1. X

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.

**Traducción**: Los datos se organizan en tablas que contienen filas y columnas en una base de datos relacional. Las bases de datos relacionales utilizan varias tablas con filas y columnas para almacenar datos estructurados, y cada fila contiene el mismo conjunto de columnas.

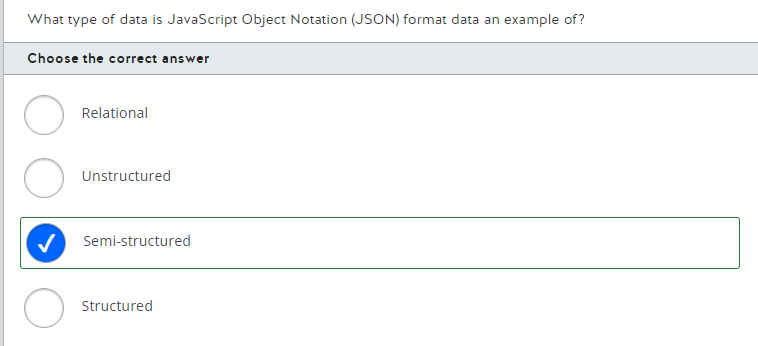
Los datos no están organizados en documentos que contienen campos y valores en una base de datos relacional. Los documentos que contienen campos y valores son un tipo de datos semiestructurados con una estructura más flexible. Los documentos generalmente se representan en formato de notación de objetos JavaScript (JSON), y los documentos pueden tener diferentes campos para representar la misma clase de información.

Los datos no están organizados en tablas que contienen claves y valores en una base de datos relacional. Estos datos semiestructurados se utilizan en un almacén de clave-valor, que es similar a una tabla relacional. Sin embargo, cada fila (representada como clave) puede tener un conjunto diferente de columnas (representadas como valor).

Los datos no están organizados en gráficos que contienen bordes y nodos en una base de datos relacional. Estos datos semiestructurados son utilizados por bases de datos de gráficos, que están especializadas para almacenar y consultar información sobre relaciones complejas.

**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)

1. D



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.

**Traducción**: Los corchetes se utilizan para encerrar el documento y cualquier subdocumento. Azure Cosmos DB es la solución de almacenamiento más común para este tipo de datos.

Los datos no estructurados incluyen archivos como archivos de video o audio sin estructura de esquema. Este tipo de datos generalmente se almacena en Azure Blob Storage.

Los datos estructurados y los datos relacionales se refieren a la misma estructura de datos con datos en un formato altamente normalizado y almacenados en múltiples tablas relacionadas. La solución de almacenamiento más común es algún tipo de sistema de gestión de bases de datos SQL.

**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)

1. D



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.

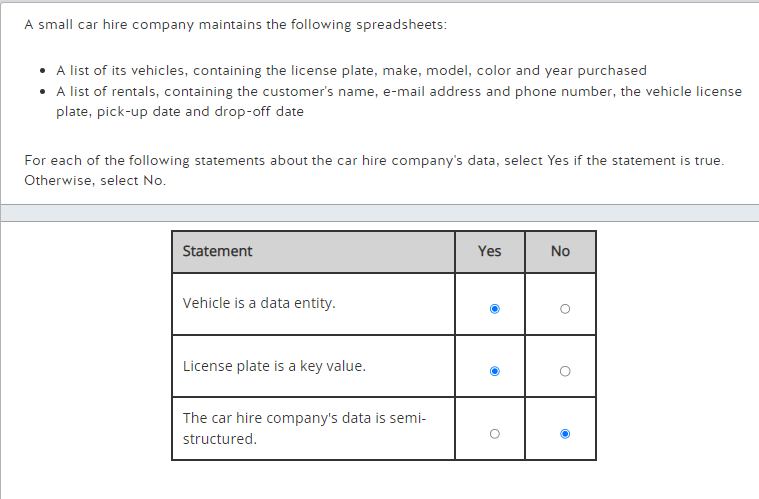
**Traducción**:

No necesita definir un esquema estricto en datos semiestructurados. Con datos semiestructurados, no necesita definir un esquema. Se enfoca en almacenar los datos de los atributos de la entidad tal cual en lugar de manipular los datos en tablas y columnas, que siguen un esquema estricto, como una base de datos relacional.

Puede utilizar datos semiestructurados para almacenar datos que tienen una estructura muy variable. Puede almacenar entidades con diferentes campos con datos semiestructurados, por ejemplo, un cliente en una plataforma de comercio electrónico podría tener varios números de contacto o direcciones, mientras que otro cliente podría tener solo un número de contacto. Los datos semiestructurados le brindan esta flexibilidad.

Puede utilizar el lenguaje de marcado extensible (XML) o la notación de objetos de JavaScript (JSON) como formato de archivo para almacenar datos semiestructurados. Ambos formatos de archivo son lo suficientemente flexibles para almacenar datos semiestructurados. Puede agregar nuevos atributos a las entidades agregando nuevas claves a un objeto JSON o agregando nuevos elementos o atributos a un documento XML.  
**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)

1. D



Vehicle is a data entity. A data entity models a real-world object, which has characteristics (attributes). The car hire company's data models two entities: vehicle and rental. In the case of the vehicle, its attributes are the license plate, make, model, color and year purchased.  
  
License plate is a key value. License plate is a unique value for vehicles. It is referenced in both the vehicle spreadsheet and the rental spreadsheet, allowing a single vehicle to be related to many rentals; this is called a one-to-many relationship. The license plate value acts as a foreign key for rentals.  
  
The car hire company's data is structured. Structured data is regular; the same set of information is recorded for each entity instance (in this instance, each line in the spreadsheet). For this reason, it is often referred to as tabular data. In semi-structured data, similar information with some variation is recorded for each instance and it is often hierarchical (one record can be a parent to another). For example, one record could record a home address and an e-mail address, and the next one might have a home address and a shipping address but not an e-mail address. For this car hire company, the same information is held for all records in the vehicle and rental spreadsheets, and so the data is structured. The rental and vehicle tables are related, and the two spreadsheets act as a basic relational database.

**Traducción**:

El vehículo es una entidad de datos. Una entidad de datos modela un objeto del mundo real, que tiene características (atributos). Los datos de la empresa de alquiler de coches modelan dos entidades: vehículo y alquiler. En el caso del vehículo, sus atributos son la placa, marca, modelo, color y año de adquisición.

La matrícula es un valor clave. La matrícula es un valor único para los vehículos. Se hace referencia tanto en la hoja de cálculo del vehículo como en la hoja de cálculo del alquiler, lo que permite relacionar un solo vehículo con muchos alquileres; esto se llama una relación de uno a muchos. El valor de la matrícula actúa como clave externa para los alquileres.

Los datos de la empresa de alquiler de coches están estructurados. Los datos estructurados son regulares; el mismo conjunto de información se registra para cada instancia de entidad (en esta instancia, cada línea en la hoja de cálculo). Por esta razón, a menudo se los denomina datos tabulares. En los datos semiestructurados, se registra información similar con alguna variación para cada instancia y, a menudo, es jerárquica (un registro puede ser el padre de otro). Por ejemplo, un registro podría registrar una dirección particular y una dirección de correo electrónico, y el siguiente podría tener una dirección particular y una dirección de envío, pero no una dirección de correo electrónico. Para esta empresa de alquiler de coches, se conserva la misma información para todos los registros en las hojas de cálculo de vehículos y alquiler, y así se estructuran los datos. Las tablas de alquiler y de vehículos están relacionadas, y las dos hojas de cálculo actúan como una base de datos relacional básica.

**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

1. D

Interfaz de usuario gráfica, Texto, 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.

**Traducción:**

El correo electrónico es un atributo de la entidad Cliente. Es una columna en la tabla de datos del Cliente. Los atributos representan características de una entidad de datos, que modela un objeto del mundo real. Para datos tabulares, la tabla es la entidad de datos y las columnas dentro de la tabla son atributos.

La fecha de compra es un atributo de la entidad de compra. Es una columna en la tabla de datos de Compra.

Compra es una tabla de datos, análoga a una entidad. Las entidades de datos modelan objetos del mundo real y sus características son atributos. En los datos tabulares estructurados, las entidades de datos a menudo se modelan como tablas y las columnas individuales son sus atributos (o características). Las columnas individuales en la tabla de datos de compra (ID, fecha de compra, ID de cliente, ID de producto, cantidad y pago recibido) son atributos de la entidad de compra.

El producto también es una tabla de datos. Las columnas individuales en la tabla de datos de Compra (ID, Nombre y Tipo) son atributos de la entidad Producto.

**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

Las bases de datos relacionales almacenan datos estructurados en tablas. Los registros en tablas separadas se relacionan entre sí mediante claves, que son columnas que almacenan valores únicos por fila de la tabla. Las tablas permiten el almacenamiento de registros en la misma estructura, que contienen los mismos atributos. Los datos estructurados generalmente se almacenan de forma relacional; minimiza los costos de almacenamiento al reducir la duplicación de datos. Los datos estructurados se consultan comúnmente mediante SQL (lenguaje de consulta estructurado), por lo que las bases de datos no relacionales que almacenan datos semiestructurados a veces se denominan NoSQL.

Las bases de datos de valores clave no son relacionales y se pueden utilizar para almacenar datos semiestructurados. Los datos semiestructurados permiten la variación entre cada instancia de una entidad de datos. Por ejemplo, un registro puede contener una sola dirección de correo electrónico como atributo, otro puede contener tres y otro puede no contener ninguna dirección de correo electrónico. Las bases de datos de valores clave permiten almacenar dos componentes de datos relacionados con un registro individual: una clave única y un valor, que pueden contener diferentes tipos, formatos y cantidades de datos por registro.

Las bases de datos de familias de columnas no son relacionales y se pueden usar para almacenar datos semiestructurados. Las tablas en las bases de datos de familias de columnas se pueden dividir en grupos (familias de columnas), que contienen conjuntos de columnas que están relacionadas entre sí.

Las bases de datos de gráficos no son relacionales y se pueden usar para almacenar datos semiestructurados. Las bases de datos de gráficos contienen una dimensión adicional a muchos otros tipos de bases de datos; almacenan entidades como nodos y permiten que las relaciones con otros nodos (bordes) tengan una dirección. Por ejemplo, una relación padre/hijo entre dos nodos de persona indicaría cuál es el padre y cuál es el hijo.

**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)

1. D

Tabla

Descripción generada automáticamente

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.

**Traducción**

Los datos semiestructurados no necesitan adherirse a un esquema fijo. Los datos semiestructurados permiten flexibilidad en los datos que se almacenan. Cada instancia de un elemento de datos (por ejemplo, un cliente) puede tener diferentes atributos, que se pueden almacenar en un orden diferente. Los datos estructurados requieren un esquema fijo; es decir, cada fila de datos debe contener el mismo conjunto de atributos en el mismo orden.

Los datos semiestructurados permiten la variación entre instancias de entidades de datos. Los datos semiestructurados permiten el almacenamiento de diferentes conjuntos de atributos por instancia de una entidad de datos. Por ejemplo, un cliente puede tener dos direcciones de correo electrónico y ningún número de teléfono móvil, y un segundo cliente puede tener tres números de teléfono móvil y una dirección de correo electrónico.

Los datos semiestructurados permiten un esquema jerárquico. Los datos semiestructurados permiten configurar entidades de datos de forma jerárquica; es decir, algunas entidades se pueden modelar en una relación padre/hijo. Por ejemplo, puede representar una relación compleja de gerente y empleado, en la que un empleado puede informar a muchos gerentes, o un gerente puede ser responsable de muchos empleados, compartiendo potencialmente la responsabilidad de algunos empleados con otros gerentes.

Los datos semiestructurados no requieren que todas las entidades de datos tengan los mismos atributos. Esto es cierto para los datos estructurados. Los datos semiestructurados permiten la variación entre instancias de entidades de datos; se pueden registrar diferentes atributos y números del mismo atributo para cada instancia.

**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

1. D

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.

**Traducción**

Una clave principal es el nombre que se le da a una columna específica en una tabla de base de datos que almacena un identificador único para cada fila, que es análogo a una instancia de una entidad de datos.

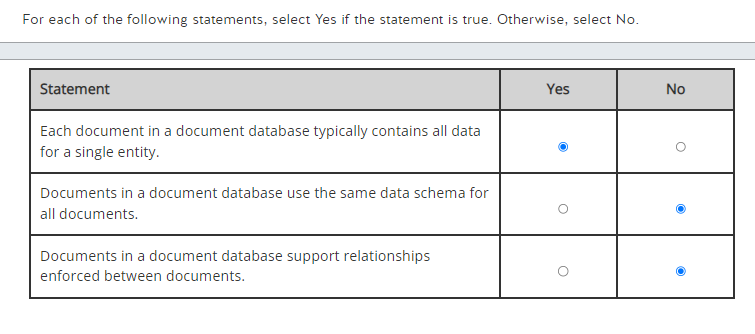
Una clave externa es el nombre dado a cualquier columna en una tabla de base de datos que hace referencia a filas en otra tabla por su propio identificador único. Esto permite la construcción de relaciones entre dos instancias de entidades de datos. Por ejemplo, la tabla del aula tiene una fila con los ID 1 y 2. La tabla de profesores tiene una fila con el ID 3. Ambas filas del aula pueden hacer referencia a la fila 3 de la tabla de profesores para indicar que el mismo profesor está enseñando en ambas aulas.

Las filas son instancias de una entidad de datos; un conjunto de valores para cada columna en la tabla de la base de datos. Las filas contienen un valor de clave principal, que actúa como un identificador único para esa instancia de la entidad de datos.

Se crea un índice a partir de varias columnas para mejorar la velocidad de las consultas. Pueden (y generalmente lo hacen) incluir la clave principal de la tabla.

**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)

1. D



Each document in a document database typically contains all data for a single entity. The data contained in the document can vary between documents. Each document is identified by a unique key used to identify the document and each document is written or retrieved as a single block.  
  
Documents in a document database do not use the same data schema for all documents. The schema is defined internally in the document, and each individual document can have a different schema. This allows for easy support of denormalized data and variations between entities.  
  
Documents in a document database do not support relationships enforced between documents. Document databases do not provide a way to establish relationships between documents.

**Traducción**

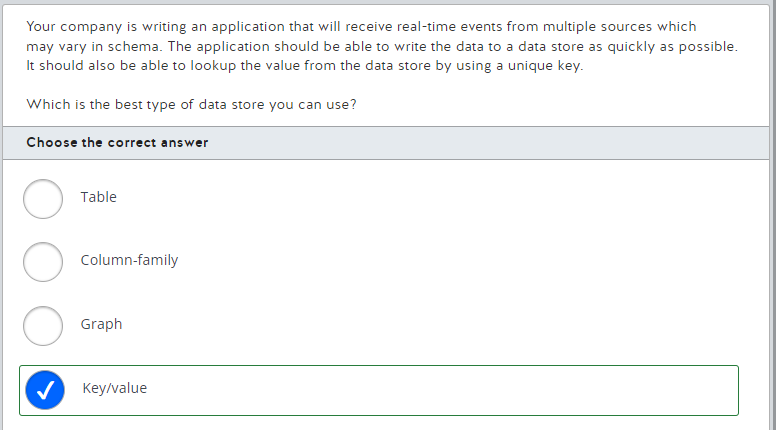
Cada documento en una base de datos de documentos normalmente contiene todos los datos de una sola entidad. Los datos contenidos en el documento pueden variar entre documentos. Cada documento se identifica mediante una clave única utilizada para identificar el documento y cada documento se escribe o recupera como un solo bloque.

Los documentos de una base de datos de documentos no utilizan el mismo esquema de datos para todos los documentos. El esquema se define internamente en el documento y cada documento individual puede tener un esquema diferente. Esto permite un fácil soporte de datos desnormalizados y variaciones entre entidades.

Los documentos de una base de datos de documentos no admiten las relaciones impuestas entre documentos. Las bases de datos de documentos no proporcionan una forma de establecer relaciones entre documentos.

**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)

1. D



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.

**Traducción**

Debe utilizar un almacén de datos clave/valor. Un almacén de datos clave/valor funciona esencialmente como una gran tabla hash y está optimizado para escrituras rápidas de datos. Cada fila de datos está referenciada por un único valor de clave. Las únicas operaciones admitidas son operaciones simples de consulta, inserción y eliminación. Las actualizaciones de datos requieren que la aplicación reescriba los datos para el valor completo. Las consultas se pueden ejecutar por una clave o un rango de claves.

