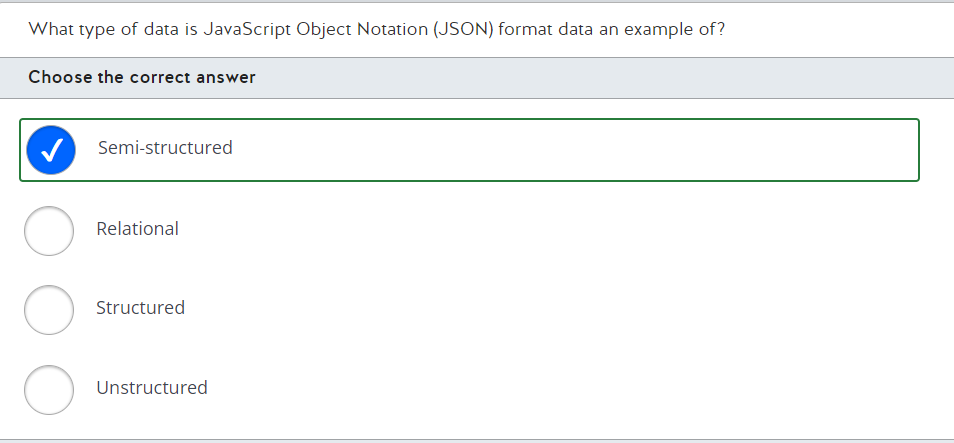
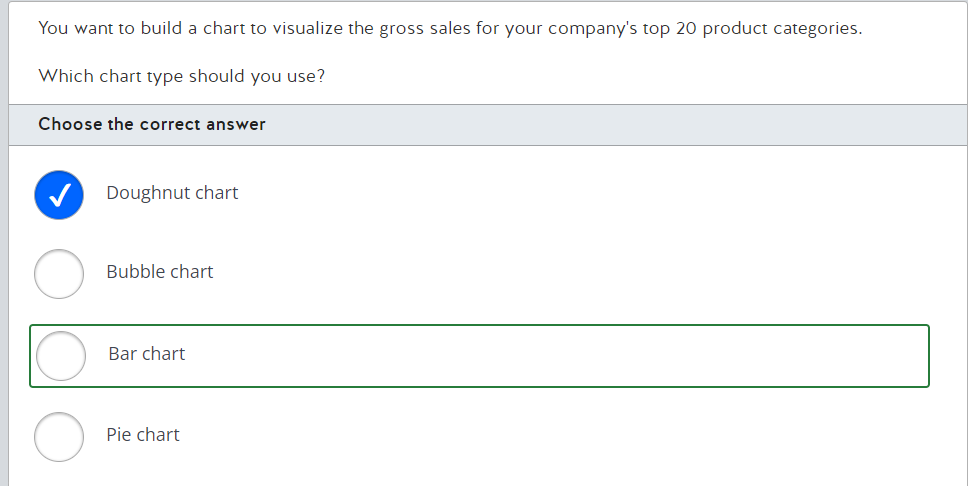
1.-



A JSON format data file is an example of semi-structured data. In a JSON document, each data field is identified by a label, followed by a colon, and the field value. A field can have multiple values, as in:

A JSON format data file is an example of semi-structured data. In a JSON document, each data field is identified by a label, followed by a colon, and the field value. A field can have multiple values, as in:  
  
{  
  "ID": "1",  
  "Name": "John Doe",  
  "Telephone": [  
    { "Home": "1-999-9999999" },  
    { "Business": "1-888-8888888" },  
    { "Cell": "1-555-55555555" }  
  ],  
  "Address": [  
    { "Home": [  
      { "StreetAddress": "121 Some Street" },  
      { "City": "Some City" },  
      { "State": "MO" },  
      { "Zip": "63601" }  
    ] },  
    { "Business": [  
      { "StreetAddress": "87 Some Building" },  
      { "City": "Another City" },  
      { "State": "MO" },  
      { "Zip": "63121" }  
    ] }  
  ]  
}  
  
Curly brackets are used to enclose the document and any subdocuments. Azure Cosmos DB is the most common storage solution for this type of data.  
  
Unstructured data includes files like video or audio files with no schema structure. This type of data is usually stored in Azure Blob storage.  
  
Structured data and relational data refer to the same data structure with data in a highly normalized format and stored in multiple related tables. The most common storage solution is some type of SQL database management system.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)  
  
[Non-relational data and NoSQL](https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data)

2.-



You should use a bar chart. Bar charts are the most common charts used to visualize and compare data across categories. You can determine the top selling product categories by sorting the gross sales field.  
  
You should not use a bubble chart. Bubble charts are used to visualize three dimensions of data, with the x- and y-axis representing two dimensions of data, and the third dimension represented by the bubble size.  
  
You should not use a pie or a doughnut chart. These charts are best suited when comparing a few categories only. When you have more than eight categories, reading and interpreting becomes quite difficult. Doughnut charts are similar to pie charts, with the difference that the center is empty for doughnut charts.  
  
**References**  
  
[Explore data visualization](https://docs.microsoft.com/en-us/learn/modules/explore-concepts-of-data-analytics/3-explore-data-visualization)  
  
[Visualization types in Power BI](https://docs.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-types-for-reports-and-q-and-a)  
  
[Tips and tricks for creating reports in Power BI Desktop](https://docs.microsoft.com/en-us/power-bi/create-reports/desktop-tips-and-tricks-for-creating-reports)

3.-

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

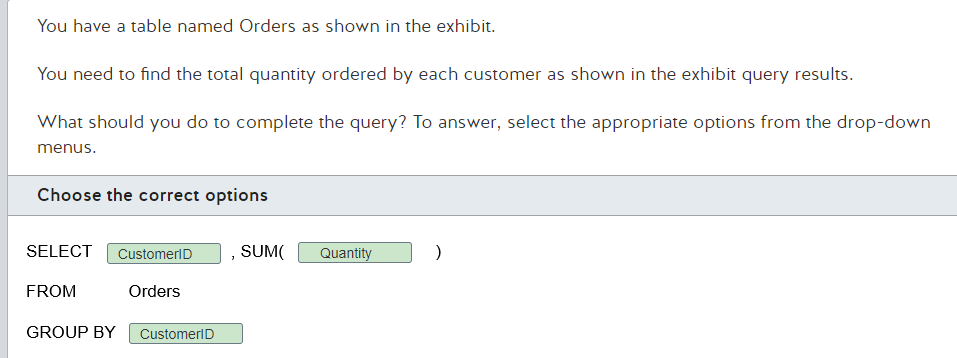
Descripción generada automáticamente

You should use the UPDATE statement. The UPDATE statement can update data in the existing rows of a table. You can update the stock quantity to 250 for the product with id 5323 by using the following query:  
  
UPDATE SET StockQuantity = 250 WHERE ProductId = 5323;  
  
You should not use the TRUNCATE statement. The TRUNCATE statement removes all rows from a table without logging the individual row deletions.  
  
You should not use the INSERT statement. The INSERT statement is used to add new rows in a table.  
  
You should not use the CREATE statement. The CREATE statement is used to define database objects, like tables, views, indexes, and others.  
  
**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
[UPDATE (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql?view=sql-server-ver15)  
  
[TRUNCATE TABLE (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/statements/truncate-table-transact-sql?view=sql-server-ver15)  
  
[INSERT (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/statements/insert-transact-sql?view=sql-server-ver15)  
  
[CREATE TABLE (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver15)

4. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

ALTER is a DDL statement type. DDL statements are used to define data structures in a relational database, like tables, views, and indexes. Other DDL statements are CREATE and DROP.  
  
