: a legacy kind that allows only block blobs and append blobs

Microsoft recommends that you use the **General-purpose v2** option for new storage accounts.

There are a few special cases that can be exceptions to this rule. For example, pricing for transactions is lower in general purpose v1, which would allow you to slightly reduce costs if that matches your typical workload.

The core advice here is to choose the **Resource Manager** deployment model and the **StorageV2 (general purpose v2)** account kind for all your storage accounts. The other options still exist primarily to allow existing resources to continue operation. For new resources, there are few reasons to consider the other choices.

https://www.ais.com/how-to-choose-the-right-kind-of-azure-storage-account/

Question 49: Skipped

Which is the default distribution used for a table in Synapse Analytics?

- Replicated Table distribution
- HASH distribution
- B-tree distribution
- Non-clustered distribution
- Round-Robin distribution **(Correct)**
- Clustered distribution

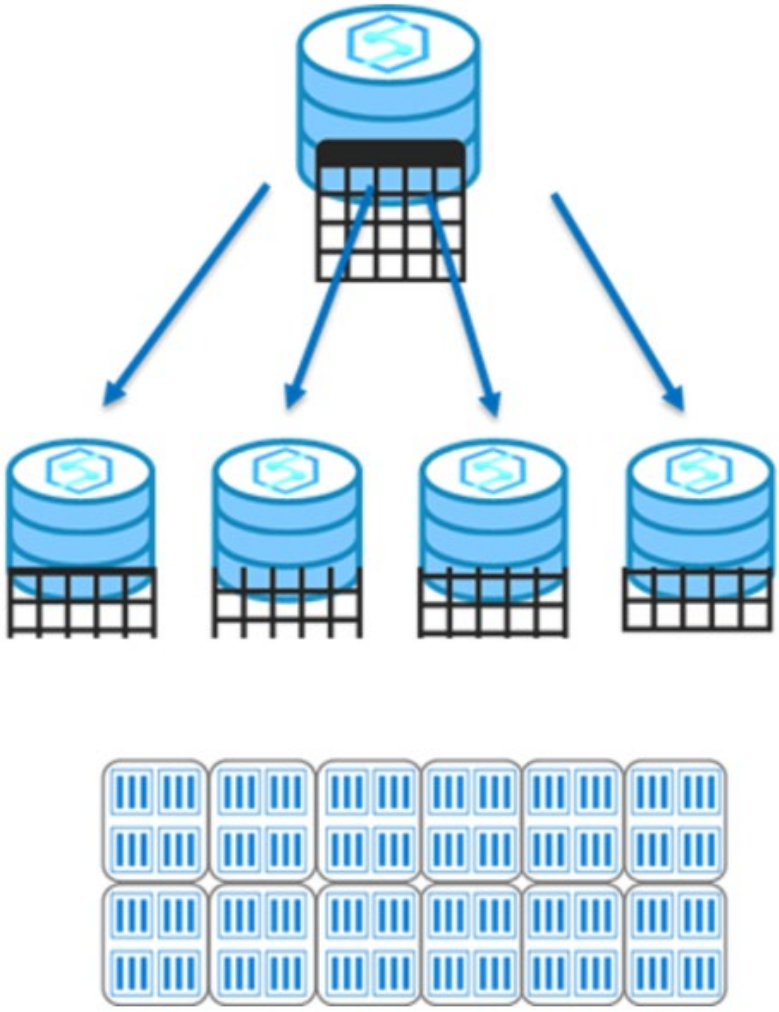
Explanation

Round-Robin is the default distribution created for a table and delivers fast performance when used for loading data but may negatively impact larger queries.

There are three main table distributions available in Synapse Analytics SQL Pools.

Selecting the correct table distribution can have an impact on the data load and query performance as follows:

Round robin distribution



This is the default distribution created for a table and delivers fast performance when used for loading data.