No debe utilizar un almacén de datos de familia de columnas (columnar). Un almacén de datos de familia de columnas es similar a un almacén de datos relacional en el sentido de que los datos se organizan en filas y columnas, pero las columnas se dividen en familias de columnas que pueden almacenar varios valores en una sola columna. Una fila no tiene necesariamente un valor en cada familia de columnas. Las columnas dentro de una familia de columnas se almacenan físicamente en el mismo archivo.

No debe utilizar un almacén de datos de tabla. Un almacén de datos de tabla utiliza un formato de datos de fila y columna con los datos algo normalizados, pero no se aplica el mismo esquema en todas las filas. Cada fila puede tener un número diferente de columnas. En Azure Table Storage, los datos se organizan en función de una clave de partición y una clave de fila. La clave de partición identifica la partición en la que se almacenan los datos y la fila identifica de forma única la fila dentro de la partición.

No debe usar un almacén de datos de gráficos. Un almacén de datos de gráficos está diseñado para admitir relaciones extensas y complejas entre entidades. Esto ayuda a que sea más fácil realizar análisis de relaciones complejas.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use a graph data store, which is made up of entities and relationships that are referred to as nodes and edges. You can have multiple relationships between entities, including hierarchical relationships. A graph data store is designed to support extensive, complex relationships between entities. This helps to make it easier to perform complex relation analysis.  
  
You should not use a document store. A document store supports semi-structured documents. Each document is identified as a single key, and the data schema is defined internally in each document named fields and values. Therefore,  the schema and content can vary between documents. Each document typically contains the data for a single entity. Relationships are not defined between documents.  
  
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. Data is denormalized and relationships are not defined between entities.  
  
You should not use an object store. An object store is used to store unstructured data, such as audio or video files, and it does not provide for relationship analysis.

**Traducción**

Debe usar un almacén de datos de gráficos, que se compone de entidades y relaciones que se denominan nodos y bordes. Puede tener varias relaciones entre entidades, incluidas las relaciones jerárquicas. Un almacén de datos de gráficos está diseñado para admitir relaciones extensas y complejas entre entidades. Esto ayuda a que sea más fácil realizar análisis de relaciones complejas.

No debe utilizar un almacén de documentos. Un almacén de documentos admite documentos semiestructurados. Cada documento se identifica como una sola clave y el esquema de datos se define internamente en cada documento con nombres de campos y valores. Por lo tanto, el esquema y el contenido pueden variar entre documentos. Cada documento normalmente contiene los datos de una sola entidad. Las relaciones no están definidas entre documentos.

No debe utilizar un almacén de datos de familia de columnas (columnar). Un almacén de datos de familia de columnas es similar a un almacén de datos relacional en el sentido de que los datos se organizan en filas y columnas, pero las columnas se dividen en familias de columnas que pueden almacenar varios valores en una sola columna. Una fila no tiene necesariamente un valor en cada familia de columnas. Las columnas dentro de una familia de columnas se almacenan físicamente en el mismo archivo. Los datos se desnormalizan y las relaciones no se definen entre las entidades.

No debe utilizar un almacén de objetos. Un almacén de objetos se utiliza para almacenar datos no estructurados, como archivos de audio o video, y no proporciona análisis de relaciones.

**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)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

For large audio and video files that are used as the source for streaming content, you should choose an object data store. Files of this type are unstructured, non-relational data. The typical storage solution for this type of file is an object store, such as Azure Blob storage.  
  
Semi-structured data with each entity providing its own field definitions is a description of document-type data, and therefore a document store is your best choice. Documents are written and retrieved as complete documents. The embedded field definitions make it possible to query documents in order to retrieve field values. You would typically use an Azure Cosmos DB storage solution.  
  
For entities and relationships, including multiple and complex relationships with other entities, you should use a graph data store. A graph data store is made up of entities and relationships that are referred to as nodes and edges. You can have multiple relationships between entities, including hierarchical relationships. A graph data store is designed to support extensive, complex relationships between entities. This helps to make it easier to perform complex relation analysis.  
  
None of the descriptions refer to data that should be stored in a key/value data store. A key/value 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 key or a range of keys.

**Traducción**

Para archivos de audio y video de gran tamaño que se utilizan como fuente para la transmisión de contenido, debe elegir un almacén de datos de objetos. Los archivos de este tipo son datos no estructurados y no relacionales. La solución de almacenamiento típica para este tipo de archivo es un almacén de objetos, como Azure Blob Storage.

Los datos semiestructurados en los que cada entidad proporciona sus propias definiciones de campo son una descripción de los datos de tipo de documento y, por lo tanto, un almacén de documentos es su mejor opción. Los documentos se escriben y recuperan como documentos completos. Las definiciones de campo incrustadas hacen posible consultar documentos para recuperar valores de campo. Por lo general, usaría una solución de almacenamiento de Azure Cosmos DB.

Para las entidades y las relaciones, incluidas las relaciones múltiples y complejas con otras entidades, debe utilizar un almacén de datos de gráfico. Un almacén de datos de gráficos se compone de entidades y relaciones que se denominan nodos y bordes. Puede tener varias relaciones entre entidades, incluidas las relaciones jerárquicas. Un almacén de datos de gráficos está diseñado para admitir relaciones extensas y complejas entre entidades. Esto ayuda a que sea más fácil realizar análisis de relaciones complejas.

Ninguna de las descripciones hace referencia a los datos que deben almacenarse en un almacén de datos de clave/valor. Un almacén de clave/valor funciona esencialmente como una gran tabla hash y está optimizado para escrituras rápidas de datos. Cada fila de datos está referenciada por un único valor de clave. Las únicas operaciones admitidas son operaciones simples de consulta, inserción y eliminación. Las actualizaciones de datos requieren que la aplicación reescriba los datos para el valor completo. Las consultas se pueden ejecutar por clave o un rango de claves.

**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 core data concepts](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-identify-need-data-solutions)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should recommend an object data store. Object data stores are optimized for storing and retrieving large binary objects or blobs such as text files, images, and videos. An example of service capable of storing this kind of information is Azure Blob Storage.  
  
You should not recommend a document data store. A document data store is used to store semi-structured data in fields and values with a flexible structure. Documents are usually represented in JavaScript Object Notation (JSON) format and can have different fields to represent the same class of information.  
  
You should not recommend a key/value data store. A key/value data store is used to store values associated with a unique key. This kind of data store is optimized for performing simple lookups using the value of the key.  
  
You should not recommend a graph data store. A graph data store manages nodes and edges. Nodes represent entities, and edges specify the relationships between these entities.

**Traducción**

Debe recomendar un almacén de datos de objetos. Los almacenes de datos de objetos están optimizados para almacenar y recuperar grandes objetos binarios o blobs, como archivos de texto, imágenes y videos. Un ejemplo de servicio capaz de almacenar este tipo de información es Azure Blob Storage.

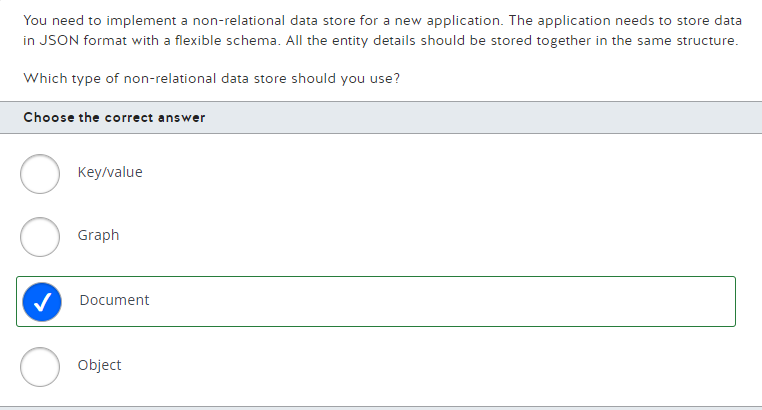
No debe recomendar un almacén de datos de documentos. Un almacén de datos de documentos se utiliza para almacenar datos semiestructurados en campos y valores con una estructura flexible. Los documentos generalmente se representan en formato de notación de objetos JavaScript (JSON) y pueden tener diferentes campos para representar la misma clase de información.

No debe recomendar un almacén de datos de clave/valor. Un almacén de datos de clave/valor se utiliza para almacenar valores asociados con una clave única. Este tipo de almacén de datos está optimizado para realizar búsquedas simples utilizando el valor de la clave.

No debe recomendar un almacén de datos de gráficos. Un almacén de datos de gráficos administra nodos y bordes. Los nodos representan entidades y los bordes especifican las relaciones entre estas entidades.

**References**[Non-relational data and NoSQL](https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data)  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)

1. D



You should use a document data store. A document data store contains the entire data for an entity stored in single documents with a unique id. This document supports a flexible schema and the entity data is usually stored in JSON format.  
  
You should not use a key/value data store. 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.  
  
You should not use a graph data store. A graph data store consists of graphs containing edges and nodes and is used to store and query complex relationships among entities.  
  
You should not use an object data store. An object data store is used to store large binary objects, such as images and media files.

**Traducción**

Debe utilizar un almacén de datos de documentos. Un almacén de datos de documentos contiene todos los datos de una entidad almacenada en documentos individuales con una identificación única. Este documento admite un esquema flexible y los datos de la entidad generalmente se almacenan en formato JSON.

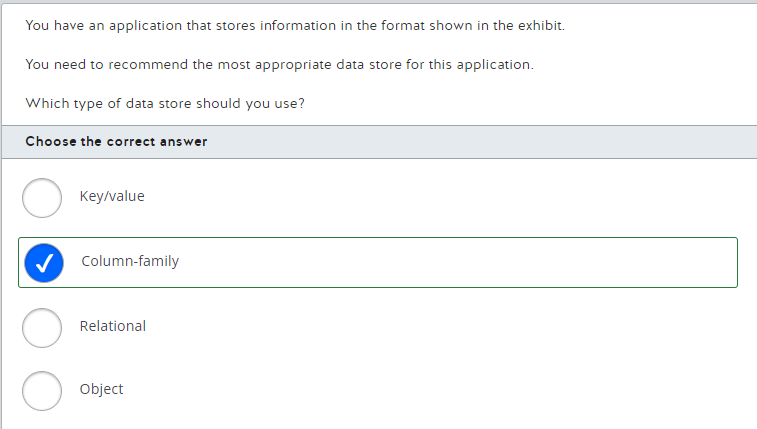
No debe utilizar un almacén de datos de clave/valor. Un almacén de datos clave/valor consiste en una estructura de datos simple y rápida donde puede almacenar información en un valor que se identifica mediante una clave.

No debe usar un almacén de datos de gráficos. Un almacén de datos de gráficos consta de gráficos que contienen bordes y nodos y se utiliza para almacenar y consultar relaciones complejas entre entidades.

No debe utilizar un almacén de datos de objetos. Un almacén de datos de objetos se utiliza para almacenar objetos binarios de gran tamaño, como imágenes y archivos multimedia.

**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)

1. D



You should use a column-family data store. A column-family data store consists of row identifiers and a group of information stored in a column. Each group of information is stored in independent columns, just like the customers' identities are shown in the exhibit.  
  
You should not use a relational data store. A relational data store consists of rows and columns defined in tables, and each table stores a specific set of data in a normalized format. In this scenario, the data is stored in a single column, in which each row contains denormalized data.  
  
You should not use an object data store. An object data store consists of large binary objects, such as images, media files, and other types of files.  
  
You should not use a key/value data store. 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.

**Traducción**

Debe utilizar un almacén de datos de familias de columnas. Un almacén de datos de familia de columnas consta de identificadores de fila y un grupo de información almacenada en una columna. Cada grupo de información se almacena en columnas independientes, al igual que las identidades de los clientes que se muestran en la exposición.

No debe utilizar un almacén de datos relacional. Un almacén de datos relacionales consta de filas y columnas definidas en tablas, y cada tabla almacena un conjunto específico de datos en un formato normalizado. En este escenario, los datos se almacenan en una sola columna, en la que cada fila contiene datos desnormalizados.

No debe utilizar un almacén de datos de objetos. Un almacén de datos de objetos consta de grandes objetos binarios, como imágenes, archivos multimedia y otros tipos de archivos.

No debe utilizar un almacén de datos de clave/valor. Un almacén de datos clave/valor consiste en una estructura de datos simple y rápida donde puede almacenar información en un valor que se identifica mediante una clave.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

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.

**Traducción**

Un almacén de datos de gráficos consta de bordes y nodos que se utilizan para almacenar y consultar relaciones complejas entre entidades. Almacena todas las entidades como nodos y crea la relación entre esas entidades mediante el uso de bordes.

Un almacén de datos clave/valor consiste en una estructura de datos simple y rápida donde puede almacenar información en un valor que se identifica mediante una clave.

Un almacén de datos en columnas consta de identificadores de fila y un grupo de información almacenada en una columna. Cada grupo de información se almacena en columnas independientes.

Un almacén de datos de objetos consta de grandes objetos binarios, como imágenes, archivos multimedia y otros tipos de archivos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Document databases and graph databases are examples of non-relational data stores. Document databases store data in JSON or XML format and do not require all documents to have the same structure. Graph databases store information in the form of edges and nodes. They are used to represent complex relationships such as social interactivity.  
  
The Azure Database for MariaDB and SQL databases are examples of relational databases. Relational databases store information in the form of tables, which you can connect through relationships. Relational databases are used for highly structured data.

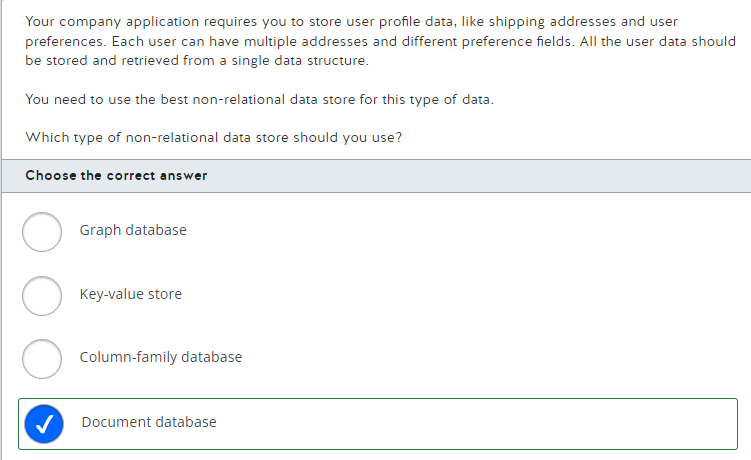
**Traducción**

Las bases de datos de documentos y las bases de datos de gráficos son ejemplos de almacenes de datos no relacionales. Las bases de datos de documentos almacenan datos en formato JSON o XML y no requieren que todos los documentos tengan la misma estructura. Las bases de datos de gráficos almacenan información en forma de aristas y nodos. Se utilizan para representar relaciones complejas como la interactividad social.

Azure Database for MariaDB y las bases de datos SQL son ejemplos de bases de datos relacionales. Las bases de datos relacionales almacenan información en forma de tablas, que puede conectar a través de relaciones. Las bases de datos relacionales se utilizan para datos altamente estructurados.

**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)  
  
[Relational vs. NoSQL data](https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data)  
  
[What is Azure Database for MariaDB?](https://docs.microsoft.com/en-us/azure/mariadb/overview)

1. D



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.

**Traducción**

Debe utilizar una base de datos de documentos. Un almacén de datos de documentos consta de entidades que tienen sus datos relacionados almacenados en un solo documento. Este documento admite un esquema flexible y los datos de la entidad generalmente se almacenan en formato JSON. Se pueden utilizar para mantener la información del perfil del usuario, lo que permite almacenar toda la información en un solo bloque.

No debe utilizar un almacén de datos no relacional de clave-valor. Los almacenes de clave-valor están altamente optimizados para estructuras de datos simples. Un almacén de clave-valor asocia cada valor de datos con una clave que se puede usar para acceder a los datos.

No debe utilizar una base de datos de familias de columnas. Las bases de datos de familias de columnas almacenan datos en filas y columnas, agrupando columnas relacionadas en familias de columnas que se usan juntas. En lugar de una base de datos basada en documentos, puede usar una base de datos de familias de columnas para recuperar solo las familias de columnas requeridas en lugar de todos los datos del usuario.