SELECT, INSERT, and DELETE are not DDL statements. These statements are classified as Data Manipulation Language (DML) statements. DML statements are used to manipulate information stored in a relational database.  
  
**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
  
[Transact-SQL statements](https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver15)

5. 

Tabla

Descripción generada automáticamente

You should use the following query to find the total quantity ordered by each customer:  
  
SELECT CustomerID, SUM(Quantity)  
FROM Orders  
GROUP BY CustomerID  
  
You use the SUM() aggregate function to calculate the total quantity. To be able to see the total quantity bought by each customer, you need to add CustomerID to the GROUP BY clause allowing it to show aggregated results by CustomerID.  
  
**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
  
[SELECT - GROUP BY- Transact-SQL](https://docs.microsoft.com/en-us/sql/t-sql/queries/select-group-by-transact-sql?view=sql-server-ver15)  
  
[SUM (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/functions/sum-transact-sql?view=sql-server-ver15)

6. Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Data warehouses are relational datastores where the schema is optimized for analytics. It is denormalized to allow for faster and simpler queries. Both star and snowflake schemas are examples; they have a central fact table and orbiting dimension tables (with hierarchies in the case of snowflake).  
  
Data lakes are file stores used for high performance access. They can store files containing structured, semi-structured, or unstructured data, or even a mix of the three.   
  
Data lakehouses are a hybrid of a data warehouse and a lake; they store files with an abstracted relational storage layer, which can be queried using SQL.  
  
**References**[Explore analytical data stores](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/4-analytical-data-stores)

7.- Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

You should use the DROP command to remove a database table and its content. The DROP command is an example of a Data Definition Language (DDL) command. DDL commands are used to create, modify, and drop database objects.  
  
You should not use the ALTER command. ALTER is also a DDL command, but it is used to modify a database object, such as adding a column to a table.  
  
You should not use the DELETE or UPDATE commands. These are both Data Manipulation Language (DML) commands that are used to manipulate data, such as running queries to retrieve data. The DELETE command is used to delete rows from a table. The UPDATE command is used to edit column values.  
  
**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
  
[Transact-SQL statements](https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver15)  
  
[DROP TABLE (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/statements/drop-table-transact-sql?view=sql-server-ver15)

8.- Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Azure SQL Managed Instance. Azure SQL Managed Instance is used for deployment where you need to have complete feature parity with SQL Server on-premises. You should use a managed instance if your database uses features like SQL Server Agent and Database Mail.  
  
You should not use Azure SQL Database. Azure SQL Database gives you a single database in the cloud with minimal cost and administration, in which you can create databases and tables. Azure manages administrative tasks such as backup and recovery. However, Azure SQL Database does not implement features like SQL Server Agent and Database Mail, which is required in this scenario.  
  
You should not use SQL Server on Azure Virtual Machines. SQL Server on Azure Virtual Machines gives you a similar administrative experience as your on-premises server, which will therefore increase your administrative effort.  
  
You should not use an Azure database for MariaDB. An Azure database for MariaDB allows you to have MariaDB community edition implementation in Azure. MariaDB is a relational database management system.  
  
**References**  
  
[Migration overview: SQL Server to Azure SQL Managed Instance](https://docs.microsoft.com/en-us/azure/azure-sql/migration-guides/managed-instance/sql-server-to-managed-instance-overview)  
  
[Features comparison: Azure SQL Database and Azure SQL Managed Instance](https://docs.microsoft.com/en-us/azure/azure-sql/database/features-comparison)  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)  
  
[Explore fundamental relational data concepts](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-azure-sql-database)  
  
[Describe Azure services for open-source databases](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/3-azure-database-open-source)

9.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Data is organized in tables containing rows and columns in a relational database. Relational databases use multiple tables with rows and columns to store structured data, with each row containing the same set of columns.  
  
Data is not organized in documents containing fields and values in a relational database. Documents containing fields and values are a type of semi-structured data with a more flexible structure. Documents are usually represented in JavaScript Object Notation (JSON) format, and documents can have different fields to represent the same class of information.  
  
Data is not organized in tables containing keys and values in a relational database. This semi-structured data is used in a key-value store, which is similar to a relational table. However, each row (represented as key) can have a different set of columns (represented as value).  
  
Data is not organized in graphs containing edges and nodes in a relational database. This semi-structured data is used by graph databases, which are specialized to store and query information about complex relationships.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)

10.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should select Azure File Sync. The Azure File Sync service connects an on-premises server with cached copies of files with Azure File Storage data. This will allow users to access files via the cached copies inside your on-premises network, which improves performance.  
  
You should not select AzCopy. This utility allows users to upload files to Azure File Storage.  
  
You should not select Azure storage account. A storage account is a shared pool for storage. You can use it to create Azure file shares or Blob Storage containers.  
  
You should not select Azure Files AD Authentication. This allows users to authenticate via single sign-on (SSO) when they access an Azure file share and a directory, just as they would with Office 365 applications. There would be no improvement in file-serving performance.  
  
**References**[Overview of Azure Files identity-based authentication options for SMB access](https://docs.microsoft.com/en-us/azure/storage/files/storage-files-active-directory-overview)  
  
[What is Azure File Sync?](https://docs.microsoft.com/en-us/azure/storage/file-sync/file-sync-introduction)  
  
[What is Azure Files?](https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction)  
  
[Explore Azure Files](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/4-azure-files)

11.

Tabla

Descripción generada automáticamente

Azure Database for PostgreSQL can be deployed as a single server or as a cluster. You can deploy Azure Database for PostgreSQL as a single server or as a Hyperscale (Citus) cluster. Azure Database for PostgreSQL Hyperscale (Citus) can horizontally scale queries across multiple machines by using sharding.  
  
Azure Database for PostgreSQL enforces TLS connections by default. Enforcing TLS connections improves security by encrypting the connection between the client and the database server. You can disable the TLS connection in single server deployments, but this is not advised.  
  
Azure Database for PostgreSQL supports Azure AD authentication. You can configure Azure AD authentication with Azure Database for PostgreSQL to enable users to connect to the database using their own credentials. This centralizes the users' management in one place.  
  
**References**  
  
[What is Azure Database for PostgreSQL?](https://docs.microsoft.com/en-us/azure/postgresql/overview)  
  
[Configure TLS connectivity in Azure Database for PostgreSQL - Single Server](https://docs.microsoft.com/en-us/azure/postgresql/concepts-ssl-connection-security)  
  
[Configure TLS in Azure Database for PostgreSQL - Hyperscale (Citus)](https://docs.microsoft.com/en-us/azure/postgresql/concepts-hyperscale-ssl-connection-security)  
  
[Use Azure Active Directory for authenticating with PostgreSQL](https://docs.microsoft.com/en-us/azure/postgresql/concepts-aad-authentication)

12.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should create a report on Power BI Desktop. In a common workflow, you begin by connecting to data sources and building a report in Power BI Desktop. You can also create reports on Power BI service but with limited access to data sources.  
  
Then, you should share a report on Power BI service. You can publish and share reports on a Power BI service workspace to make them available to end users.  
  
Finally, you should view and interact with reports on Power BI mobile. After a report is shared on Power BI service, you can view and interact with this report using Power BI mobile. You can use Power BI service to interact and view reports with end users with desktop access only.  
  
You should not create or share a report on Power BI mobile. Power BI mobile apps can only be used to view and interact with reports.  
  