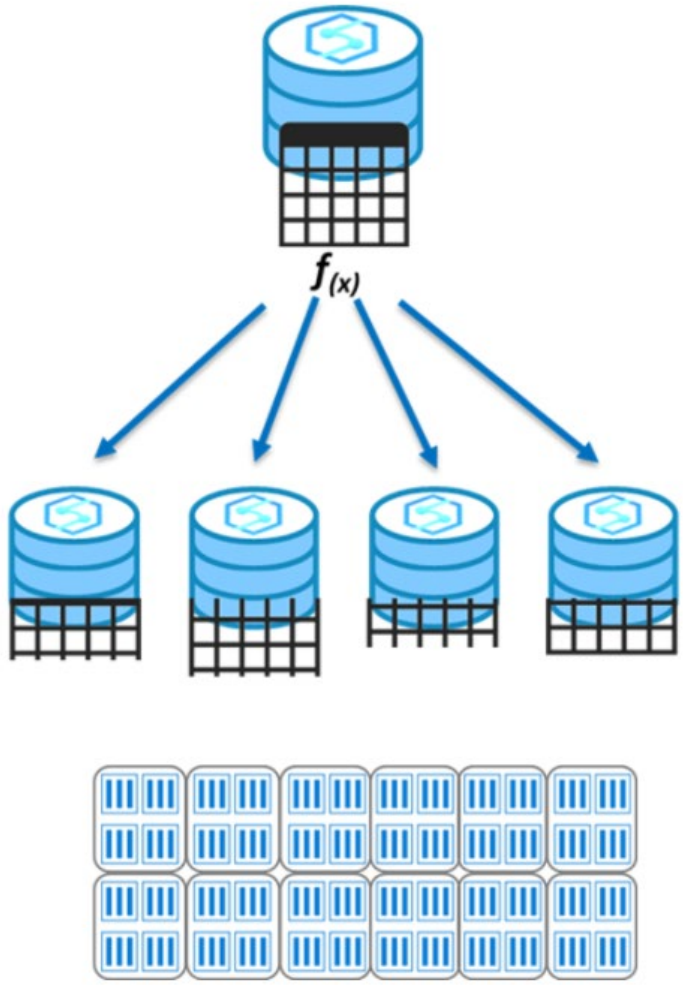
A round-robin distributed table distributes data evenly across the table but without any further optimization. A distribution is first chosen at random and then buffers of rows are assigned to distributions sequentially.

It is quick to load data into a round-robin table, but query performance can often be better with hash distributed tables for larger datasets.

Joins on round-robin tables may negatively affect query workloads, as data that is gathered for processing then has to be reshuffled to other compute nodes, which take additional time and processing.

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

Hash distribution



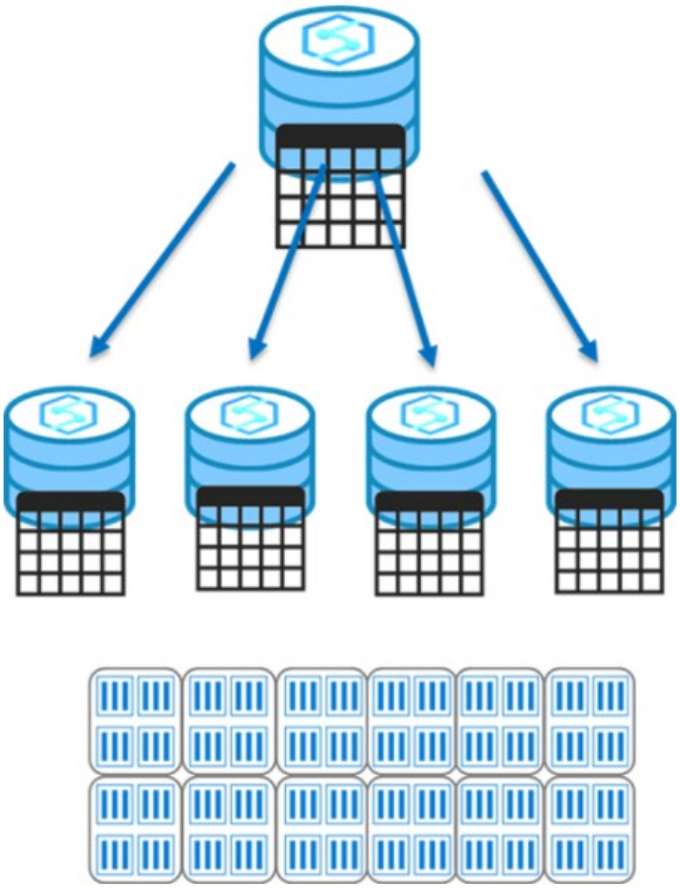
This distribution can deliver the highest query performance for joins and aggregations on large tables.

To shard data, a hash function is used to deterministically assign each row to a distribution. In the table definition, one of the columns is designated as the distribution column.

There are performance considerations for the selection of a distribution column, such as distinctness, data skew, and the types of queries that run on the system.

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

Replicated tables



A replicated table provides the fastest query performance for small tables.

A table that is replicated caches a full copy of the table on each compute node. Consequently, replicating a table removes the need to transfer data among compute nodes before a join or aggregation. As such extra storage is required and there is additional overhead that is incurred when writing data, which make large tables impractical.

Frequent data modifications will cause the cached copy to be invalidated, and require the table be recached.

Scaling the SQL Pool will also require the table be recached.

https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/design-guidance-for-replicated-tables