No debe utilizar una base de datos de gráficos. Las bases de datos de gráficos almacenan información en forma de aristas y nodos. Se utilizan para representar relaciones complejas como la interactividad social.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

The file format shown in the exhibit is an extensible markup language (XML). XML uses tags (elements) for entities within angle brackets, much like HTML. Attributes are added inside tags in the following format, separated by spaces. XML is a traditional format for web applications, but many newer applications now use JSON instead. The following is an example of XML:  
  
<attribute name="attribute value">  
  
JavaScript object notation (JSON) encloses entities in parentheses, {}; collections in square brackets, []; attributes inside entities in the below format, separated by commas. JSON is a newer way of storing semi-structured data than XML, and is commonly used by web applications.  
  
{"attribute name":"attribute value"}  
  
Parquet contains row groups, each of which stores chunks of data. It stores data in columns rather than rows. Parquet is a file format developed by Apache; it is useful for efficient querying of very complex data, such as that generated by social networking.  
  
Comma-separated values (CSV) files are simple lists of attributes separated by a delimiter, which is often a comma. Sometimes, the names of the columns will be added in the first row. CSV is a very simple and ubiquitous structure that can be easily interpreted by older applications. CSV data would look like the example below.  
  
attribute 1 name, attribute 2 name  
attribute 1 value, attribute 2 value

**Traducción**

El formato de archivo que se muestra en la exposición es un lenguaje de marcado extensible (XML). XML usa etiquetas (elementos) para entidades dentro de paréntesis angulares, al igual que HTML. Los atributos se agregan dentro de las etiquetas en el siguiente formato, separados por espacios. XML es un formato tradicional para aplicaciones web, pero muchas aplicaciones más nuevas ahora usan JSON en su lugar. El siguiente es un ejemplo de XML:

<atributo nombre="valor de atributo">

La notación de objetos de JavaScript (JSON) encierra entidades entre paréntesis, {}; colecciones entre corchetes, []; atributos dentro de entidades en el siguiente formato, separados por comas. JSON es una forma más nueva de almacenar datos semiestructurados que XML, y las aplicaciones web lo usan comúnmente.

{"nombre del atributo":"valor del atributo"}

Parquet contiene grupos de filas, cada uno de los cuales almacena fragmentos de datos. Almacena datos en columnas en lugar de filas. Parquet es un formato de archivo desarrollado por Apache; es útil para consultas eficientes de datos muy complejos, como los generados por las redes sociales.

Los archivos de valores separados por comas (CSV) son listas simples de atributos separados por un delimitador, que suele ser una coma. A veces, los nombres de las columnas se agregarán en la primera fila. CSV es una estructura muy simple y omnipresente que las aplicaciones más antiguas pueden interpretar fácilmente. Los datos CSV se verían como el siguiente ejemplo.

atributo 1 nombre, atributo 2 nombre

atributo 1 valor, atributo 2 valor

**References**[What is Parquet?](https://databricks.com/glossary/what-is-parquet)  
  
[Explore file storage](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/3-file-storage)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

Graph databases store semi-structured data.  
  
Nodes represent instances of data entities, like individual people. Nodes are analogous to rows in a table within a relational database.  
  
Edges represent the relationships between nodes; they are also called graphs or relationships. Edges can be directed or undirected, depending on the structure of the individual database.  
  
Properties represent attributes of data against a node. For instance, a person might have an email address or phone number. Properties are analogous to columns on a table in a relational database. However, remember that NoSQL databases allow for flexibility on the attributes stored against a node. For example, a node may have many instances of a single attribute or it may be absent.  
  
Directions are a property of an edge. Edges can be directed or undirected. Directed edges store two pieces of information relating to each of the nodes they connect. For instance, a parent/child directed edge would store which person node represents the parent and which the child. Undirected edges link nodes together, where the direction of the relationship does not matter; for example, a friendship between two people.

**Traducción**

Las bases de datos de gráficos almacenan datos semiestructurados.

Los nodos representan instancias de entidades de datos, como personas individuales. Los nodos son análogos a las filas de una tabla dentro de una base de datos relacional.

Los bordes representan las relaciones entre los nodos; también se les llama gráficos o relaciones. Los bordes pueden ser dirigidos o no, según la estructura de la base de datos individual.

Las propiedades representan atributos de datos contra un nodo. Por ejemplo, una persona puede tener una dirección de correo electrónico o un número de teléfono. Las propiedades son análogas a las columnas de una tabla en una base de datos relacional. Sin embargo, recuerde que las bases de datos NoSQL permiten flexibilidad en los atributos almacenados en un nodo. Por ejemplo, un nodo puede tener muchas instancias de un solo atributo o puede estar ausente.

Las direcciones son una propiedad de una arista. Los bordes pueden ser dirigidos o no dirigidos. Los bordes dirigidos almacenan dos piezas de información relacionadas con cada uno de los nodos que conectan. Por ejemplo, un borde dirigido a padre/hijo almacenaría qué nodo de persona representa al padre y cuál al hijo. Los bordes no dirigidos unen los nodos, donde la dirección de la relación no importa; por ejemplo, una amistad entre dos personas.

**References**  
  
[Graph database](https://en.wikipedia.org/wiki/Graph_database)  
  
[Explore databases](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-databases)

1. D



Relational databases store structured data.  
  
A row represents an instance of a data entity, like an individual person or company. You can think of a row like a line in a spreadsheet.  
  
A key represents a relationship between two individual data entities. Each table has a primary key column, which gives each row a unique value (usually numerical) that acts as an identifier that can be referenced using a foreign key column on another table.  
  
A column represents a data attribute on a table. An example might be an e-mail address column in the table Person. Each row (individual person record) may have an e-mail address value in that column.  
  
A table represents a data entity itself, like a person. A table has columns (attributes) and rows (instances of the entity).

**Traducción**

Las bases de datos relacionales almacenan datos estructurados.

Una fila representa una instancia de una entidad de datos, como una persona individual o una empresa. Puede pensar en una fila como una línea en una hoja de cálculo.

Una clave representa una relación entre dos entidades de datos individuales. Cada tabla tiene una columna de clave principal, que otorga a cada fila un valor único (generalmente numérico) que actúa como un identificador al que se puede hacer referencia mediante una columna de clave externa en otra tabla.

Una columna representa un atributo de datos en una tabla. Un ejemplo podría ser una columna de dirección de correo electrónico en la tabla Persona. Cada fila (registro de persona individual) puede tener un valor de dirección de correo electrónico en esa columna.

Una tabla representa una entidad de datos en sí misma, como una persona. Una tabla tiene columnas (atributos) y filas (instancias de la entidad).

**References**  
  
[Understand relational data](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/2-understand-relational-data)

1. D

Interfaz de usuario gráfica, Aplicación

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.

**Traducción**

Los archivos de valores separados por comas (CSV) están estructurados. Almacenan texto sin formato separado por delimitadores de campo, a menudo comas, creando un esquema establecido. Cada fila de datos debe tener el mismo conjunto de atributos en el mismo orden. Puede guardar archivos de datos estructurados, como hojas de cálculo de Excel, como archivos .csv para proporcionar un formato flexible que pueden usar muchas aplicaciones.

Los archivos de lenguaje de marcado extensible (XML) están semiestructurados. Almacenan elementos y atributos en etiquetas en un esquema jerárquico, que parecen etiquetas HTML, p. <cliente />, pero se pueden definir según los requisitos de almacenamiento de datos. XML es una buena opción para almacenar datos semiestructurados, ya que los elementos y las etiquetas son flexibles.

Los archivos Binary Large Object (BLOB) almacenan datos binarios sin procesar sin un esquema. Los archivos BLOB se utilizan para datos no estructurados, como audio y video.

Los archivos de notación de objetos de JavaScript (JSON) almacenan entidades de datos en un esquema jerárquico, muy parecido a XML. Cada entidad puede tener un conjunto o número de atributos diferente, lo que la convierte en una buena opción para almacenar datos semiestructurados.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

La clave principal está resaltada. La columna de clave principal contiene valores que son identificadores únicos para una fila en particular. En este escenario, la columna ID almacena un número único para cada cliente.

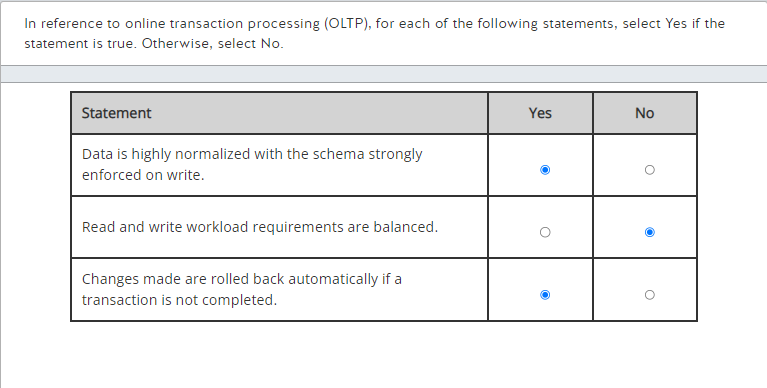
Las claves foráneas no están resaltadas. Las columnas de clave externa contienen valores que hacen referencia a la clave principal de otra tabla. La columna ID de ciudad es una columna de clave externa; hace referencia a los valores de clave principal de la tabla Customer, normalizando los datos. La normalización es el proceso de eliminación de datos duplicados. Hacer referencia a una fila en la tabla Ciudad significa que no tenemos que almacenar columnas adicionales en la tabla Cliente para cada fila en la tabla Ciudad.

Una instancia de una entidad de cliente no está resaltada. Cada fila de datos que se muestra en la exposición representa un cliente diferente, una instancia de la entidad de datos del cliente.

Un índice no agrupado no está resaltado. Un índice no agrupado es una estructura de una tabla de base de datos que funciona como una página de índice en un libro de texto. Se utiliza para acelerar las consultas.

**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)

1. D



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.

**Traducción**

Los sistemas OLTP se utilizan para registrar las actividades comerciales y las interacciones diarias a medida que ocurren. Esto incluye actividades tales como pedidos recibidos, servicios prestados y pagos recibidos o realizados.

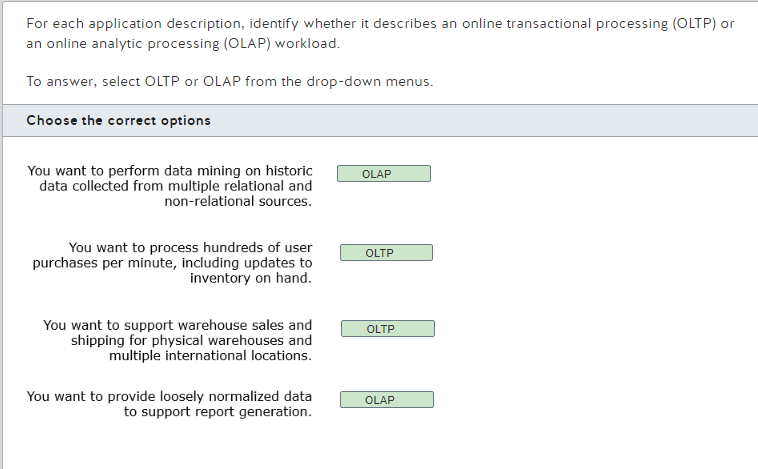
En un sistema OLTP, los datos están altamente normalizados y el esquema se aplica fuertemente en la escritura. Los sistemas OLTP generalmente se estructuran en torno a un almacén de datos relacionales que admite aplicaciones transaccionales.

Una carga de trabajo OLTP tiene grandes requisitos de escritura con requisitos de lectura mínimos (en comparación).

En un entorno OLTP, los cambios realizados se revierten automáticamente si una transacción no se completa, de modo que ninguna transacción quede en un estado parcialmente completado. Esto se conoce como atomicidad y es un requisito para 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)

1. D



An application to perform data mining on historic data collected from multiple relational and non-relational sources is an example of an OLAP workload. OLAP applications often manipulate data based on complex queries. Data mining queries are complex multidimensional queries that are designed to discover insights from the data that are not immediately apparent.  
  
An application to process hundreds of user purchases per minute, including updates to inventory on hand, is an example of an OLTP workload. OLTP applications are optimized for write operations and entering and updating data across multiple, related tables. Partial changes made to data are rolled back automatically if a transaction is not completed so that no transaction is left in a partially completed state.  
  
An application to support warehouse sales and shipping for physical warehouses and multiple international locations is an example of an OLTP application. OLTP transactions can be distributed geographically and supported by one or more relational databases. This scenario requires a solution that supports consistent and reliable data writes.  
  
An application to provide loosely normalized data to support report generation is an example of an OLAP workload. Companies will often maintain live data for transactional processing and a separate copy of historic data for analysis and report generation. This prevents analytic processing from interfering with the performance during transactional processing.

**Traducción**

Una aplicación para realizar minería de datos en datos históricos recopilados de múltiples fuentes relacionales y no relacionales es un ejemplo de una carga de trabajo OLAP. Las aplicaciones OLAP a menudo manipulan datos en función de consultas complejas. Las consultas de minería de datos son consultas multidimensionales complejas que están diseñadas para descubrir información de los datos que no son evidentes de inmediato.

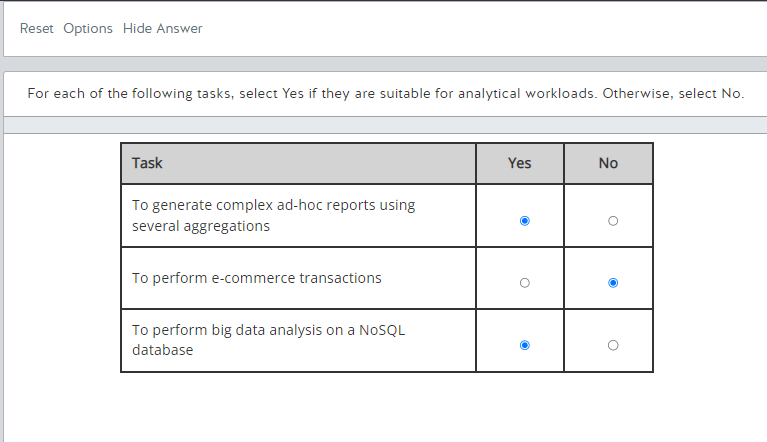
Una aplicación para procesar cientos de compras de usuarios por minuto, incluidas las actualizaciones del inventario disponible, es un ejemplo de una carga de trabajo de OLTP. Las aplicaciones OLTP están optimizadas para operaciones de escritura e ingreso y actualización de datos en varias tablas relacionadas. Los cambios parciales realizados en los datos se revierten automáticamente si una transacción no se completa para que ninguna transacción quede en un estado parcialmente completado.

Una aplicación para admitir ventas y envíos de almacén para almacenes físicos y varias ubicaciones internacionales es un ejemplo de una aplicación OLTP. Las transacciones OLTP pueden distribuirse geográficamente y respaldarse en una o más bases de datos relacionales. Este escenario requiere una solución que admita escrituras de datos consistentes y confiables.

Una aplicación para proporcionar datos poco normalizados para respaldar la generación de informes es un ejemplo de una carga de trabajo OLAP. Las empresas a menudo mantendrán datos en vivo para el procesamiento transaccional y una copia separada de los datos históricos para el análisis y la generación de informes. Esto evita que el procesamiento analítico interfiera con el rendimiento durante el procesamiento transaccional.

**References**  
  
[Explore transactional data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/5-transactional-data-processing)  
  
[Explore analytical data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/6-analytical-processing)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)  
  
[Online analytical processing (OLAP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing)

1. D



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

**Traducción**

Los sistemas de procesamiento analítico en línea (OLAP) están diseñados para realizar análisis complejos y proporcionar inteligencia comercial. OLAP se utiliza para cargas de trabajo analíticas, como:

• Generación de informes ad-hoc complejos que incluyen varias agregaciones

• Realizar análisis de big data en una base de datos NoSQL

Los sistemas de procesamiento de transacciones en línea (OLTP) están diseñados para realizar transacciones comerciales a medida que ocurren. OLTP se usa para cargas de trabajo transaccionales, como:

• Realización de transacciones de comercio electrónico

• Seguimiento de los sistemas de gestión de inventario

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use an OLAP workload. You can use OLAP workloads to organize large business databases and perform complex analytics, like data mining, without negatively affecting transactional systems.  
  
You should not use 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.  
  
You should not use a stream processing workload. You can use streaming processing workloads to handle a continuous stream of data used by time-critical operations.  
  
You should not use a batch processing workload. You can use batch processing workloads to collect a group of data with a scheduled time interval or when a certain amount of data has arrived that is not time-sensitive.

**Traducción**

Debe usar una carga de trabajo OLAP. Puede usar cargas de trabajo OLAP para organizar grandes bases de datos comerciales y realizar análisis complejos, como la extracción de datos, sin afectar negativamente los sistemas transaccionales.