**References**  
  
[Describe Power BI tools and workflow](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-data-visualization/2-power-bi)  
  
[What is Power BI?](https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview)  
  
[Comparing Power BI Desktop and the Power BI service](https://docs.microsoft.com/en-us/power-bi/fundamentals/service-service-vs-desktop)  
  
[What are the Power BI mobile apps?](https://docs.microsoft.com/en-us/power-bi/consumer/mobile/mobile-apps-for-mobile-devices)

13.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Relational databases store structured data in tables. Records in separate tables are related to each other using keys, which are columns storing unique values per table row. Tables allow the storage of records in the same structure, containing the same attributes. Structured data is usually stored in a relational fashion; it minimizes storage costs through reducing duplication of data. Structured data is commonly queried using SQL (structured query language), which is why non-relational databases that store semi-structured data are sometimes referred to as NoSQL.  
  
Key-value databases are non-relational and can be used to store semi-structured data. Semi-structured data allows for variation between each instance of a data entity. For example, one record may contain a single email address as an attribute, another may contain three, and another may not contain any email addresses. Key-value databases allow two components of data to be stored related to an individual record: a unique key and a value, which can contain different types, formats and amounts of data per record.  
  
Column family databases are non-relational and can be used to store semi-structured data. Tables in column family databases can be divided into groups (column-families), holding sets of columns that are related to each other.  
  
Graph databases are non-relational and can be used to store semi-structured data. Graph databases contain an additional dimension to many other types of database; they store entities as nodes and allow relationships to other nodes (edges) to have a direction. For instance, a parent/child relationship between two person nodes would indicate which is the parent and which is the child.  
  
**References**[Explore databases](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-databases)[Relational database](https://en.wikipedia.org/wiki/Relational_database)

14.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Table. Table visuals allow you to display data in two dimensions with columns and rows, like a spreadsheet.  
  
You should also use Card. Card visuals display one or more data points about a particular row.  
  
You should not use Funnel chart. Funnel charts are used to display a process with stages; for instance, a sales pipeline.  
  
You should not use Key influencers chart. Key influencers chart are used to help users understand the reasons behind a metric.  
  
You should not select Basic map. Basic maps allow you to associate data with a geographic area and display it on a map view.  
  
**References**  
  
[Visualization types in Power BI](https://docs.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-types-for-reports-and-q-and-a)  
  
[Data modeling and visualizations](https://docs.microsoft.com/en-us/learn/modules/introduction-power-bi/2a-data-modeling-visualizations)

15.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Email is an attribute of the Customer entity. It is a column on the Customer data table. Attributes represent characteristics of a data entity, which models a real-world object. For tabular data, the table is the data entity and columns within the table are attributes.  
  
Purchase Date is an attribute of the Purchase entity. It is a column on the Purchase data table.  
  
Purchase is a data table, analogous to an entity. Data entities model real-world objects and their characteristics are attributes. In structured, tabular data, data entities are often modeled as tables and the individual columns are their attributes (or characteristics). The individual columns on the Purchase data table (ID, Purchase Date, Customer ID, Product ID, Quantity and Payment Taken) are attributes of the Purchase entity.  
  
Product is also a data table. The individual columns on the Purchase data table (ID, Name and Type) are attributes of the Product entity.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

16.

Tabla

Descripción generada automáticamente

OLTP systems are used to record day-to-day business activities and interactions as they occur. This includes activities such as orders taken, services performed, and payments received or made.  
  
In an OLTP system, data is highly normalized with the schema strongly enforced on write. OLTP systems are usually structured around a relational data store supporting transactional applications.  
  
An OLTP workload has heavy write requirements with minimal (in comparison) read requirements.  
  
In an OLTP environment, changes made are rolled back automatically if a transaction is not completed so that no transaction is left in a partially completed state. This is known as atomicity and is a requirement for OLTP.  
  
**References**  
  
[Explore transactional data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/5-transactional-data-processing)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)

17.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

A primary key is a constraint used to enforce data integrity in relational databases by indicating which column (or combination of columns) uniquely identifies each row in a table. The primary key must have a unique value across a table and is also used to create relationships between different tables.  
  
A foreign key is a constraint used to create relationships between tables by referencing a primary key from other tables.  
  
A heap is a table that does not have a clustered index implemented. In a heap, data is stored in an unordered structure.  
  
**References**  
  
[Explore the characteristics of relational data](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/2-explore-characteristics)  
  
[Primary and Foreign Key Constraints](https://docs.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver15)  
  
[Heaps (Tables without Clustered Indexes)](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-ver15)

18.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Azure Cosmos DB has very low latency for data reads and writes, and it is very fast and elastically scalable. It will support the very fast data reads and writes required for an online game. Halo 5: Guardians uses Azure Cosmos DB as its database tier.  
  
Azure Cosmos DB supports multiple geographic regions. You can enable multi-region writes, adding the Azure regions of your choice to your Cosmos DB account so that globally distributed users can each work with data in their local replica.  
  
Azure Cosmos DB is platform as a service (PaaS) application, and so the Database Administrators do need to perform some administration with this model.  
  
Azure Cosmos DB is a NoSQL database, meaning that it is designed to support semi-structured data, such as that used by an online game.  
  
Azure Cosmos DB supports multiple APIs such as Cassandra, Gremin, and Table.  
  
**References**[Welcome to Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)[Describe Azure Cosmos DB](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/2-describe-azure-cosmos-db)  
  
[Common Azure Cosmos DB use cases](https://docs.microsoft.com/en-us/azure/cosmos-db/use-cases)

19.

Tabla

Descripción generada automáticamente

Azure Data Lake Storage is not built on top of Azure File storage. Azure Data Lake Storage Gen2 is built on top of Azure Blob storage, combining the features of the previous generation of Azure Data Lake Storage with Azure Blob storage.  
  
Azure Data Lake Storage is capable of storing a large amount of data in a cost-effective way. Azure Data Lake Storage can store large amounts of data, such as hundreds of terabytes and more, and you only pay for what you use. You can reduce the storage cost even more by using features such as storage lifecycle to archive or move data that is not used frequently to cheaper storage tiers.  
  
Azure Data Lake Storage enables hierarchical namespace compatible with Hadoop Distributed File System (HDFS). Azure Data Lake Storage provides a layer to access Azure Blob Storage data as an HDFS storage, including support to organize files in directories and subdirectories, allowing you to examine large quantities of data quickly.  
  
**References**  
  
[Explore Azure DataLake Storage Gen2](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/3-azure-data-lake-gen2)  
  
[Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)  
  
[Optimize costs by automatically managing the data lifecycle](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-lifecycle-management-concepts?tabs=azure-portal)

20.

Tabla

Descripción generada automáticamente

You can create reports and dashboards in the Power BI service. Although you can create basic reports and dashboards in the Power BI service, it is more common to create reports in Power BI Desktop for a complete design experience and for access to more data sources.  
  
You can share and distribute reports in the Power BI service. You can create workspaces in the Power BI service to collaborate and share your reports with other team members and your company.  
  
You cannot design data modeling in the Power BI service. You should use Power BI Desktop instead to design data modeling, like creating custom columns and managing model relationships.  
  