No debe usar una carga de trabajo OLTP. Puede usar cargas de trabajo OLTP con sistemas transaccionales utilizados en las operaciones diarias de una organización, como sistemas contables, financieros y de otro tipo, que requieren una gran coherencia para las transacciones.

No debe utilizar una carga de trabajo de procesamiento de secuencias. Puede usar cargas de trabajo de procesamiento de transmisión para manejar un flujo continuo de datos utilizados por operaciones de tiempo crítico.

No debe usar una carga de trabajo de procesamiento por lotes. Puede usar cargas de trabajo de procesamiento por lotes para recopilar un grupo de datos con un intervalo de tiempo programado o cuando llega una cierta cantidad de datos que no son sensibles al tiempo.

**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)  
  
[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)  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

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.

**Traducción**

Este es un ejemplo de una carga de trabajo de OLTP. Puede usar cargas de trabajo OLTP con sistemas transaccionales utilizados en las operaciones diarias de una organización, como sistemas contables, financieros y de otro tipo, que requieren una gran coherencia para las transacciones.

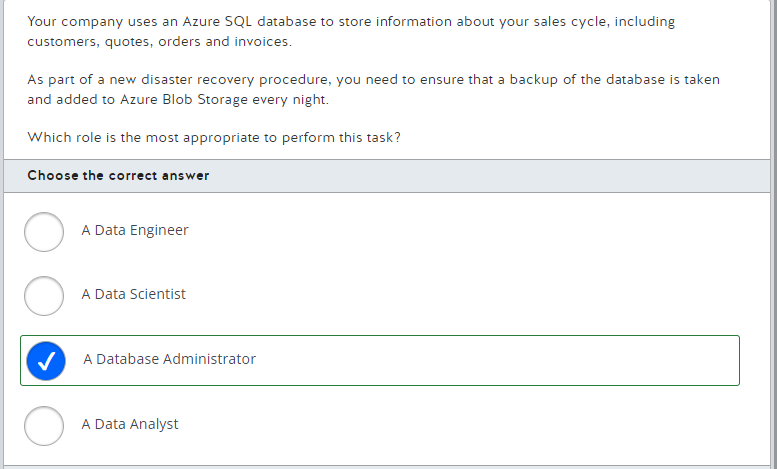
Las cargas de trabajo OLAP organizan grandes bases de datos empresariales y realizan análisis complejos, como la extracción de datos, sin afectar negativamente a los sistemas transaccionales.

Un procesamiento de flujo maneja un flujo continuo de datos utilizados por operaciones de tiempo crítico.

El procesamiento por lotes recopila un grupo de datos dentro de un intervalo de tiempo programado o cuando llega una cierta cantidad de datos que no son sensibles al tiempo.

**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)

1. D



A Database Administrator is the most appropriate role to perform this task. Database Administrators (DBAs) are responsible for maintaining and designing databases; they do the day-to-day work of keeping the company's data-related systems up and running. They will work with business stakeholders to perform tasks like updating the database structure to reflect new business requirements (where needed), ensuring an acceptable database performance and creating backups and copies of the database.  
  
A Data Analyst is not the most appropriate role to perform this task. Data Analysts are responsible for interpreting the data generated by the business, using it to create visual reports or identify trends that can provide business insights. They do not deal with the mechanics of how the data is recorded; they only look at the data itself.  
  
A Data Engineer is not the most appropriate role to perform this task. Data Engineers are responsible for manipulating data (performing cleansing or enrichment routines) and moving it where it needs to go (via pipelines or other integrations). Although they often have the same skillsets as Data Scientists, this is a more practical, hands-on job; they make day-to-day systems by connecting individual data sources together.  
  
A Data Scientist is not the most appropriate role to perform this task. Data Scientists are responsible for working with data on a mathematical level. They design and implement processes like machine learning to gain insights from existing data. Their job is to ask questions about a data set and use theories and experiments to find out the answers.

**Traducción**

Un administrador de base de datos es el rol más apropiado para realizar esta tarea. Los administradores de bases de datos (DBA) son responsables de mantener y diseñar bases de datos; hacen el trabajo diario de mantener en funcionamiento los sistemas relacionados con los datos de la empresa. Trabajarán con las partes interesadas del negocio para realizar tareas como actualizar la estructura de la base de datos para reflejar los nuevos requisitos comerciales (cuando sea necesario), garantizar un rendimiento aceptable de la base de datos y crear copias de seguridad y copias de la base de datos.

Un analista de datos no es el rol más apropiado para realizar esta tarea. Los analistas de datos son responsables de interpretar los datos generados por el negocio, usándolos para crear informes visuales o identificar tendencias que pueden proporcionar información comercial. No se ocupan de la mecánica de cómo se registran los datos; solo miran los datos en sí.

Un ingeniero de datos no es el rol más apropiado para realizar esta tarea. Los ingenieros de datos son responsables de manipular los datos (realizando rutinas de limpieza o enriquecimiento) y moverlos a donde deben ir (a través de canalizaciones u otras integraciones). Aunque a menudo tienen las mismas habilidades que los científicos de datos, este es un trabajo más práctico; crean sistemas cotidianos conectando fuentes de datos individuales.

Un científico de datos no es el rol más apropiado para realizar esta tarea. Los científicos de datos son responsables de trabajar con datos a nivel matemático. Diseñan e implementan procesos como el aprendizaje automático para obtener información de los datos existentes. Su trabajo es hacer preguntas sobre un conjunto de datos y usar teorías y experimentos para encontrar las respuestas.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

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.

**Traducción**

El nuevo analista de datos deberá crear informes y paneles. Este será el enfoque principal del trabajo de un analista de datos. Usan herramientas como Power BI y Tableau para contar una historia con datos, creando visualizaciones que ilustran las tendencias y las relaciones que han descubierto para los usuarios comerciales.

El analista de datos identificará tendencias y relaciones en los datos. Es parte del trabajo de un analista de datos usar sus habilidades para descubrir información en los datos que son útiles para el negocio.

El analista de datos también creará y mantendrá modelos de datos y conjuntos de datos. Al crear informes, los analistas de datos incorporarán datos sin procesar y luego los moldearán en un modelo diferente, que proporcionará la base correcta y el rendimiento óptimo para el informe particular que están creando. A menudo crearán y mantendrán conjuntos de datos seleccionados que son útiles para que las personas informen, que pueden pertenecer a modelos o estar en un formato independiente.

El analista de datos no necesitará diseñar e implementar planes de recuperación ante desastres. Esta tarea implica trabajar con los datos en el origen (por ejemplo, dentro de una base de datos de Azure SQL), que no forma parte del trabajo de un analista de datos. Esta tarea es parte del trabajo de un administrador de base de datos.

El trabajo del analista de datos no incluirá el diseño y mantenimiento de un almacén de datos. Un almacén de datos es un almacén central de datos, que ingiere datos de muchas fuentes y los almacena de una manera que cumple con los requisitos de la empresa para la transformación o el análisis de datos. Esta tarea formaría parte del trabajo de un Ingeniero de Datos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Power BI is a data analytics tool that allows users to build data models, reports and dashboards to visualize and interpret data. As such, it would typically be used by a Data Analyst.  
  
SQL Server Management Studio is a tool that allows users to design and maintain SQL Server databases and related components. A Database Administrator would typically use it.  
  
Azure Databricks is an analytics platform used by Data Engineers and Data Scientists to ingest and transform data and develop machine learning models.  
  
Azure Data Studio is a cloud-based cross-platform database tool used by Database Administrators. Similar to SQL Server Management Studio, it allows them to build complex queries for raw data with intelligence to assist them, and create reports and dashboards to check the performance and status of databases they manage.

**Traducción**

Power BI es una herramienta de análisis de datos que permite a los usuarios crear modelos de datos, informes y paneles para visualizar e interpretar datos. Como tal, normalmente lo usaría un analista de datos.

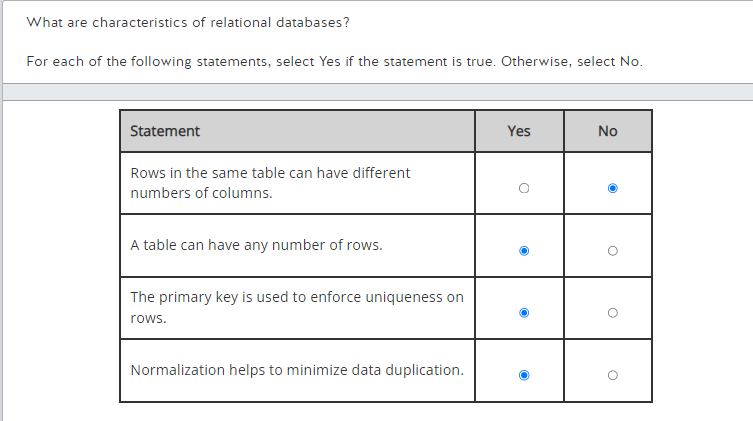
SQL Server Management Studio es una herramienta que permite a los usuarios diseñar y mantener bases de datos de SQL Server y componentes relacionados. Un administrador de base de datos normalmente lo usaría.

Azure Databricks es una plataforma de análisis utilizada por ingenieros y científicos de datos para ingerir y transformar datos y desarrollar modelos de aprendizaje automático.

Azure Data Studio es una herramienta de base de datos multiplataforma basada en la nube que utilizan los administradores de bases de datos. Similar a SQL Server Management Studio, les permite crear consultas complejas para datos sin procesar con inteligencia para ayudarlos, y crear informes y paneles para verificar el rendimiento y el estado de las bases de datos que administran.

**References**  
  
[What is Power BI?](https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview)  
  
[What is Azure Databricks?](https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks)  
  
[What is SQL Server Management Studio (SSMS)?](https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15)  
  
[What is Azure Data Studio?](https://docs.microsoft.com/en-us/sql/azure-data-studio/what-is-azure-data-studio?view=sql-server-ver15)

1. D



Relational data is stored in multiple tables. Each table is made up of rows and columns. Each row represents an instance of an entity for which you want to store information. The columns contain information about the entities, with a given column value for each entity.  
  
The tabular structure of relational databases means that each of the rows in a table will have the same number of columns. Not all columns will necessarily contain a value. Data tables let you have any number of rows.  
  
The primary key is a unique value assigned to a row. Even if all other columns' information is the same in two rows, you must have a column or a group of columns to store the primary key value that makes each row unique. When setting up relationships between tables through foreign keys, each foreign key value must have a corresponding value in a primary key.  
  
Normalization is the process used to split an entity into multiple tables. This helps to minimize data duplication through the use of related tables. For example, an online order might need to include customer information and information about the items ordered. Rather than putting all of this information in order, you can have foreign keys pointing to the detailed customer and item information in other tables.

**Traducción**

Los datos relacionales se almacenan en varias tablas. Cada tabla se compone de filas y columnas. Cada fila representa una instancia de una entidad para la que desea almacenar información. Las columnas contienen información sobre las entidades, con un valor de columna determinado para cada entidad.

La estructura tabular de las bases de datos relacionales significa que cada una de las filas de una tabla tendrá el mismo número de columnas. No todas las columnas contendrán necesariamente un valor. Las tablas de datos le permiten tener cualquier número de filas.

La clave principal es un valor único asignado a una fila. Incluso si la información de todas las demás columnas es la misma en dos filas, debe tener una columna o un grupo de columnas para almacenar el valor de la clave principal que hace que cada fila sea única. Al establecer relaciones entre tablas a través de claves foráneas, cada valor de clave foránea debe tener un valor correspondiente en una clave principal.

La normalización es el proceso utilizado para dividir una entidad en varias tablas. Esto ayuda a minimizar la duplicación de datos mediante el uso de tablas relacionadas. Por ejemplo, es posible que un pedido en línea deba incluir información del cliente e información sobre los artículos pedidos. En lugar de poner toda esta información en orden, puede tener claves externas que apunten a la información detallada del cliente y del artículo en otras tablas.

**References**  
  
[Explore the characteristics of relational data](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/2-explore-characteristics)  
  
[Relational vs. NoSQL data](https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data)  
  
[Relational Data Model](https://binaryterms.com/relational-data-model.html)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

In a relational database, you can use normalization to eliminate data repetition and inconsistent dependencies by separating data into multiple tables and relating these tables by using a foreign key.  
  
A primary key is a constraint that is used to enforce data integrity in relational databases by indicating which column (or combination of columns) uniquely identifies each row in a table. It must have a unique value across a table. A primary key is also used to create relationships between different tables.  
  
A clustered index is a data structure associated with a table that defines the order in which rows are stored on a disk.

**Traducción**

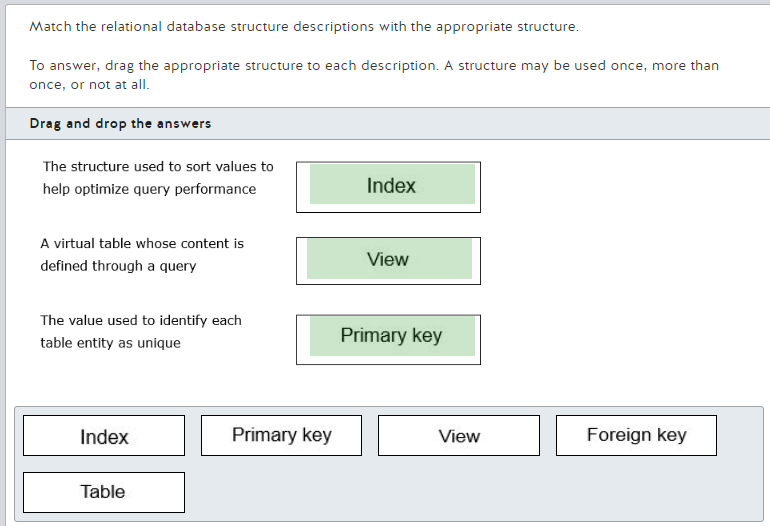
En una base de datos relacional, puede utilizar la normalización para eliminar la repetición de datos y las dependencias incoherentes separando los datos en varias tablas y relacionando estas tablas mediante una clave externa.

Una clave principal es una restricción que se utiliza para imponer la integridad de los datos en las bases de datos relacionales al indicar qué columna (o combinación de columnas) identifica de forma única cada fila de una tabla. Debe tener un valor único en una tabla. Una clave primaria también se usa para crear relaciones entre diferentes tablas.

Un índice agrupado es una estructura de datos asociada a una tabla que define el orden en que se almacenan las filas en un disco.

**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Explore databases](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-databases)  
  
[Description of the database normalization basics](https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description)  
  
[Clustered and nonclustered indexes described](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15)

1. D



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.

**Traducción**

Un índice es la estructura utilizada para ordenar los valores para ayudar a optimizar el rendimiento de las consultas. La mayoría de los sistemas de administración de bases de datos relacionales (RDMS) admiten índices agrupados y no agrupados. La diferencia es que una tabla se ordena físicamente en orden de índice agrupado cuando se usa y un índice no agrupado no cambia el orden de la tabla, pero contiene punteros a las filas de tabla adecuadas.

Una vista es una tabla virtual cuyo contenido se define a través de una consulta. Una vista simple podría ser similar a la siguiente:

CREAR VISTA Etiquetas personalizadas

COMO

SELECCIONE c.Número de cliente, c.Nombre de cliente

DE Clientes c

Esto crearía una vista que incluye los valores de las columnas CustomerNumber y CustomerName. Una vez creada, una vista se puede utilizar como fuente de datos en otras consultas.

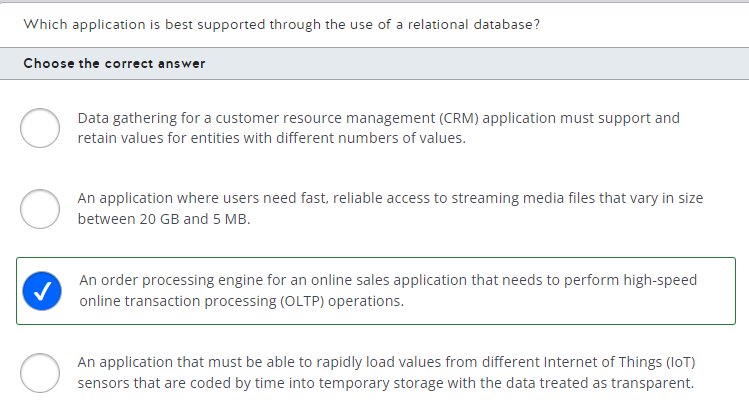
La clave principal es el valor utilizado para identificar cada entidad de tabla como única. El valor de la clave principal para cada fila será único.