**References**  
  
[Describe Power BI tools and workflow](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-data-visualization/2-power-bi)  
  
[What is the Power BI service?](https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-service-overview)  
  
[Comparing Power BI Desktop and the Power BI service](https://docs.microsoft.com/en-us/power-bi/fundamentals/service-service-vs-desktop)

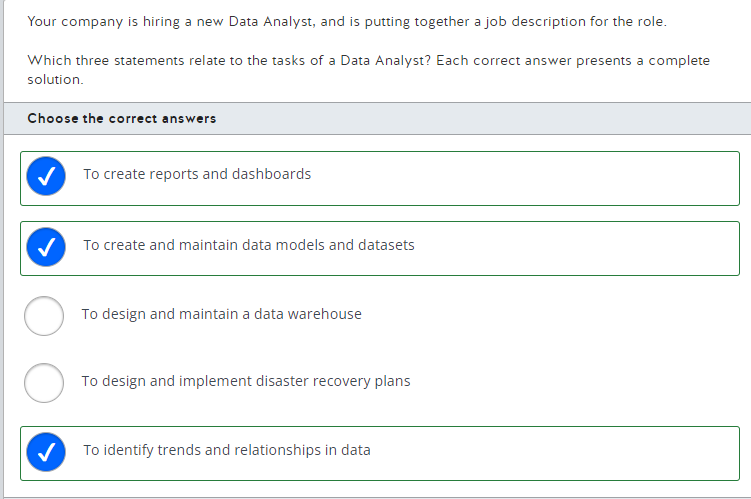
21.

Tabla

Descripción generada automáticamente

Azure Databricks is an analytics platform based on Apache Spark. Azure Databricks is a complete platform for big data processing, streaming, and machine learning optimized for the Microsoft Azure cloud services platform and built on top of Apache Spark.  
  
Azure Databricks can handle batch and stream processing. You can also perform real-time data processing and event streaming from Azure Event Hubs with Azure Databricks.  
  
Azure Databricks provides an interactive workspace for exploration and data visualization. Azure Databricks provides a workspace for collaboration between data scientists, data engineers, and business analysts. You can run notebooks in R, Python, Scala, or SQL, and interact with the data very quickly.  
  
**References**  
  
[What is Azure Databricks?](https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks)  
  
[Explore Apache Spark on Microsoft Azure](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/6-spark-streaming)  
  
[Tutorial: Stream data into Azure Databricks using Event Hubs](https://docs.microsoft.com/en-us/azure/databricks/scenarios/databricks-stream-from-eventhubs)

22.



The new Data Analyst will need to create reports and dashboards. This will be the main focus of a Data Analyst's job. They use tools such as Power BI and Tableau to tell a story with data, creating visualizations that illustrate the trends and relationships they have uncovered for business users.  
  
The Data Analyst will identify trends and relationships in data. It is part of a Data Analyst's job to use their skills to uncover insights in data that are useful to the business.  
  
The Data Analyst will also create and maintain data models and datasets. When creating reports, Data Analysts will ingest raw data and then shape it into a different model, which will provide the correct basis and optimal performance for the particular report they are building. They will often create and maintain curated sets of data that are useful for people to report on, which can belong to models or be in a stand-alone format.  
  
The Data Analyst will not need to design and implement disaster recovery plans. This task involves working with the data at source (for example, within an Azure SQL database), which is not part of a Data Analyst's job. This task is part of a Database Administrator's job.  
  
The Data Analyst's job will not include designing and maintaining a data warehouse. A data warehouse is a central store of data, ingesting data from many sources and storing it in a way that meets the company's requirements for data transformation or analytics. This task would form part of the job of a Data Engineer.  
  
**References**  
  
[Explore job roles in the world of data](https://docs.microsoft.com/en-us/learn/modules/explore-roles-responsibilities-world-of-data/2-explore-job-roles)

23.

Tabla

Descripción generada automáticamente

The pipeline is orchestrating an ETL process. ETL stands for extract, transform, load. This means that some manipulation operations are performed on the data before reaching its final destination. This operation is called data enrichment (data is being added). As the geocoding is being applied before the data reaches the target warehouse, this process is an example of ETL.  
  
The pipeline is not orchestrating an ELT process. ELT stands for extract, load, transform. Manipulation operations are performed in-situ in the target system. The data is originally loaded into the target system as it was in the source system. As the geocoding is being applied before the data reaches the target warehouse, this process is ETL.  
  
Geocoding the employee records is a pipeline activity. Activities are operations that are performed on a data pipeline.  
  
**References**  
  
[Extract, transform, and load (ETL)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl)  
  
[Explore data ingestion pipelines](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/3-data-ingestion-pipelines)  
  
[Pipelines and activities in Azure Data Factory and Azure Synapse Analytics](https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipelines-activities?tabs=data-factory)

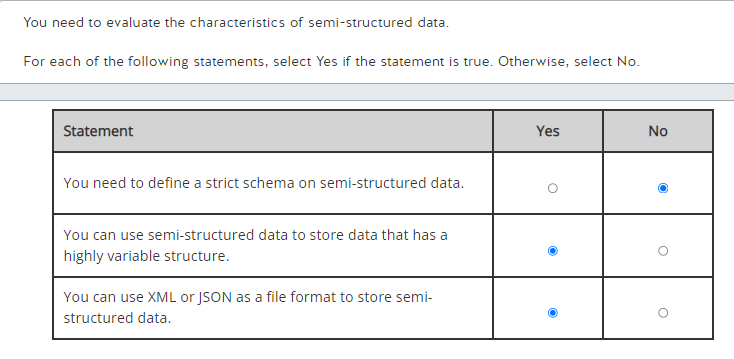
24.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Microsoft recommends that any new data project created from scratch uses the Core (SQL) API. The Core (SQL) API is used to store semi-structured data in a document and uses a SQL-like query language to manipulate the data stored in the documents. This includes applications with key/value data. This type of application is also supported by the Table API, but the Core (SQL) API is recommended as the best solution because it provides improved indexing and a richer query experience.  
  
When creating a new application that analyzes detailed relationship information for non-relational data you should use the Gremlin API. This is one of the few cases where the Core (SQL) API is not recommended as the best solution. The statement described a graph database to which the Gremlin API is specifically suited.  
  
When moving application data to the cloud that uses semi-structured documents to store data, you should use the Core (SQL) API. The Core (SQL) API gives you the ability to create, query, and update data documents.  
  
You should use the Cassandra API when moving column-family format data to the cloud to support an existing application. Microsoft suggests limiting the use of the Cassandra API to support existing data, such as when moving data to the cloud. The Cassandra API is specifically designed to support column-family data.  
  
None of the data applications should use the Azure Table API or MongoDB API. Microsoft recommends that either should only be used when supporting an existing data application, such as when moving an Azure Table store or MongoDB to Azure.  
  
**References**  
  
[Identify Azure Cosmos DB APIs](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/3-cosmos-db-apis)  
  
[Work with Azure Cosmos DB](https://docs.microsoft.com/en-us/learn/modules/choose-api-for-cosmos-db/3-analyze-the-decision-criteria)  
  
[Welcome to Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)  
  
[Non-relational data and NoSQL](https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data)

25.



You do not need to define a strict schema on semi-structured data. With semi-structured data, you do not need to define a schema. It focuses on storing the entity attributes data as-is rather than manipulating the data in tables and columns, that follow a strict schema, like a relational database.  
  
You can use semi-structured data to store data that has a highly variable structure. You can store entities with different fields with semi-structured data, for example, a customer in an e-commerce platform could have multiple contact numbers or addresses, while another customer could have only one contact number. Semi-structured data provides you with this flexibility.  
  
You can use Extensible Markup Language (XML) or JavaScript Object Notation (JSON) as a file format to store semi-structured data. Both file formats are flexible enough to store semi-structured data. You can add new attributes to entities by adding new keys to a JSON object or adding new elements or attributes to an XML document.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Explore file storage](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/3-file-storage)

26.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Azure Blob is the only Azure storage option that supports access tiers. The default is the Hot tier, which is designed for frequently accessed data. The Cool tier is optimized for data that will be stored for at least 30 days. The Cool tier has lower storage costs than the Hot tier but higher costs for early access. The Archive tier is designed for data that is rarely accessed and will remain in storage for at least 180 days. Access to Archive tier data requires the data to be rehydrated to a Hot or Cool tier. This can mean a latency of several hours. Access tier support requires Data Lake Storage Gen2.  
  