Ninguna de las descripciones es para una mesa. Una tabla es la estructura de almacenamiento básica para una base de datos relacional que consta de filas y columnas.

Ninguna de las descripciones es para una clave externa. Las claves foráneas se utilizan con claves primarias para establecer relaciones entre tablas. La clave externa en una tabla está asociada con la clave principal en otra tabla. Una clave externa se asociará con una sola clave principal, pero se pueden asociar varias claves externas con la misma clave principal.

**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)

1. D



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.

**Traducción**

La aplicación más adecuada para un almacén de datos relacional es un motor de procesamiento de pedidos para una aplicación de ventas en línea que necesita realizar operaciones de procesamiento de transacciones en línea (OLTP) de alta velocidad. Las características de los datos en las bases de datos relacionales incluyen:

• Altamente normalizado con esquemas forzados

• Requiere alta integridad y fuerte consistencia

• Las relaciones se mantienen entre las tablas de datos

Además de la gestión de pedidos, las bases de datos relacionales suelen utilizarse para dar soporte a aplicaciones de control de inventario y contabilidad.

Para una aplicación en la que los usuarios necesitan un acceso rápido y confiable a la transmisión de archivos multimedia que varían en tamaño entre 20 GB y 5 MB, debe elegir una solución de almacenamiento de objetos como Azure Blob Storage o Azure Data Lake Storage Gen2.

Al recopilar datos para una aplicación de administración de recursos del cliente (CRM) que debe admitir y retener valores para entidades con diferentes números de valores, lo más probable es que utilice una solución de almacenamiento de documentos. Los datos del documento están semiestructurados y cada documento define internamente su propio esquema. Cada documento se escribe y recupera como un solo bloque. Lo más probable es que utilice Azure Cosmos DB como su solución de almacenamiento.

Para admitir una aplicación que debe poder cargar rápidamente valores de diferentes sensores IoT que están codificados por tiempo en almacenamiento temporal con los datos tratados como transparentes, lo más probable es que utilice una solución de almacenamiento de clave/valor. Lo más probable es que use Azure Cosmos DB Table API o Azure Table Storage como su almacén de datos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should create a view. A view is a virtualized table that is recreated each time you execute the view query. The creation table for 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 CustomerName column values. Once created, a view can be used as a data source in other queries.  
  
Because the view is a virtualized table, its content is not physically stored in the database but is instead repopulated each time it is required. This helps to keep your data normalized and prevents data duplication.  
  
You should not create an index. An index is a relational database structure that is used to sort data to help optimize searches during queries.  
  
You should not create a table. Creating a table would result in data duplication, and it would also require you to code a method to keep the table up-to-date when data is modified in the source table.  
  
You should not create a key. Relational database management systems (RDMSs) typically use primary and foreign keys. Primary keys are used to enforce uniqueness on rows in a table. Foreign keys are used with primary keys to establish relationships between tables.

**Traducción**

Deberías crear una vista. Una vista es una tabla virtualizada que se recrea cada vez que ejecuta la consulta de vista. La tabla de creación de una vista simple podría tener el siguiente aspecto:

CREAR VISTA Etiquetas personalizadas

COMO

SELECCIONE c.Número de cliente, c.Nombre de cliente

DE Clientes c

Esto crearía una vista que incluye los valores de las columnas CustomerNumber y CustomerName. Una vez creada, una vista se puede utilizar como fuente de datos en otras consultas.

Debido a que la vista es una tabla virtualizada, su contenido no se almacena físicamente en la base de datos, sino que se vuelve a llenar cada vez que se requiere. Esto ayuda a mantener sus datos normalizados y evita la duplicación de datos.

No debe crear un índice. Un índice es una estructura de base de datos relacional que se utiliza para clasificar datos para ayudar a optimizar las búsquedas durante las consultas.

No debe crear una tabla. La creación de una tabla daría como resultado la duplicación de datos y también requeriría codificar un método para mantener la tabla actualizada cuando se modifican los datos en la tabla de origen.

No debe crear una clave. Los sistemas de gestión de bases de datos relacionales (RDMS) suelen utilizar claves primarias y externas. Las claves principales se utilizan para imponer la unicidad en las filas de una tabla. Las claves foráneas se utilizan con claves primarias para establecer relaciones entre tablas.

**References**  
  
[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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

Una clave principal es una restricción que se utiliza para hacer cumplir la integridad de los datos en las bases de datos relacionales al indicar qué columna (o combinación de columnas) identifica de forma única cada fila de una tabla. La clave principal debe tener un valor único en una tabla y también se usa para crear relaciones entre diferentes tablas.

Una clave externa es una restricción que se utiliza para crear relaciones entre tablas al hacer referencia a una clave principal de otras tablas.

Un montón es una tabla que no tiene implementado un índice agrupado. En un montón, los datos se almacenan en una estructura desordenada.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

A view is a database object where content is based on a query. You can use a view to query data from one or more tables and display the data in a different structure, like filtering or joining data from different tables.  
  
A table is a database object that stores data in rows and columns. Tables are used to represent entities with each row representing a single entity. Each characteristic from this entity is represented as a column.  
  
A heap is a table that does not have a clustered index implemented. A clustered index is a data structure associated with a table that defines the order that rows are stored in a disk. In a heap, data is stored in an unordered structure.  
  
An index is a structure that improves the read performance to find specific data in a table.

**Traducción**

Una vista es un objeto de base de datos cuyo contenido se basa en una consulta. Puede usar una vista para consultar datos de una o más tablas y mostrar los datos en una estructura diferente, como filtrar o unir datos de diferentes tablas.

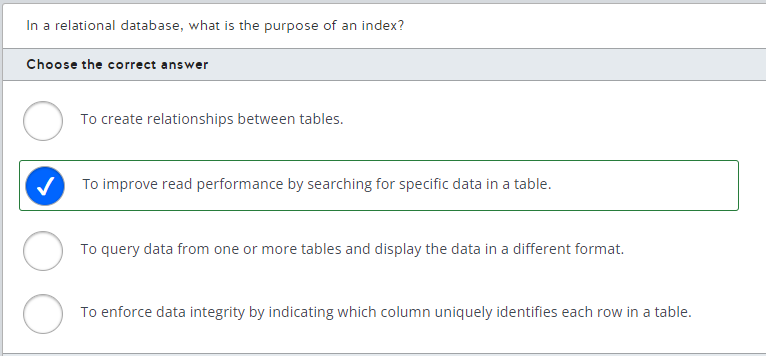
Una tabla es un objeto de base de datos que almacena datos en filas y columnas. Las tablas se utilizan para representar entidades en las que cada fila representa una sola entidad. Cada característica de esta entidad se representa como una columna.

Un montón es una tabla que no tiene implementado un índice agrupado. Un índice agrupado es una estructura de datos asociada con una tabla que define el orden en que se almacenan las filas en un disco. En un montón, los datos se almacenan en una estructura desordenada.

Un índice es una estructura que mejora el rendimiento de lectura para encontrar datos específicos en una tabla.

**References**  
  
[Explore relational data structures](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/3-explore-structures)  
[Explore the characteristics of relational data](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/2-explore-characteristics)  
  
[Heaps (Tables without Clustered Indexes)](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes?view=sql-server-ver15)  
  
[Clustered and nonclustered indexes described](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15)

1. D



The purpose of an index in a relational database is to improve read performance by searching for specific data in a table. An index stores values and references to specific rows in a table and uses these values to locate the specific rows more quickly.  
  
An index does not enforce data integrity by indicating which column uniquely identifies each row in a table. This is the purpose of a primary key.  
  
An index does not create relationships between tables. This is the purpose of a foreign key.  
  
An index does not query data from one or more tables and display the data in a different format. This is the purpose of a view.

**Traducción**

El propósito de un índice en una base de datos relacional es mejorar el rendimiento de lectura mediante la búsqueda de datos específicos en una tabla. Un índice almacena valores y referencias a filas específicas en una tabla y usa estos valores para ubicar las filas específicas más rápidamente.

Un índice no impone la integridad de los datos al indicar qué columna identifica de forma única cada fila en una tabla. Este es el propósito de una clave primaria.

Un índice no crea relaciones entre tablas. Este es el propósito de una clave externa.

Un índice no consulta datos de una o más tablas y muestra los datos en un formato diferente. Este es el propósito de una vista.

**References**  
  
[Explore relational data structures](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/3-explore-structures)  
  
[Primary and Foreign Key Constraints](https://docs.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver15)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

You should use a relational database to guarantee strong consistency while processing transactions. A relational database can guarantee the consistency and availability of the data while processing transactions. A relational database must ensure Atomicity, Consistency, Isolation, Durability (ACID) for transactional workloads.  
  
You should not use a relational database to store data in columnar format. A columnar data store consists of row identifiers and a group of information stored in a column. Each group of information is stored in a keyspace, composed of a single column. You should use a column family database like Apache Cassandra to store this kind of data instead.  
  
You should not use a relational database for dynamic and semi-structured data schema. A relational database requires a data schema, defined as tables and columns. You should use a document database like MongoDB to store this kind of data instead.  
  
You should not use a relational database for large images and videos. A relational database is not optimized to store binary data. You should use a binary large object (Blob) like Azure Blob storage instead.

**Traducción**

Debe utilizar una base de datos relacional para garantizar una gran coherencia al procesar las transacciones. Una base de datos relacional puede garantizar la consistencia y disponibilidad de los datos mientras se procesan las transacciones. Una base de datos relacional debe garantizar la atomicidad, la consistencia, el aislamiento y la durabilidad (ACID) para las cargas de trabajo transaccionales.

No debe utilizar una base de datos relacional para almacenar datos en formato de columnas. Un almacén de datos en columnas consta de identificadores de fila y un grupo de información almacenada en una columna. Cada grupo de información se almacena en un espacio de claves, compuesto por una sola columna. En su lugar, debe usar una base de datos de familia de columnas como Apache Cassandra para almacenar este tipo de datos.

No debe utilizar una base de datos relacional para esquemas de datos dinámicos y semiestructurados. Una base de datos relacional requiere un esquema de datos, definido como tablas y columnas. Debería usar una base de datos de documentos como MongoDB para almacenar este tipo de datos.

No debe usar una base de datos relacional para imágenes y videos grandes. Una base de datos relacional no está optimizada para almacenar datos binarios. En su lugar, debe usar un objeto binario grande (Blob) como Azure Blob Storage.

**References**  
  
[Identify data formats](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-data-formats)  
  
[Explore the characteristics of relational data](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/2-explore-characteristics)  
  
[Relational vs. NoSQL data](https://docs.microsoft.com/en-us/dotnet/architecture/cloud-native/relational-vs-nosql-data)  
  
[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)

1. D

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

You should use a primary key to uniquely identify each row in a table. A primary key enforces uniqueness automatically by creating a unique constraint on primary key columns.  
  
You should use a view to easily access aggregated results from a table. Views are derived from a result of a table. You can also filter view data.  
  
You should use an index to quickly access rows using specific product tags in a table.  You can create an index in the product column in a table to quickly access rows using the product tag. Indexes help you to quickly find the desired rows based on your query.  
  
You should not use a stored procedure. Stored procedures are used to execute a set of commands or SQL statements in a single batch. They are often used for more complex logic. They can return output from input parameters.

**Traducción**

Debe usar una clave principal para identificar de forma única cada fila en una tabla. Una clave principal impone la unicidad automáticamente mediante la creación de una restricción única en las columnas de la clave principal.

Debe usar una vista para acceder fácilmente a los resultados agregados de una tabla. Las vistas se derivan de un resultado de una tabla. También puede filtrar los datos de vista.

Debe usar un índice para acceder rápidamente a las filas usando etiquetas de productos específicas en una tabla. Puede crear un índice en la columna del producto en una tabla para acceder rápidamente a las filas usando la etiqueta del producto. Los índices lo ayudan a encontrar rápidamente las filas deseadas según su consulta.

No debe utilizar un procedimiento almacenado. Los procedimientos almacenados se utilizan para ejecutar un conjunto de comandos o declaraciones SQL en un solo lote. A menudo se utilizan para una lógica más compleja. Pueden devolver la salida de los parámetros de entrada.

**References**  
  
[Describe database objects](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/5-database-objects)  
  
[Explore relational data structures](https://docs.microsoft.com/en-us/learn/modules/describe-concepts-of-relational-data/3-explore-structures)  
  
[Primary and Foreign Key Constraints](https://docs.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver15)  
  
[Views](https://docs.microsoft.com/en-us/sql/relational-databases/views/views?view=sql-server-ver15)  
  
[Indexes](https://docs.microsoft.com/en-us/sql/relational-databases/indexes/indexes?view=sql-server-ver15)  
  
[Stored Procedures (Database Engine)](https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15)

1. D

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

The UPDATE command is an example of a DML command. It is used to edit column values for rows in a table.  
  
The SQL language has four basic types of commands. DML commands are used to manipulate rows in a table. They include:

* DELETE
* INSERT
* SELECT
* UPDATE

Data Definition Language (DDL) commands are used to create, modify, and delete database objects. They include:

* ALTER
* CREATE
* DROP
* RENAME

Data Control Language (DCL) commands are for access control and permission management. They include:

* DENY
* GRANT
* REVOKE

**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
  
[UPDATE - SQL Command](https://docs.microsoft.com/en-us/sql/odbc/microsoft/update-sql-command?view=sql-server-ver15)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

Debe usar el comando DROP para eliminar una tabla de base de datos y su contenido. El comando DROP es un ejemplo de un comando de lenguaje de definición de datos (DDL). Los comandos DDL se utilizan para crear, modificar y eliminar objetos de la base de datos.

No debe usar el comando ALTER. ALTER también es un comando DDL, pero se usa para modificar un objeto de base de datos, como agregar una columna a una tabla.

No debe utilizar los comandos ELIMINAR o ACTUALIZAR. Ambos son comandos de lenguaje de manipulación de datos (DML) que se utilizan para manipular datos, como ejecutar consultas para recuperar datos. El comando DELETE se usa para eliminar filas de una tabla. El comando ACTUALIZAR se usa para editar valores de columna.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

INSERT is a DML statement type. DML statements are used to manipulate information stored in a relational database. Other DML statements include UPDATE, DELETE, and SELECT.  
  
CREATE, ALTER, and DROP are not DML statements. These statements are classified as Data Definition Language (DDL) statements. DDL statements are used to define data structures in a relational database, like tables, views, and indexes.

**Traducción**

INSERT es un tipo de instrucción DML. Las declaraciones DML se utilizan para manipular la información almacenada en una base de datos relacional. Otras instrucciones DML incluyen UPDATE, DELETE y SELECT.

CREATE, ALTER y DROP no son sentencias DML. Estas declaraciones se clasifican como declaraciones de lenguaje de definición de datos (DDL). Las declaraciones DDL se utilizan para definir estructuras de datos en una base de datos relacional, como tablas, vistas e índices.

**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)

1. D

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.

**Traducción**

ALTER es un tipo de instrucción DDL. Las declaraciones DDL se utilizan para definir estructuras de datos en una base de datos relacional, como tablas, vistas e índices. Otras declaraciones DDL son CREATE y DROP.

SELECT, INSERT y DELETE no son sentencias DDL. Estas declaraciones se clasifican como declaraciones de lenguaje de manipulación de datos (DML). Las declaraciones DML se utilizan para manipular la información almacenada en una base de datos relacional.

**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)

1. D

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.

**Traducción**

Debe usar la instrucción UPDATE. La instrucción UPDATE puede actualizar datos en las filas existentes de una tabla. Puede actualizar la cantidad de existencias a 250 para el producto con ID 5323 mediante la siguiente consulta:

ACTUALIZAR SET StockQuantity = 250 DONDE ProductId = 5323;

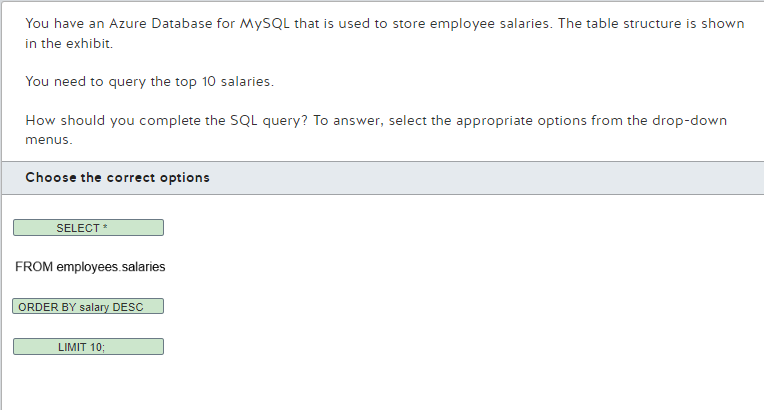
No debe utilizar la sentencia TRUNCATE. La instrucción TRUNCATE elimina todas las filas de una tabla sin registrar las eliminaciones de filas individuales.

No debe utilizar la sentencia INSERT. La declaración INSERT se usa para agregar nuevas filas en una tabla.

No debe utilizar la sentencia CREATE. La declaración CREATE se usa para definir objetos de base de datos, como tablas, vistas, índices y otros.

**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)

1. D



Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should complete the SQL query as follows:  
  
SELECT \*  
FROM employees.salaries  
ORDER BY salary DESC  
LIMIT 10;  
  
You should use the SELECT \* clause to return all fields from the salaries table.  
  
You should not use the SELECT TOP(10) \* clause. You can use SELECT TOP(10) with SQL Server or other T-SQL compatible databases to return the specified number of rows.  
  
You should not use the SELECT MAX(salary) clause. You can use this clause to return the maximum salary in the salaries table.  
  
You should use the ORDER BY salary DESC clause to order the salaries in descending order, from the highest to the lowest salary.  
  
You should not use the ORDER BY salary ASC or ORDER BY salary clauses. This returns the salaries ordered from the lowest to the highest salary.  
  
You should use the LIMIT 10; clause to limit the number of rows returned from the query and show the top 10 salaries.  
  
You should not use the LIMIT 1, 10; clause. This limits the number of rows returned to 10, starting from the second row. In this case, the top salary will be omitted, returning the second highest salary to the eleventh one, instead of the top ten.  
  
You should not use the DESC; clause. There is no difference if you insert a line break after the ORDER BY clause.  
  
If you are using a T-SQL compatible database, a valid query to retrieve the top ten salaries would be:  
  
SELECT TOP(10) \*  
FROM employees.salaries  
ORDER BY salary  
DESC;

**Traducción**

Debe completar la consulta SQL de la siguiente manera:

SELECCIONE \*

DE empleados.salarios

ORDEN POR salario DESC

LÍMITE 10;

Debe usar la cláusula SELECT \* para devolver todos los campos de la tabla de salarios.

No debe usar la cláusula SELECT TOP(10) \*. Puede usar SELECT TOP(10) con SQL Server u otras bases de datos compatibles con T-SQL para devolver el número especificado de filas.

No debe usar la cláusula SELECT MAX (salario). Puede utilizar esta cláusula para devolver el salario máximo en la tabla de salarios.

Se debe utilizar la cláusula ORDER BY desalario DESC para ordenar los sueldos en orden descendente, del mayor al menor.

No debe utilizar las cláusulas ORDER BY salario ASC u ORDER BY salario. Devuelve los salarios ordenados de menor a mayor salario.

Debe usar el LÍMITE 10; cláusula para limitar el número de filas devueltas por la consulta y mostrar los 10 mejores salarios.

No debe usar el LÍMITE 1, 10; cláusula. Esto limita el número de filas devueltas a 10, a partir de la segunda fila. En este caso, se omitirá el salario más alto, devolviendo el segundo salario más alto al undécimo, en lugar de los diez primeros.

No debe usar el DESC; cláusula. No hay diferencia si inserta un salto de línea después de la cláusula ORDER BY.

Si está utilizando una base de datos compatible con T-SQL, una consulta válida para recuperar los diez salarios principales sería:

SELECCIONA ARRIBA(10) \*

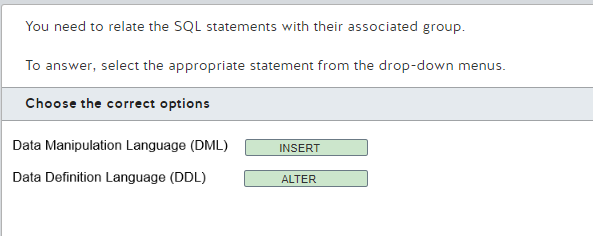
DE empleados.salarios

ORDEN POR salario

DESC;

**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
  
[Query relational data in Azure SQL Database for MySQL](https://docs.microsoft.com/en-us/learn/modules/query-relational-data/5-sql-database-for-mysql)  
  
[SELECT Statement](https://dev.mysql.com/doc/refman/8.0/en/select.html)  
  
[TOP (Transact-SQL)](https://docs.microsoft.com/en-us/sql/t-sql/queries/top-transact-sql?view=sql-server-ver15)

1. D



You should use the INSERT statement with the DML group. DML statements deal with information stored in the database table. The following are examples of DML statements: SELECT, INSERT, UPDATE, and DELETE.  
  
You should use the ALTER statement with the DDL group. DDL statements deal with creating, altering, or deleting a database object or structure. The following are examples of DDL statements: CREATE, ALTER, DROP, and RENAME.  
  
You should not use the GRANT statement for either the DDL or DML group. GRANT statements are related to the Data Control Language (DCL) group. They are used to grant permission on database objects.

**Traducción**

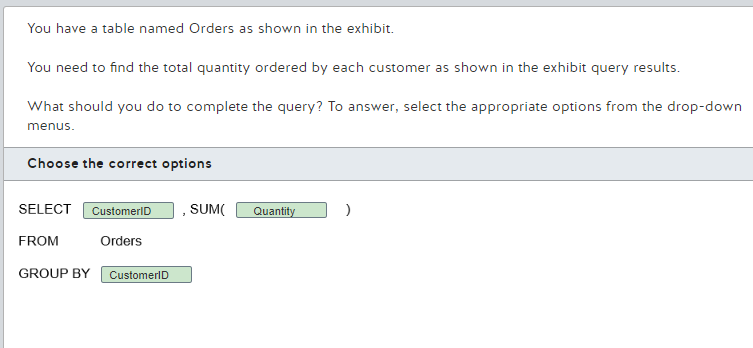
Debe usar la declaración INSERT con el grupo DML. Las declaraciones DML se ocupan de la información almacenada en la tabla de la base de datos. Los siguientes son ejemplos de sentencias DML: SELECT, INSERT, UPDATE y DELETE.

Debe usar la instrucción ALTER con el grupo DDL. Las declaraciones DDL se ocupan de la creación, modificación o eliminación de un objeto o estructura de base de datos. Los siguientes son ejemplos de sentencias DDL: CREATE, ALTER, DROP y RENAME.

No debe usar la declaración GRANT para el grupo DDL o DML. Las sentencias GRANT están relacionadas con el grupo de lenguaje de control de datos (DCL). Se utilizan para otorgar permisos sobre los objetos de la base de datos.

**References**  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)  
[Introduction to SQL](https://docs.microsoft.com/en-us/learn/modules/query-relational-data/2-introduction-to-sql)  
  
[Transact-SQL statements](https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver15)

1. D



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.

**Traducción**

Debe usar la siguiente consulta para encontrar la cantidad total ordenada por cada cliente:

SELECCIONE CustomerID, SUMA (Cantidad)

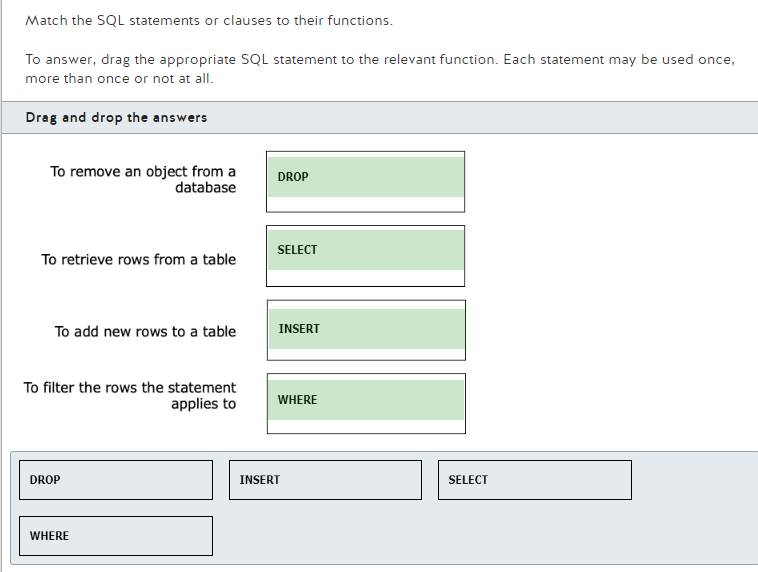
DESDE Pedidos

AGRUPAR POR ID de cliente

Utilice la función de agregado SUM() para calcular la cantidad total. Para poder ver la cantidad total comprada por cada cliente, debe agregar CustomerID a la cláusula GROUP BY, lo que le permite mostrar resultados agregados por 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)

1. D



DROP is used to remove objects from a database. It can be used to remove columns, views, stored procedures or whole tables. In order to remove rows, you should use DELETE.  
  
SELECT is used to retrieve rows from a table. By default, it will be performed on all rows unless a filtering criteria is added to the statement.  
  
INSERT is used to add new rows to a table. It can add a single row, or multiple rows at a time.  
  
WHERE is used to filter rows the statement applies to. It will be used after the initial statement, such as SELECT or UPDATE. Its usage will be within a WHERE clause.

**Traducción**

DROP se utiliza para eliminar objetos de una base de datos. Se puede usar para eliminar columnas, vistas, procedimientos almacenados o tablas completas. Para eliminar filas, debe usar DELETE.

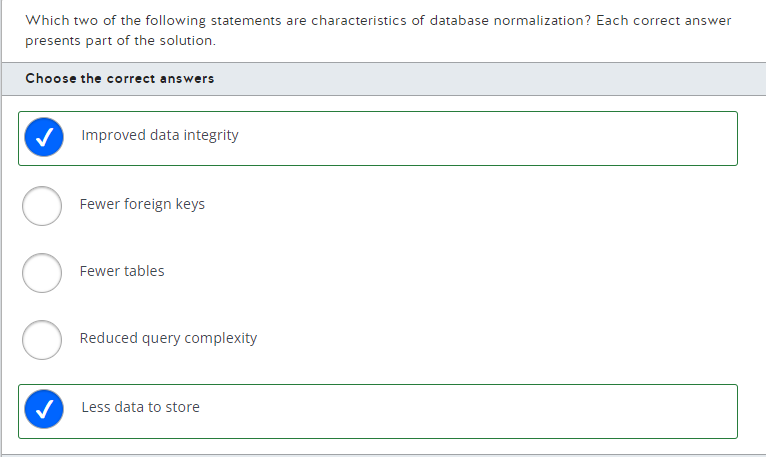
SELECT se utiliza para recuperar filas de una tabla. De forma predeterminada, se realizará en todas las filas a menos que se agregue un criterio de filtrado a la declaración.

INSERT se usa para agregar nuevas filas a una tabla. Puede agregar una sola fila o varias filas a la vez.

WHERE se usa para filtrar las filas a las que se aplica la declaración. Se utilizará después de la instrucción inicial, como SELECCIONAR o ACTUALIZAR. Su uso estará dentro de una cláusula WHERE.

**References**[Transact-SQL statements](https://docs.microsoft.com/en-us/sql/t-sql/statements/statements?view=sql-server-ver15)  
[SQL Keywords Reference](https://www.w3schools.com/sql/sql_ref_keywords.asp)  
  
[Explore SQL](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/4-query-with-sql)

1. D



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.

**Traducción**

Menos datos para almacenar es una característica de la normalización. La normalización elimina los datos redundantes al eliminar la duplicación, lo que genera menos datos en general. Potencialmente, esto podría conducir a una reducción en sus costos de almacenamiento.

La integridad de datos mejorada también es una característica de la normalización. La normalización elimina los datos duplicados; cada vez que tiene los mismos datos agregados en más de un lugar, aumenta la posibilidad de que se actualice de manera inconsistente debido a un error.

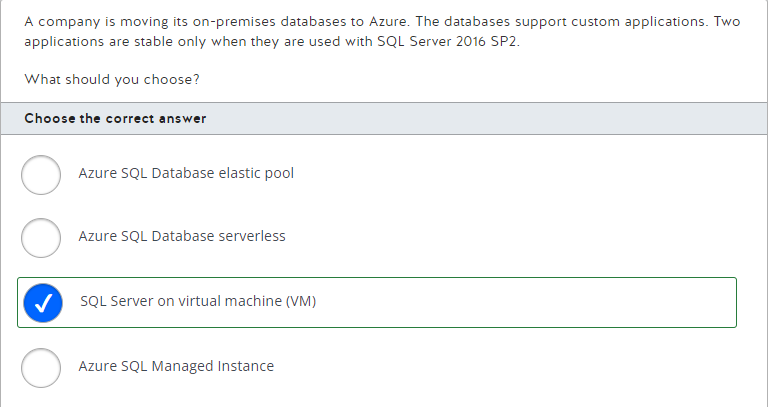
Menos tablas no son una característica de la normalización. La normalización aumenta la cantidad de tablas al agregar cada entidad de datos en una tabla diferente y hacer referencia a registros en otras tablas a través de columnas de clave externa.

La complejidad reducida de la consulta no es una característica de la normalización. Es más sencillo escribir una consulta en una tabla que en dos, ya que deberá unir las dos tablas a través de una relación entre los datos para escribir su consulta. La normalización aumenta la cantidad de tablas, lo que significa que necesitará usar más uniones en sus consultas y aumentar su complejidad.

Menos claves foráneas no es una característica de la normalización. La normalización aumenta el número de tablas al dividir los datos duplicados en su propia tabla, a la que luego se hace referencia en la tabla original a través de una clave externa. Por lo tanto, la normalización aumentará el número de claves foráneas en la base de datos.

**References**  
  
[Understand normalization](https://docs.microsoft.com/en-us/learn/modules/explore-relational-data-offerings/3-normalization)

1. D



You should choose SQL Server on VM. This type of move is often called lift-and-shift and provides for rapid migration to the cloud. SQL Server on VM is the option most compatible with your on-premises database servers and is the only deployment option that lets you run a SQL Server version other than the latest stable version. This is an Infrastructure as a Service (IaaS) deployment, where you are responsible for managing the operating system and database server.  
  
Azure SQL Database serverless, Azure SQL Database elastic pool, and Azure SQL Managed Instance are all Platform as a Service (PaaS) deployments. With PaaS, you do not have access to the operating system, and some database administrative activities, such as keeping SQL Server updated, are handled for you. These deployments run the most recent stable SQL Server version. While these implementations are close, they are not 100 % compatible with an on-premises implementation.  
  
The deployment option most compatible with an on-premises instance is SQL Server managed instance. This can be used in most lift-and-shift scenarios, but it is limited to the latest SQL Server version.

**Traducción**

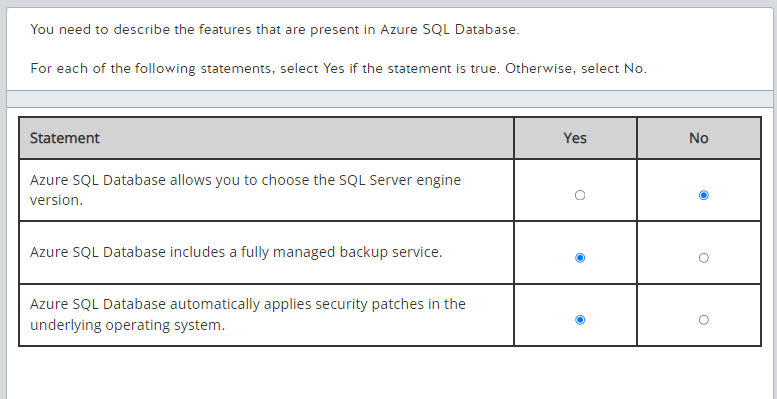
Debe elegir SQL Server en VM. Este tipo de movimiento a menudo se denomina lift-and-shift y proporciona una migración rápida a la nube. SQL Server en VM es la opción más compatible con sus servidores de bases de datos locales y es la única opción de implementación que le permite ejecutar una versión de SQL Server distinta de la última versión estable. Esta es una implementación de Infraestructura como servicio (IaaS), donde usted es responsable de administrar el sistema operativo y el servidor de la base de datos.

Azure SQL Database sin servidor, el grupo elástico de Azure SQL Database y Azure SQL Managed Instance son implementaciones de plataforma como servicio (PaaS). Con PaaS, no tiene acceso al sistema operativo y algunas actividades administrativas de la base de datos, como mantener actualizado SQL Server, se manejan por usted. Estas implementaciones ejecutan la versión estable más reciente de SQL Server. Si bien estas implementaciones están cerca, no son 100 % compatibles con una implementación local.