Azure Blob also supports two performance tiers. The Standard performance tier provides for high performance using hard disk-based storage media. The Premium performance provides greater throughput than the Standard tier and uses solid-state drive (SSD) media. The Standard and Premium tiers are also supported for other storage options including Azure File storage and Azure SQL Database.  
  
Azure Table and Azure File do not support access tiers. Access tiers is a feature supported through Cosmos DB. Azure Table and Azure File are distinct storage types and are not implemented through Cosmos DB APIs. Table storage is used for storing structured, non-relational data. File storage provides file storage with shared access, similar to a file server.  
  
**References**  
  
[Explore Azure blob storage](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/2-azure-blob-storage)  
  
[Introduction to Azure Blob storage](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction)  
  
[Hot, Cool, and Archive access tiers for blob data](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers?tabs=azure-portal)

27.

Tabla

Descripción generada automáticamente

Azure Data Factory can have multiple pipelines. Azure Data Factory allows you to create multiple data pipelines.  
  
Pipeline is a logical grouping of activities that performs a task. You can have multiple activities in a pipeline.  
  
Activities in a pipeline can either run sequentially or operate in parallel. An activity represents a step in a pipeline.  
  
**References**  
  
[What is Azure Data Factory?](https://docs.microsoft.com/en-us/azure/data-factory/introduction)  
  
[Explore data ingestion pipelines](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/3-data-ingestion-pipelines)

28.

Diagrama

Descripción generada automáticamente

Data ingestion is the process of combining your structured, semi-structured, and unstructured data into a common data store. Azure Data Factory is specifically designed to provide end-to-end support for extract-transform-load (ETL) operations for data warehouse data load. Often, moving data into a data warehouse requires more aggressive cleaning and transformation than is supported through Data Factory.  
  
The consolidated data is stored in Azure Blob storage through Azure Data Lake Storage. This gives you a flexible storage environment to give access to Azure Databricks for more intensive data analytics to prepare cleaned and transformed data.  
  
Native connectors let you move data at scale from Azure Databricks to Azure Synapse Analytics, which acts as a single hub for your structured data. From here, the data is available for detailed analysis and reporting, including using Azure Analysis Services to give end-users access to the data.  
  
Azure Cosmos DB is not part of the Azure data warehouse infrastructure as a non-relational data store.  
  
**References**  
  
[Enterprise data warehouse](https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/modern-data-warehouse)  
  
[What is Azure Data Factory?](https://docs.microsoft.com/en-us/azure/data-factory/introduction)  
  
[What is Azure Databricks?](https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks)  
  
[What is Azure Synapse Analytics?](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is)  
  
[Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)  
  
[Welcome to Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)

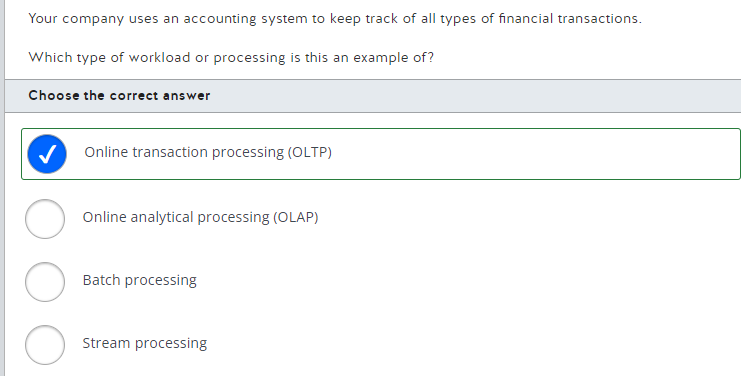
29.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should select Q&A. The Q&A visual gives users a question (text input) box that they can use to ask questions in natural language as well as a pre-populated list of suggested questions and receive visual answers.  
  
You should not select doughnut chart. A doughnut chart is similar to a pie chart; it shows the relationship of parts to a whole.  
  
You should not select gauge chart. These display a semi-circle gauge displaying progress toward a goal or KPI.  
  
You should not select Treemap. These are composed of colored rectangles and are used to represent hierarchical data or portions of parts toward a whole where there are too many parts to display in a pie or doughnut chart adequately.  
  
**References**  
  
[Data modeling and visualizations](https://docs.microsoft.com/en-us/learn/modules/introduction-power-bi/2a-data-modeling-visualizations)  
  
[Create a Q&A visual in a report in Power BI](https://docs.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-q-and-a)  
  
[Visualization types in Power BI](https://docs.microsoft.com/en-us/power-bi/visuals/power-bi-visualization-types-for-reports-and-q-and-a)

30.



This is an example of an OLTP workload. You can use OLTP workloads with transactional systems used in the day-to-day operations of an organization, like accounting, financial, and other systems, that require strong consistency for transactions.  
  
OLAP workloads organize large business databases and perform complex analytics, like data mining, without negatively affecting transactional systems.  
  
A stream processing handles a continuous stream of data used by time-critical operations.  
  
Batch processing collects a group of data within a scheduled time interval or when a certain amount of non-time-sensitive data has arrived.  
  
**References**  
  
[Explore transactional data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/5-transactional-data-processing)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)  
  
[Explore analytical data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/6-analytical-processing)  
  
[Online analytical processing (OLAP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing)  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)

31.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

A graph data store consists of edges and nodes used to store and query complex relationships among entities. It stores all the entities as nodes and builds the relationship between those entities by using edges.  
  
A key/value data store consists of a simple and quick data structure where you can store some information in a value that is identified by a key.  
  
A columnar data store consists of row identifiers and a group of information stored in a column. Each group of information is stored in independent columns.  
  
An object data store consists of large binary objects, such as images, media files, and other types of files.  
  
**References**  
  
[Explore Azure Storage for non-relational data](https://docs.microsoft.com/en-us/learn/modules/explore-concepts-of-non-relational-data/4-describe-types-nosql-databases)  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)

32.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Azure SQL Database for PaaS relational data services offerings. Azure SQL Database is a managed database server in the cloud.  
  
You can also use Azure SQL Database Managed Instance for PaaS relational data services offerings. Azure SQL Database Managed Instance allows you to run multiple databases in the same instance.  
  
You should not use SQL Server on Azure Virtual Machines. SQL Server is an Infrastructure-as-a-Service (IaaS) offering for data services. It allows you to run SQL Server on an Azure virtual machine by providing the relevant infrastructure.  
  
You should not use Azure Cosmos DB. Azure Cosmos DB is a PaaS offering mainly targeted for non-relational data stores. It allows you to store documents and objects.  
  
**References**  
  
[What is Azure SQL?](https://docs.microsoft.com/en-us/azure/azure-sql/azure-sql-iaas-vs-paas-what-is-overview)  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)  
  
[Explore fundamental relational data concepts](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/2-azure-data-services)  
  
[Microsoft Azure Data Fundamentals: Explore non-relational data in Azure](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-offerings-azure/5-explore-azure-cosmos-database)

33.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

A primary key is the name given to a specific column in a database table that stores a unique identifier for each row, which is analogous to an instance of a data entity.  
  
A foreign key is the name given to any column in a database table which references rows in another table by their own unique identifier. This allows the construction of relationships between two data entity instances. For example, the classroom table has a row with IDs 1 and 2. The teachers table has a row with ID 3. Both classroom rows can reference the teachers table row 3 to indicate that the same teacher is teaching in both classrooms.  
  
Rows are instances of a data entity; a set of values for each column in the database table. Rows contain a primary key value, which acts as a unique identifier for that instance of the data entity.  
  
An index is created from several columns to improve the speed of queries. They can (and usually do) include the primary key of the table.  
  
**References**[Indexes](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/indexes?view=sql-server-ver15)[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

34.

Interfaz de usuario gráfica

Descripción generada automáticamente

Comma-Separated Values (CSV) files are structured. They store plain text separated by field delimiters, often commas, creating a set schema. Each row of data must have the same set of attributes in the same order. You can save structured data files, like Excel spreadsheets, as .csv files to provide a flexible format, which many applications can use.  
  
Extensible Markup Language (XML) files are semi-structured. They store elements and attributes in tags in a hierarchical schema, which look like HTML tags, e.g. <customer />, but they can be defined according to data storage requirements. XML is a good choice for storing semi-structured data, as elements and tags are flexible.  
  
Binary Large Object (BLOB) files store raw binary data without a schema. BLOB files are used for unstructured data, such as audio and video.  
  
JavaScript Object Notation (JSON) files store data entities in a hierarchical schema, much like XML. Each entity can have a different set or number of attributes, making it a good choice to store semi-structured data.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Explore file storage](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/3-file-storage)

35.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

The primary key is highlighted. The primary key column contains values that are unique identifiers for a particular row. In this scenario, the ID column stores a unique number for each customer.   
  
Foreign keys are not highlighted. Foreign key columns contain values referencing the Primary key of another table. The City ID column is a foreign key column; it references the primary key values of the Customer table, normalizing the data. Normalization is the process of removing duplicate data. Referencing a row in the City table means that we do not have to store additional columns in the Customer table for every row in the City table.  
  
An instance of a customer entity is not highlighted. Each row of data shown in the exhibit represents a different customer, an instance of the customer data entity.  
  
A non-clustered index is not highlighted. A non-clustered index is a structure of a database table that works as an index page in a textbook. It is used to speed up queries.  
  
**References**  
  
[Understand normalization](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/3-normalization)  
  
[Explore databases](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-databases)

36.

Interfaz de usuario gráfica, Texto, Aplicación

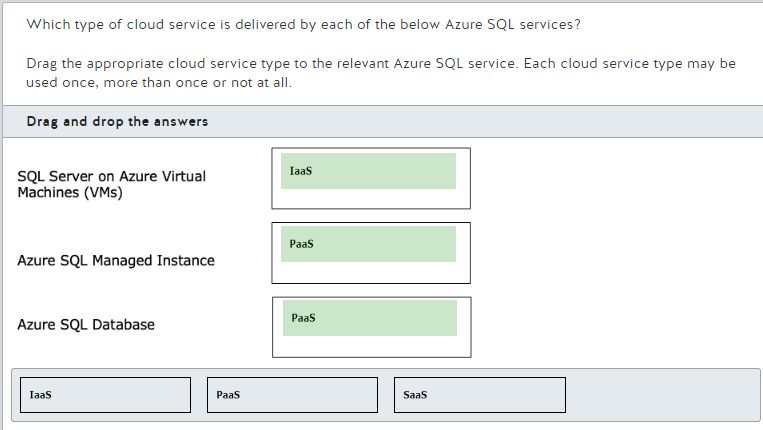
Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

The data model shown in the exhibit is an example of the star schema. In the star schema, fact tables are directly connected to dimension tables. In this case, the Customer and Product tables are dimensions and the Order table is a fact.  
  
The exhibit does not show the snowflake schema. In the snowflake schema, dimensions are interrelated with each other, and a dimension table links to the fact table. Dimension tables in the snowflake schema are highly normalized.  
  
You should not use the OLAP system. OLAP systems are designed to perform complex analysis and provide business intelligence.  
  
**References**  
  
[Describe core concepts of data modeling](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-data-visualization/3-data-modeling)  
  
[Understand star schema and the importance for Power BI](https://docs.microsoft.com/en-us/power-bi/guidance/star-schema)  
  
[Online analytical processing (OLAP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing)

37.



SQL Server on Azure Virtual Machines (VMs) is an Infrastructure as a Service (IaaS) service. This form of cloud service requires the most responsibility for the customer, as they need to manage everything other than the physical devices used. SQL Server on Azure Virtual Machines creates virtual hardware for customers to use as they wish.  
  
Azure SQL Managed Instance is a Platform as a Service (PaaS) service. It does not require customers to manage physical hardware or operating systems and includes automated software updates and backup; customers typically use this service to migrate existing systems to the cloud.  
  
Azure SQL Database is a Platform as a Service (PaaS) service. Similar to Azure SQL Managed Instance, this is a fully managed solution that handles backups, monitoring, and upgrades for the customer. Customers use this service as a high availability data layer for their cloud-based systems.  
  
SaaS stands for Software as a Service. SaaS does not require the customer to manage physical devices, operating systems, or updates to the software. Users only need to create accounts and manage access. An example of SaaS would be Microsoft Dynamics 365 or Power BI.  
  
**References**[Shared responsibility in the cloud](https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility)  
  
[What is SQL Server on Windows Azure Virtual Machines?](https://docs.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/sql-server-on-azure-vm-iaas-what-is-overview?view=azuresql)  
  
[What is Azure SQL Managed Instance?](https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview?view=azuresql)  
  
[What is Azure SQL Database?](https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview?view=azuresql)  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)

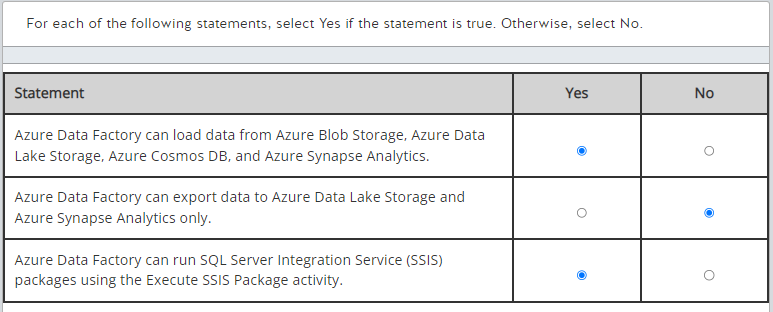
38.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

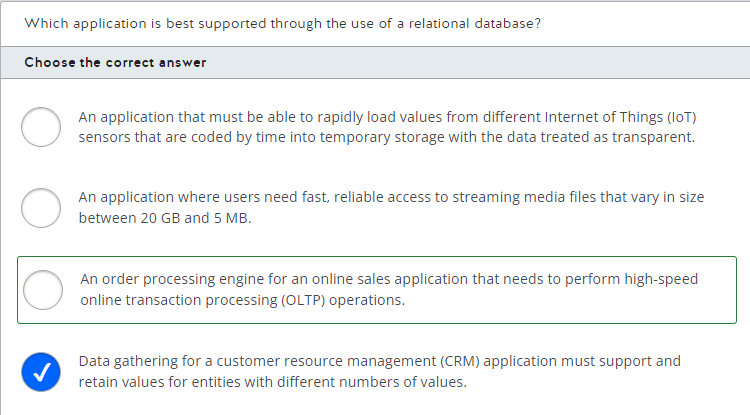
You should use Azure Blob storage to store training videos. Azure Blob storage allows you to store large object files such as images, videos, and virtual machines (VMs).  
  
You should not use Azure Table storage to store training videos. Azure Table storage allows you to store semi-structured data into key/value format. This means it stores data into a rows and columns format, but unlike a relational database, each row has a key and each column contains entire data value.  
  
You should not use Azure File storage to store training videos. Azure File storage allows you to create file shares in the cloud, which can be accessible for network users.  
  
You should not use Azure SQL database to store training videos. Azure SQL database allows you to store relational data in a cloud database.  
  