La opción de implementación más compatible con una instancia local es la instancia administrada de SQL Server. Esto se puede usar en la mayoría de los escenarios de elevación y cambio, pero está limitado a la última versión de SQL Server.

**References**  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)  
  
[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)  
  
[What is Azure SQL Database?](https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview)  
  
[What is Azure SQL Managed Instance?](https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview)

1. D



Azure SQL Database does not allow you to choose the SQL Server engine version. Azure SQL Database uses the latest stable version of the SQL Server engine. You should use SQL Server on Azure Virtual Machine (VM) to choose which SQL Server version to use.  
  
Azure SQL Database includes a fully managed backup service and automatically applies security patches in the underlying operating system. Azure SQL Database is a platform as a service (PaaS) database offering, where most of the database management functions are handled by Azure. This includes backups, operating system updates, and applying security patches.

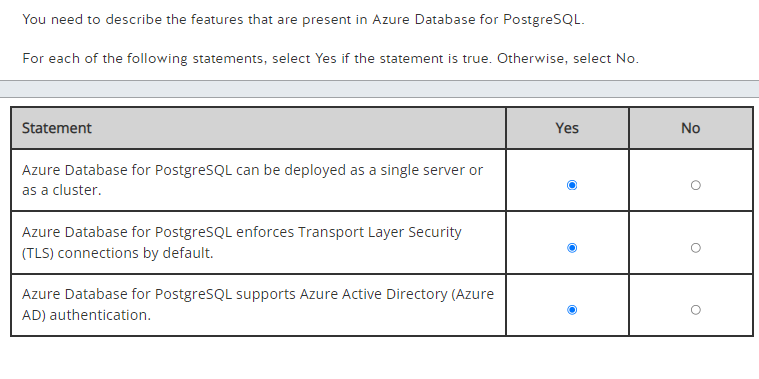
**Traducción**

Azure SQL Database no le permite elegir la versión del motor de SQL Server. Azure SQL Database usa la última versión estable del motor de SQL Server. Debe usar SQL Server en Azure Virtual Machine (VM) para elegir qué versión de SQL Server usar.

Azure SQL Database incluye un servicio de respaldo totalmente administrado y aplica automáticamente parches de seguridad en el sistema operativo subyacente. Azure SQL Database es una oferta de base de datos de plataforma como servicio (PaaS), donde Azure maneja la mayoría de las funciones de administración de la base de datos. Esto incluye copias de seguridad, actualizaciones del sistema operativo y la aplicación de parches de seguridad.

**References**  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)  
  
[What is Azure SQL Database?](https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview)  
  
[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)

1. D



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.

**Traducción**

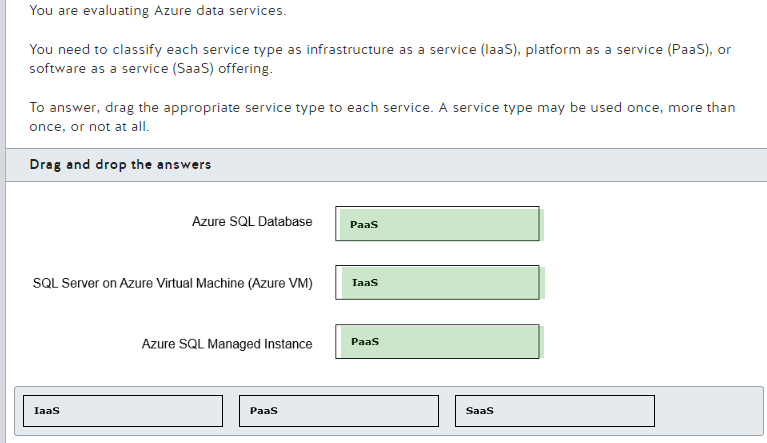
Azure Database for PostgreSQL se puede implementar como un servidor único o como un clúster. Puede implementar Azure Database for PostgreSQL como un servidor único o como un clúster de Hiperescala (Citus). Azure Database for PostgreSQL Hiperescala (Citus) puede escalar horizontalmente las consultas en varios equipos mediante la fragmentación.

Azure Database for PostgreSQL aplica conexiones TLS de forma predeterminada. Hacer cumplir las conexiones TLS mejora la seguridad al cifrar la conexión entre el cliente y el servidor de la base de datos. Puede deshabilitar la conexión TLS en implementaciones de un solo servidor, pero esto no se recomienda.

Azure Database for PostgreSQL admite la autenticación de Azure AD. Puede configurar la autenticación de Azure AD con Azure Database for PostgreSQL para permitir que los usuarios se conecten a la base de datos con sus propias credenciales. Esto centraliza la gestión de usuarios en un solo lugar.

**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)

1. D



You should classify Azure SQL Database as a PaaS offering. Most of the database management functions are handled by Azure in Azure SQL Database, including a fully managed backup service. They automatically apply security patches in the underlying operating system.  
  
You should classify SQL Server on Azure VM as an IaaS offering. You should use SQL Server on Azure VM to have complete access to the infrastructure, allowing you to choose which SQL Server version to use and to configure the underlying operating system. This model of offering is good for lift and shift scenarios, where you move an on-premises virtual machine as-is to the cloud.  
  
You should classify Azure SQL Managed Instance as a PaaS offering. Azure SQL Managed Instance brings more compatibility and resources that are really close to on-premises SQL Server deployments. However, you can still have the benefits of a PaaS offering, like a fully managed backup service, automatic application of security patches, and high availability.  
  
You should not classify any of these services as SaaS. SaaS is offered as ready-to-use applications in the cloud, such as emails, communication tools, and office tools, such as Microsoft 365.

**Traducción**

Debe clasificar Azure SQL Database como una oferta de PaaS. La mayoría de las funciones de administración de la base de datos están a cargo de Azure en Azure SQL Database, incluido un servicio de copia de seguridad completamente administrado. Aplican automáticamente parches de seguridad en el sistema operativo subyacente.

Debe clasificar SQL Server en Azure VM como una oferta de IaaS. Debe usar SQL Server en Azure VM para tener acceso completo a la infraestructura, lo que le permite elegir qué versión de SQL Server usar y configurar el sistema operativo subyacente. Este modelo de oferta es bueno para escenarios de elevación y cambio, donde mueve una máquina virtual local tal cual a la nube.

Debe clasificar Instancia administrada de Azure SQL como una oferta de PaaS. Instancia administrada de Azure SQL brinda más compatibilidad y recursos que están muy cerca de las implementaciones locales de SQL Server. Sin embargo, aún puede disfrutar de los beneficios de una oferta de PaaS, como un servicio de respaldo completamente administrado, la aplicación automática de parches de seguridad y alta disponibilidad.

No debe clasificar ninguno de estos servicios como SaaS. SaaS se ofrece como aplicaciones listas para usar en la nube, como correos electrónicos, herramientas de comunicación y herramientas de oficina, como Microsoft 365.

**References**  
  
[Describe Azure SQL services and capabilities](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-relational-database-offerings-azure/2-azure-sql)  
  
[What is SaaS?](https://azure.microsoft.com/en-us/overview/what-is-saas/)  
  
[What is Azure SQL Database?](https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview)  
  
[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)  
  
[What is Azure SQL Managed Instance?](https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview)

1. D

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.

**Traducción**

Debe usar Instancia administrada de Azure SQL. Azure SQL Managed Instance se usa para la implementación en la que necesita tener una paridad completa de características con SQL Server local. Debe usar una instancia administrada si su base de datos usa funciones como el Agente SQL Server y el Correo electrónico de la base de datos.

No debe usar Azure SQL Database. Azure SQL Database le brinda una única base de datos en la nube con un costo y una administración mínimos, en la que puede crear bases de datos y tablas. Azure gestiona tareas administrativas como la copia de seguridad y la recuperación. Sin embargo, Azure SQL Database no implementa características como el Agente SQL Server y el Correo electrónico de base de datos, que son necesarios en este escenario.

No debe usar SQL Server en Azure Virtual Machines. SQL Server en Azure Virtual Machines le brinda una experiencia administrativa similar a la de su servidor local, lo que, por lo tanto, aumentará su esfuerzo administrativo.

No debe usar una base de datos de Azure para MariaDB. Una base de datos de Azure para MariaDB le permite implementar la edición comunitaria de MariaDB en Azure. MariaDB es un sistema de gestión de bases de datos relacionales.

**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)

1. D

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.

**Traducción**

Debe usar las ofertas de servicios de datos relacionales de Azure SQL Database para PaaS. Azure SQL Database es un servidor de base de datos administrado en la nube.

También puede usar Instancia administrada de Azure SQL Database para ofertas de servicios de datos relacionales de PaaS. Instancia administrada de Azure SQL Database le permite ejecutar varias bases de datos en la misma instancia.

No debe usar SQL Server en Azure Virtual Machines. SQL Server es una oferta de infraestructura como servicio (IaaS) para servicios de datos. Le permite ejecutar SQL Server en una máquina virtual de Azure al proporcionar la infraestructura relevante.

No debe usar Azure Cosmos DB. Azure Cosmos DB es una oferta de PaaS dirigida principalmente a almacenes de datos no relacionales. Le permite almacenar documentos y objetos.

**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)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

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.

**Traducción**

SQL Server en Azure Virtual Machines (VM) es un servicio de infraestructura como servicio (IaaS). Esta forma de servicio en la nube requiere la mayor responsabilidad para el cliente, ya que necesita administrar todo lo que no sea los dispositivos físicos utilizados. SQL Server en Azure Virtual Machines crea hardware virtual para que los clientes lo usen como deseen.

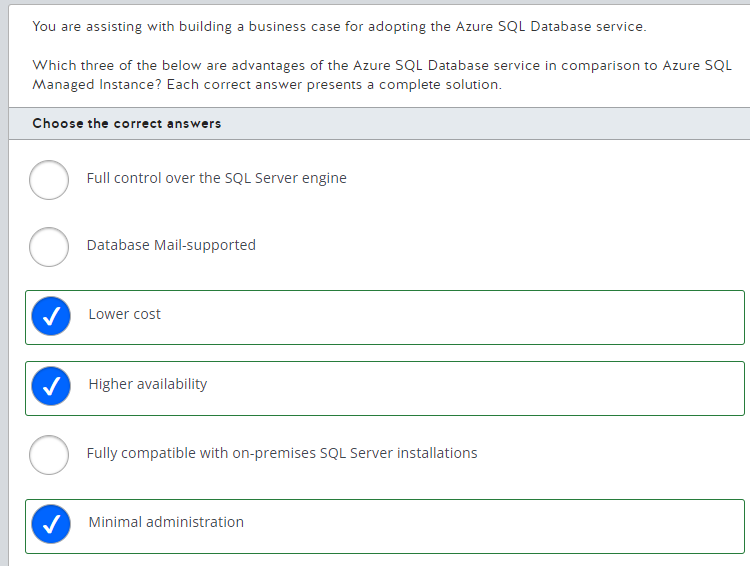
Azure SQL Managed Instance es un servicio de plataforma como servicio (PaaS). No requiere que los clientes administren hardware físico o sistemas operativos e incluye copias de seguridad y actualizaciones de software automatizadas; los clientes suelen utilizar este servicio para migrar los sistemas existentes a la nube.

Azure SQL Database es un servicio de plataforma como servicio (PaaS). Similar a Instancia administrada de Azure SQL, esta es una solución completamente administrada que maneja las copias de seguridad, el monitoreo y las actualizaciones para el cliente. Los clientes utilizan este servicio como una capa de datos de alta disponibilidad para sus sistemas basados ​​en la nube.

SaaS significa software como servicio. SaaS no requiere que el cliente administre dispositivos físicos, sistemas operativos o actualizaciones del software. Los usuarios solo necesitan crear cuentas y administrar el acceso. Un ejemplo de SaaS sería Microsoft Dynamics 365 o 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)

1. D



Azure SQL Database has a lower cost in comparison to Azure SQL Managed Instance. It is a relational database, designed for modern cloud applications. It is used by customers that require a high availability, and a scalable database layer for their cloud applications.  
  
Azure SQL Database is a platform-as-a-service (PaaS) service. Although Azure SQL Managed Instance is also a PaaS service, Azure SQL Database eliminates more administrative tasks like removing the need to manage the SQL Server configuration.  
  
Azure SQL Database offers 99.995% uptime, in comparison with Azure SQL Managed Instance that offers 99.99% uptime.  
  
Azure SQL Database is not fully compatible with on-premises SQL Server installations. Some features are available in on-premises SQL Server that are not available in Azure SQL Database. Azure SQL Managed Instance offers almost 100% compatibility as it has additional configuration options, whereas SQL Server on Azure Virtual Machines is fully compatible as it allows the most granular level of user configuration.  
  
Azure SQL Database does not allow full control over the SQL Server engine. Azure SQL Database performs some tasks for the user, such as performing back-ups and installing patches, but only SQL Server on Azure Virtual Machines allows full control over the engine. This is because SQL Server on Azure Virtual Machines is an infrastructure as a service (IaaS) service, which requires users to manage operating systems.  
  
Azure SQL Database does not support Database Mail. This feature, which allows the database to send email messages to users, is not available in the Azure SQL Database. It is, however, offered in Azure SQL Managed Instance and SQL Server on Azure Virtual Machines.

**Traducción**

Azure SQL Database tiene un costo menor en comparación con Azure SQL Managed Instance. Es una base de datos relacional, diseñada para aplicaciones modernas en la nube. Es utilizado por clientes que requieren una alta disponibilidad y una capa de base de datos escalable para sus aplicaciones en la nube.

Azure SQL Database es un servicio de plataforma como servicio (PaaS). Aunque Azure SQL Managed Instance también es un servicio PaaS, Azure SQL Database elimina más tareas administrativas, como eliminar la necesidad de administrar la configuración de SQL Server.

Azure SQL Database ofrece un 99,995 % de tiempo de actividad, en comparación con Azure SQL Managed Instance que ofrece un 99,99 % de tiempo de actividad.

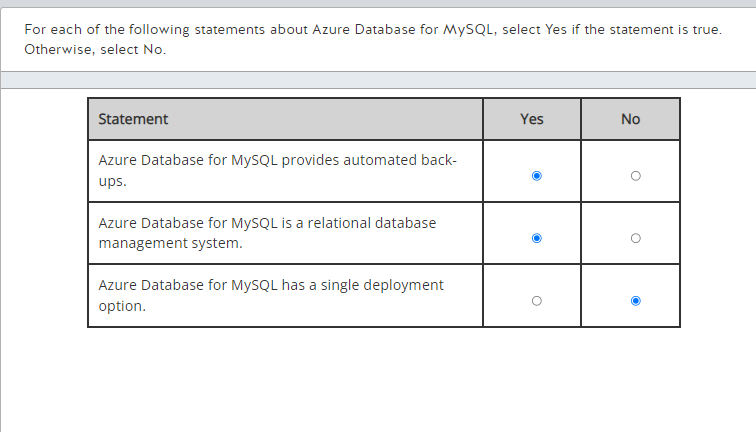
Azure SQL Database no es totalmente compatible con las instalaciones locales de SQL Server. Algunas características están disponibles en SQL Server local que no están disponibles en Azure SQL Database. Azure SQL Managed Instance ofrece casi un 100 % de compatibilidad, ya que tiene opciones de configuración adicionales, mientras que SQL Server en Azure Virtual Machines es totalmente compatible, ya que permite el nivel más granular de configuración de usuario.

Azure SQL Database no permite un control total sobre el motor de SQL Server. Azure SQL Database realiza algunas tareas para el usuario, como realizar copias de seguridad e instalar parches, pero solo SQL Server en Azure Virtual Machines permite un control total sobre el motor. Esto se debe a que SQL Server en Azure Virtual Machines es un servicio de infraestructura como servicio (IaaS), que requiere que los usuarios administren los sistemas operativos.

Azure SQL Database no es compatible con el Correo electrónico de base de datos. Esta función, que permite que la base de datos envíe mensajes de correo electrónico a los usuarios, no está disponible en Azure SQL Database. Sin embargo, se ofrece en Azure SQL Managed Instance y SQL Server en Azure Virtual Machines.

**References**[What is Azure SQL?](https://docs.microsoft.com/en-us/azure/azure-sql/azure-sql-iaas-vs-paas-what-is-overview?view=azuresql)  
  
[Features comparison: Azure SQL Database and Azure SQL Managed Instance](https://docs.microsoft.com/en-us/azure/azure-sql/database/features-comparison?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)

1. D



Azure Database for MySQL provides automated back-ups. Azure Database for MySQL is a platform as a service (PaaS) service. It offers full back-up, incremental back-up, and point-in-time database restore functions.  
  