**References**  
  
[Explore Azure blob storage](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/2-azure-blob-storage)  
  
[Explore Azure Tables](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/5-azure-tables)  
  
[Explore Azure Files](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/4-azure-files)  
  
[Explore fundamental relational data concepts](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-azure-sql-database)

39.



Azure Data Factory can load data from Azure Blob Storage, Azure Data Lake Storage, Azure Cosmos DB, and Azure Synapse Analytics. You can even load data from services outside Azure, such as Amazon S3.  
  
Azure Data Factory can export data to Azure Data Lake Storage, Azure Synapse Analytics, and many other destinations, such as Azure SQL Database, Azure Blob Storage, and Azure Cosmos DB.  
  
Azure Data Factory can run SQL Server Integration Service (SSIS) packages using the Execute SSIS Package activity. To use the Execute SSIS Package activity, you need to configure the Azure-SSIS integration runtime (IR).  
  
**References**  
  
[What is Azure Data Factory?](https://docs.microsoft.com/en-us/azure/data-factory/introduction)  
  
[Explore data ingestion pipelines](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/3-data-ingestion-pipelines)  
  
[Azure Data Factory and Azure Synapse Analytics connector overview](https://docs.microsoft.com/en-us/azure/data-factory/connector-overview)  
  
[Run an SSIS package with the Execute SSIS Package activity in Azure portal](https://docs.microsoft.com/en-us/azure/data-factory/how-to-invoke-ssis-package-ssis-activity?tabs=data-factory)

40.

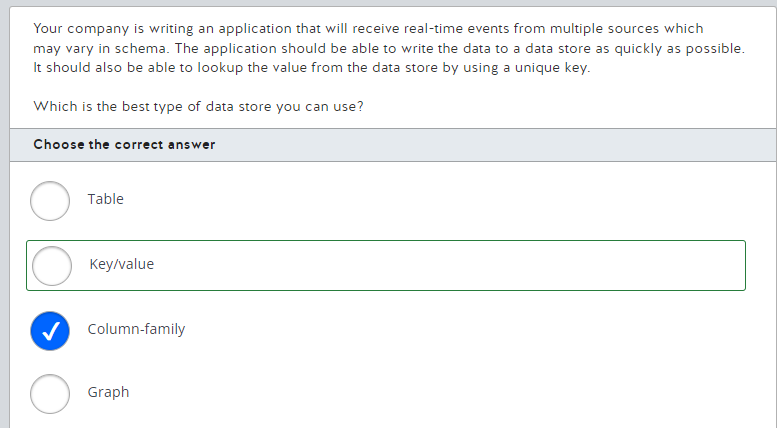


The application best suited to a relational data store is an order processing engine for an online sales application that needs to perform high-speed online transaction processing (OLTP) operations. Characteristics of data in relational databases include:

* Highly normalized with enforced schemas
* Requires high integrity and strong consistency
* Relationships are maintained between data tables

In addition to order management, relational databases are typically used to support inventory control and accounting applications.  
  
For an application where users need fast, reliable access to streaming media files that vary in size between 20 GB and 5 MB, you should choose an object storage solution such as Azure Blob storage or Azure Data Lake Storage Gen2.  
  
When gathering data for a customer resource management (CRM) application that must support and retain values for entities with different numbers of values, you would most likely use a document storage solution. Document data is semi-structured with each document internally defining its own schema. Each document is written and retrieved as a single block. You would most likely use Azure Cosmos DB as your storage solution.  
  
To support an application that must be able to rapidly load values from different IoT sensors that are coded by time into temporary storage with the data treated as transparent you would most likely use a key/value storage solution. You would most likely use the Azure Cosmos DB Table API or Azure Table storage as your data store.  
  
**References**  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)  
  
[What is Azure Table storage?](https://docs.microsoft.com/en-us/azure/storage/tables/table-storage-overview)

42.



You should use a key/value data store. A key/value data store functions essentially as a large hash table and is optimized for fast data writes. Each data row is referenced by a single key value. The only operations supported are simple query, insert, and delete operations. Data updates require the application to rewrite the data for the entire value. Queries can be run by a key or a range of keys.  
  
You should not use a column-family (columnar) data store. A column-family data store is similar to a relational data store in that data is organized as rows and columns, but the columns are divided into column families that can store multiple values in a single column. A row does not necessarily have a value in each column family. Columns within a column family are physically stored in the same file.  
  
You should not use a table data store. A table data store uses a row and column data format with the data somewhat normalized but the same schema is not enforced across all rows. Each row can have a different number of columns. In Azure Table storage, data is organized based on a partition key and a row key. The partition key identifies the partition in which the data is stored, and the row uniquely identifies the row within the partition.  
  
You should not use a graph data store. A graph data store is designed to support extensive, complex relationships between entities. This helps to make it easier to perform complex relation analysis.  
  
**References**  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)  
  
[Non-relational data and NoSQL](https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data)  
  
[Explore databases](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-databases)

43.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use a document database. A document data store consists of entities that have their related data stored in a single document. This document supports a flexible schema and the entity data is usually stored in JSON format. They can be used for maintaining the user profile information, making it possible to store all information in a single block.  
  
You should not use a key-value, non-relational data store. Key-value stores are highly optimized for simple data structures. A Key-value store associates each data value with a key that can be used to access the data.  
  
You should not use a column-family database. Column-family databases store data in rows and columns, grouping related columns into columns-families that are used together. Instead of a document-based database, you can use a column-family database to retrieve only the required columns-families instead of all user data.  
  
You should not use a graph database. Graph databases store information in the form of edges and nodes. They are used to represent complex relationships such as social interactivity.  
  
**References**  
  
[Explore Azure Storage for non-relational data](https://docs.microsoft.com/en-us/learn/modules/explore-concepts-of-non-relational-data/4-describe-types-nosql-databases)  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)

44.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Less data to store is a characteristic of normalization. Normalization removes redundant data by removing duplication, resulting in less data overall. Potentially, this could lead to a reduction in your storage costs.  
  
Improved data integrity is also a characteristic of normalization. Normalization removes duplicate data; every time you have the same data added in more than one place, you increase the possibility of being updated inconsistently due to an error.   
  
Fewer tables are not a characteristic of normalization. Normalization increases the number of tables by adding each data entity into a different table and referencing records in other tables via foreign key columns.  
  
Reduced query complexity is not a characteristic of normalization. It is simpler to write a query on one table than on two, as you will need to join the two tables via a relationship between the data to write your query. Normalization increases the number of tables, which means that you will need to use more joins in your queries and increase their complexity.  
  
Fewer foreign keys is not a characteristic of normalization. Normalization increases the number of tables by splitting duplicated data into its own table, which is then referenced in the original table via a foreign key. Normalization will therefore increase the number of foreign keys in the database.  
  
**References**  
  
[Understand normalization](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/3-normalization)

45.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

You should use the Power Query Editor. This is a tool within Power BI Desktop that allows you to interact with data by applying a series of transformations that are recorded for you as steps, which can be undone if necessary.  
  
You should not use Report view. This is the canvas where you can drag your visuals to create your report using data you have already imported, transformed, and modeled.  
  
You should not use Publish to Power BI Service. This button takes the Power BI report you have built using Power BI Desktop and send it to the Power BI Service to allow you to share it with colleagues.  
  
You should not use Connect to data. This function allows you to connect to raw data sources, for instance, and Azure SQL database. This is the step before transforming and shaping your data before the model is created.  
  
**References**[Data modeling and visualizations](https://docs.microsoft.com/en-us/learn/modules/introduction-power-bi/2a-data-modeling-visualizations)  
  
[Get started with Power BI Desktop](https://docs.microsoft.com/en-us/power-bi/fundamentals/desktop-getting-started)

46.

Tabla

Descripción generada automáticamente

Online analytical processing (OLAP) systems are designed to perform complex analyses and provide business intelligence. OLAP is used for analytical workloads, such as:

* Generating complex ad-hoc reports that include several aggregations
* Performing big data analysis on a NoSQL database

Online transaction processing (OLTP) systems are designed to perform business transactions as they occur. OLTP is used for transactional workloads, such as:

* Performing e-commerce transactions
* Tracking inventory management systems

**References**  
  
[Explore analytical data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/6-analytical-processing)  
  
[Online analytical processing (OLAP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)

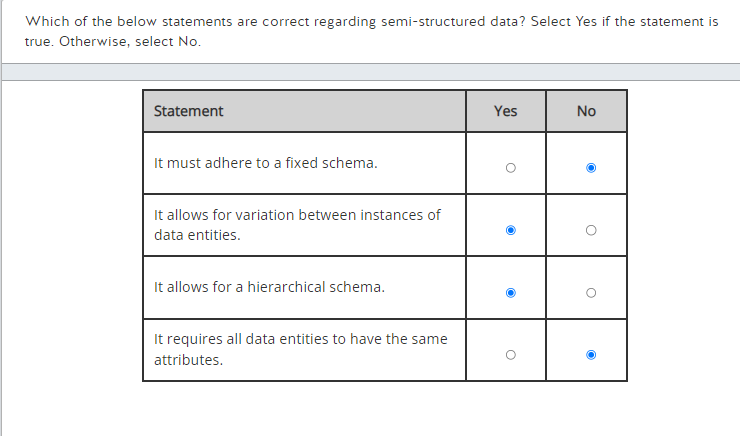
47.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

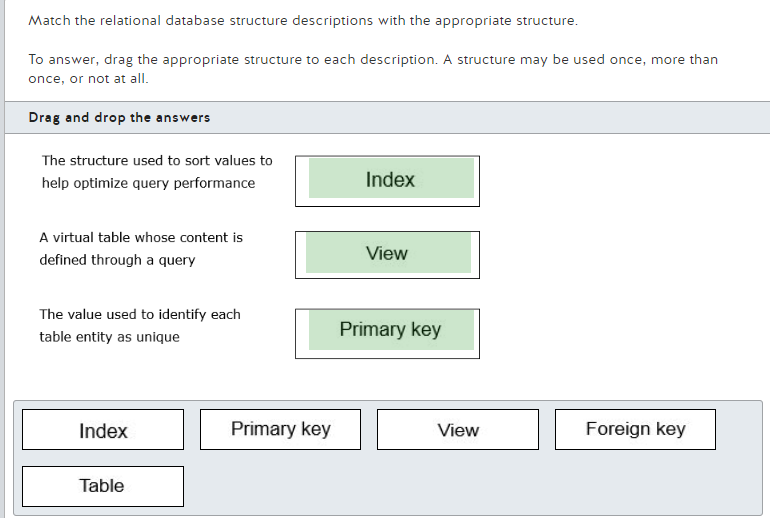
Azure File storage supports direct mounting by Windows, macOS, and Linux. This includes support for concurrent access from the cloud and on-premises. Azure File storage can be used to supplement or replace on-premises file server shares.  
  
Azure File storage does not allow you to select the underlying hardware and operating system. Azure File storage is implemented as a serverless file service in which you have neither direct access to, or administrative responsibilities for, the underlying architecture. The one infrastructure choice you can make is between hard disk (HDD) standard file shares and solid-state disk (SSD) premium file shares.  
  
Azure File storage is not the recommended storage solution for key/value storage implementation. Microsoft recommends Azure Cosmos DB Core (SQL) API for new key/value requirements. Key/value storage is also supported by Azure Table storage and Cosmos DB Table API.  
  
Azure File storage does not support redundancy across multiple regions by default. Standard file shares support locally-redundant storage (LRS) by default with options for zone redundant storage (ZRS), geo-redundant storage (GRS), and geo-zone-redundant storage (GZRS). Replication across multiple regions is supported as an option, but not as a default setting. The large file share feature and premium file shares support LRS and ZRS only.  
  
**References**  
  
[Explore Azure Files](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/4-azure-files)  
  
[What is Azure Files?](https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction)  
  
[Create an Azure file share](https://docs.microsoft.com/en-us/azure/storage/files/storage-how-to-create-file-share?tabs=azure-portal)

48.



Semi-structured data does not need to adhere to a fixed schema. Semi-structured data allows for flexibility in the data being stored. Each instance of a data element (e.g. a customer) can have different attributes, which can be stored in a different order. Structured data requires a fixed schema; that is, each row of data must contain the same set of attributes in the same order.  
  
Semi-structured data allows for variation between instances of data entities. Semi-structured data allows the storage of different sets of attributes per instance of a data entity. For example, one customer may have two email addresses and no mobile phone number, and a second customer might have three mobile phone numbers and one email address.  
  
Semi-structured data allows for a hierarchical schema. Semi-structured data allows data entities to be set up in a hierarchical fashion; that is, some entities can be modeled in a parent/child relationship. For example, it can represent a complex manager and employee relationship, whereby an employee may report to many managers, or a manager may be responsible for many employees, potentially sharing responsibility for some employees with other managers.  
  
Semi-structured data does not require all data entities to have the same attributes. This is true for structured data. Semi-structured data allows for variation between instances of data entities; different attributes and numbers of the same attribute may be recorded for each instance.  
  
**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

49.



An index is the structure used to sort values to help optimize query performance. Most relational database management systems (RDMSs) support clustered and nonclustered indexes. The difference is that a table is physically sorted in clustered index order when it is used and a nonclustered index does not change the table order, but it contains pointers to the appropriate table rows.  
  
A view is a virtual table whose contents are defined through a query. A simple view might look like the following:  
  
CREATE VIEW CustLabels  
AS  
SELECT c.CustomerNumber, c.CustomerName  
FROM Customers c  
  
This would create a view that includes the CustomerNumber and CustomerNamecolumn values. Once created, a view can be used as a data source in other queries.  
  
The primary key is the value used to identify each table entity as unique. The primary key value for each row will be unique.  
  
None of the descriptions are for a table. A table is the basic storage structure for a relational database consisting of rows and columns.  
  
None of the descriptions are for a foreign key. Foreign keys are used with primary keys to establish relationships between tables. The foreign key in one table is associated with the primary key in another table. A foreign key will be associated with only one primary key, but multiple foreign keys can be associated with the same primary key.  
  
**References**  
  
[Describe database objects](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/5-database-objects)  
  
[Clustered and nonclustered indexes described](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15)  
  
[Primary and Foreign Key Constraints](https://docs.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver15&viewFallbackFrom=sql-server-ver1)  
  
[Views](https://docs.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver15)

50.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

The Core (SQL) API allows developers to work with Cosmos DB data using SQL syntax.  
  
The Cassandra API is compatible with Apache Cassandra, a column-family structured database. It supports SQL syntax to allow developers to manipulate and retrieve data.  
  
The Gremlin API allows developers to work with graph data. It supports graph syntax.  
  
The Table API allows developers to work with key-value data in Cosmos DB. It supports requests based on a namespace, much like retrieving data from Azure Blob Storage.  
  
The MongoDB API supports MongoDB Query Language (MQL), which is object-oriented.  
  
**References**  
  
[Welcome to Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)  
  
[Choose an API in Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/choose-api)  
  
[Identify Azure Cosmos DB APIs](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/3-cosmos-db-apis)