Azure Database for MySQL is a relational database management system. MySQL is an open-source relational database management system. The Azure Database for MySQL service is a PaaS implementation of MySQL on the Azure cloud.  
  
Azure Database for MySQL supports two deployment options: single server, which has minimal required configuration, handles most database functions, such as back-ups, for the user and is best suited for existing applications, and flexible server, which allows more granular control and is best suited for new deployments.

**Traducción**

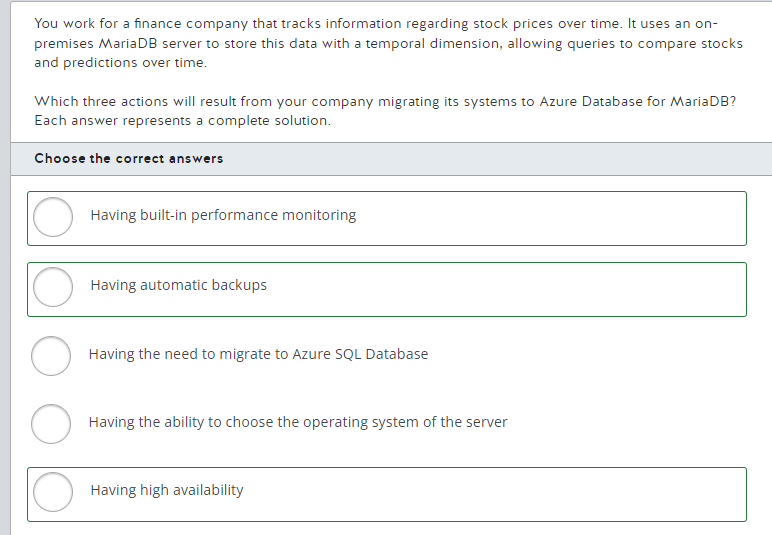
Azure Database for MySQL proporciona copias de seguridad automatizadas. Azure Database for MySQL es un servicio de plataforma como servicio (PaaS). Ofrece funciones de copia de seguridad completa, copia de seguridad incremental y restauración de bases de datos en un punto en el tiempo.

Azure Database for MySQL es un sistema de gestión de bases de datos relacionales. MySQL es un sistema de gestión de bases de datos relacionales de código abierto. El servicio Azure Database for MySQL es una implementación PaaS de MySQL en la nube de Azure.

Azure Database for MySQL admite dos opciones de implementación: servidor único, que tiene una configuración mínima requerida, maneja la mayoría de las funciones de la base de datos, como copias de seguridad, para el usuario y es más adecuado para las aplicaciones existentes, y servidor flexible, que permite un control más granular y es el más adecuado para nuevas implementaciones.

**References**[Choose the right MySQL Server option in Azure](https://docs.microsoft.com/en-us/azure/mysql/select-right-deployment-type)  
  
[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)

1. D



Migrating the company's systems to Azure Database for MariaDB will result in:

* Having high availability - Azure Database for MariaDB guarantees 99.99% up-time for its service, which is often called four nines.
* Having automatic backups -  Azure Database for MariaDB provides automatic backups and enables administrators to enact a point-in-time restore for up to 35 days.
* Having built-in performance monitoring - Azure Database for MariaDB provides various metrics and logs to help administrators understand how their database performs. They can also set up alerts based on these metrics if an item needs their attention.

Azure Database for MariaDB is a Platform as a Service (PaaS) service, meaning that Microsoft, and not the user, is responsible for the infrastructure of the virtual hardware it runs on (including the operating systems).  
  
Azure Database for MariaDB will allow the company to continue using MariaDB, but as a service. It will not need to adapt its systems to run on Azure SQL Database or perform a migration.

**Traducción**

La migración de los sistemas de la empresa a Azure Database for MariaDB dará como resultado:

• Tener alta disponibilidad: Azure Database for MariaDB garantiza un tiempo de actividad del 99,99 % para su servicio, que a menudo se denomina cuatro nueves.

• Disponer de copias de seguridad automáticas: Azure Database for MariaDB proporciona copias de seguridad automáticas y permite a los administradores realizar una restauración en un momento dado durante un máximo de 35 días.

• Supervisión del rendimiento integrada: Azure Database for MariaDB proporciona varias métricas y registros para ayudar a los administradores a comprender el rendimiento de su base de datos. También pueden configurar alertas basadas en estas métricas si un elemento necesita su atención.

Azure Database for MariaDB es un servicio de plataforma como servicio (PaaS), lo que significa que Microsoft, y no el usuario, es responsable de la infraestructura del hardware virtual en el que se ejecuta (incluidos los sistemas operativos).

Azure Database for MariaDB permitirá a la empresa seguir usando MariaDB, pero como un servicio. No necesitará adaptar sus sistemas para ejecutarse en Azure SQL Database o realizar una migración.

**References**  
[Shared responsibility in the cloud](https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility)  
  
[What is Azure Database for MariaDB?](https://docs.microsoft.com/en-us/azure/mariadb/overview)  
  
[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)

1. D

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.

**Traducción**

Azure Blob es la única opción de almacenamiento de Azure que admite niveles de acceso. El valor predeterminado es el nivel Popular, que está diseñado para datos de acceso frecuente. El nivel Cool está optimizado para datos que se almacenarán durante al menos 30 días. El nivel Cool tiene costos de almacenamiento más bajos que el nivel Hot pero costos más altos para el acceso anticipado. El nivel de archivo está diseñado para datos a los que rara vez se accede y permanecerán almacenados durante al menos 180 días. El acceso a los datos del nivel de Archivo requiere que los datos se rehidraten a un nivel Caliente o Frío. Esto puede significar una latencia de varias horas. La compatibilidad con el nivel de acceso requiere Data Lake Storage Gen2.

Azure Blob también admite dos niveles de rendimiento. El nivel de rendimiento Estándar proporciona un alto rendimiento utilizand--o medios de almacenamiento basados ​​en disco duro. El rendimiento Premium proporciona un mayor rendimiento que el nivel Estándar y utiliza medios de unidad de estado sólido (SSD). Los niveles Estándar y Premium también son compatibles con otras opciones de almacenamiento, incluido Azure File Storage y Azure SQL Database.

Azure Table y Azure File no admiten niveles de acceso. Los niveles de acceso son una característica admitida a través de Cosmos DB. Azure Table y Azure File son tipos de almacenamiento distintos y no se implementan a través de las API de Cosmos DB. El almacenamiento de tablas se utiliza para almacenar datos estructurados y no relacionales. El almacenamiento de archivos proporciona almacenamiento de archivos con acceso compartido, similar a un servidor de archivos.

**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)

1. D

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.

**Traducción**

Azure File Storage admite el montaje directo de Windows, macOS y Linux. Esto incluye soporte para acceso simultáneo desde la nube y en las instalaciones. El almacenamiento de archivos de Azure se puede usar para complementar o reemplazar los recursos compartidos del servidor de archivos local.

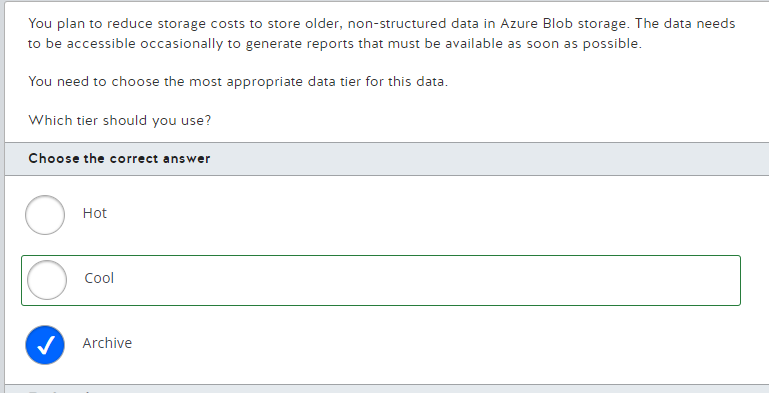
Azure File Storage no le permite seleccionar el hardware y el sistema operativo subyacentes. Azure File Storage se implementa como un servicio de archivos sin servidor en el que no tiene acceso directo ni responsabilidades administrativas sobre la arquitectura subyacente. La única opción de infraestructura que puede elegir es entre recursos compartidos de archivos estándar de disco duro (HDD) y recursos compartidos de archivos premium de disco de estado sólido (SSD).

Azure File Storage no es la solución de almacenamiento recomendada para la implementación de almacenamiento de clave/valor. Microsoft recomienda la API de Azure Cosmos DB Core (SQL) para los nuevos requisitos de clave/valor. El almacenamiento de clave/valor también es compatible con Azure Table Storage y Cosmos DB Table API.

Azure File Storage no admite la redundancia en varias regiones de forma predeterminada. Los recursos compartidos de archivos estándar admiten almacenamiento con redundancia local (LRS) de forma predeterminada con opciones de almacenamiento con redundancia de zona (ZRS), almacenamiento con redundancia geográfica (GRS) y almacenamiento con redundancia de zona geográfica (GZRS). La replicación en varias regiones se admite como opción, pero no como configuración predeterminada. La función de uso compartido de archivos grandes y los recursos compartidos de archivos premium solo admiten LRS y ZRS.

**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)

1. D



You should use the cool tier. This tier is optimized for storing data that is infrequently accessed, with a lower storage cost than the hot tier.  
  
You should not use the hot tier. This tier is optimized for frequently accessed data and has the highest storage cost among the storage tiers.  
  
You should not use the archive tier. This tier is optimized for storing data that is rarely accessed, with the lowest storage cost among the storage tiers. The archive tier data is saved in offline storage, requiring you to wait for the rehydration of the data to an online tier before you can access the data.

**Traducción**

Deberías usar el nivel genial. Este nivel está optimizado para almacenar datos a los que se accede con poca frecuencia, con un costo de almacenamiento más bajo que el nivel activo.

No debe usar el nivel caliente. Este nivel está optimizado para datos de acceso frecuente y tiene el costo de almacenamiento más alto entre los niveles de almacenamiento.

No debe usar el nivel de archivo. Este nivel está optimizado para almacenar datos a los que rara vez se accede, con el costo de almacenamiento más bajo entre los niveles de almacenamiento. Los datos del nivel de archivo se guardan en un almacenamiento fuera de línea, lo que requiere que espere la rehidratación de los datos a un nivel en línea antes de poder acceder a los datos.

**References**  
  
[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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Azure File storage. Azure File storage provides file shares compatible with the Server Message Block (SMB) protocol, replacing traditional on-premises file servers with a cloud solution.  
  
You should not use Azure Queue storage. Azure Queue storage is a service used for storing messages that are used by distributed applications. A queue message can be up to 64 KB in size.  
  
You should not use Azure Table storage. Azure Table storage is used to store data as rows and columns, forming a table in which the number of columns may vary according to each row.  
  
You should not use Azure Blob storage. Azure Blob storage can store unstructured data, such as binary objects, images, media files, and large text files, in a cost-efficient and scalable manner.

**Traducción**

Debe utilizar el almacenamiento de archivos de Azure. Azure File Storage proporciona recursos compartidos de archivos compatibles con el protocolo Server Message Block (SMB), reemplazando los servidores de archivos locales tradicionales con una solución en la nube.

No debe usar el almacenamiento en cola de Azure. Azure Queue Storage es un servicio que se usa para almacenar mensajes que usan las aplicaciones distribuidas. Un mensaje de cola puede tener un tamaño de hasta 64 KB.

No debe usar Azure Table Storage. Azure Table Storage se utiliza para almacenar datos en forma de filas y columnas, formando una tabla en la que el número de columnas puede variar según cada fila.

No debe usar Azure Blob Storage. Azure Blob Storage puede almacenar datos no estructurados, como objetos binarios, imágenes, archivos multimedia y archivos de texto de gran tamaño, de manera rentable y escalable.

**References**  
  
[What is Azure Files?](https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction)  
  
[Introduction to Azure Blob storage](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction)[What is Azure Queue Storage?](https://docs.microsoft.com/en-us/azure/storage/queues/storage-queues-introduction)  
  
[What is Azure Table storage ?](https://docs.microsoft.com/en-us/azure/storage/tables/table-storage-overview)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

The two elements that compose a key in Azure Table storage are the partition key and the row key. Data stored in Azure Table storage is referred to as rows and columns, and it forms a table in which the columns may vary according to each row. The rows in a table are split into partitions, and related rows are grouped based on a common property. This common property is called a partition key. The partition key identifies the partition inside the Azure Table storage and a row key is used to uniquely identify each row in a given partition.  
  
The table name does not compose a key in Azure Table storage. The table name is an identifier in the Azure Storage account used to store a set of data in the key/value format.  
  
The value does not compose a key in Azure Table storage. The value represents the other properties related to a given key. This is the data that is returned when you query a given key.  
  
The timestamp does not compose a key in Azure Table storage. The timestamp is a property used to record the time an entity was last modified. The timestamp is used internally by the Azure Table storage to provide optimistic concurrency.

**Traducción**

Los dos elementos que componen una clave en Azure Table Storage son la clave de partición y la clave de fila. Los datos almacenados en Azure Table Storage se denominan filas y columnas, y forman una tabla en la que las columnas pueden variar según cada fila. Las filas de una tabla se dividen en particiones y las filas relacionadas se agrupan en función de una propiedad común. Esta propiedad común se denomina clave de partición. La clave de partición identifica la partición dentro del almacenamiento de tablas de Azure y se usa una clave de fila para identificar de forma única cada fila en una partición determinada.

El nombre de la tabla no constituye una clave en Azure Table Storage. El nombre de la tabla es un identificador en la cuenta de Azure Storage que se usa para almacenar un conjunto de datos en el formato de clave/valor.

El valor no compone una clave en Azure Table Storage. El valor representa las otras propiedades relacionadas con una clave dada. Estos son los datos que se devuelven cuando consulta una clave determinada.

La marca de tiempo no compone una clave en Azure Table Storage. La marca de tiempo es una propiedad que se utiliza para registrar la hora en que se modificó por última vez una entidad. Azure Table Storage usa internamente la marca de tiempo para proporcionar una simultaneidad optimista.

**References**  
  
[Explore Azure Tables](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/5-azure-tables)  
  
[Understanding the Table service data model](https://docs.microsoft.com/en-us/rest/api/storageservices/Understanding-the-Table-Service-Data-Model)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

In Azure Table storage, a group of columns is not stored in different partitions. The rows in a table are split into partitions, which group together related rows based on a common property. This common property is called the partition key.  
  
The number of columns in each row may not be exactly the same. Azure Table lets you store semi-structured data. Unlike in a relational table, each row can have different columns of data.  
  
In Azure Table storage, data is stored as rows and columns, forming a table in which the number of columns may vary according to each row.

**Traducción**

En Azure Table Storage, un grupo de columnas no se almacena en diferentes particiones. Las filas de una tabla se dividen en particiones, que agrupan las filas relacionadas en función de una propiedad común. Esta propiedad común se denomina clave de partición.

El número de columnas en cada fila puede no ser exactamente el mismo. Azure Table le permite almacenar datos semiestructurados. A diferencia de una tabla relacional, cada fila puede tener diferentes columnas de datos.

En Azure Table Storage, los datos se almacenan en filas y columnas, formando una tabla en la que el número de columnas puede variar según cada fila.

**References**  
  
[Explore Azure Tables](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/5-azure-tables)  
  
[Understanding the Table service data model](https://docs.microsoft.com/en-us/rest/api/storageservices/Understanding-the-Table-Service-Data-Model)

1. D

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.

**Traducción**

Debe usar Azure Blob Storage para almacenar videos de capacitación. Azure Blob Storage le permite almacenar archivos de objetos de gran tamaño, como imágenes, videos y máquinas virtuales (VM).

No debe usar Azure Table Storage para almacenar videos de capacitación. Azure Table Storage le permite almacenar datos semiestructurados en formato clave/valor. Esto significa que almacena datos en un formato de filas y columnas, pero a diferencia de una base de datos relacional, cada fila tiene una clave y cada columna contiene el valor de datos completo.

No debe usar Azure File Storage para almacenar videos de capacitación. Azure File Storage le permite crear recursos compartidos de archivos en la nube, a los que pueden acceder los usuarios de la red.

No debe usar la base de datos Azure SQL para almacenar videos de capacitación. La base de datos Azure SQL le permite almacenar datos relacionales en una base de datos en la nube.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

To implement folder and directory level security in Azure Storage, you need to enable hierarchical namespace. Enabling hierarchical namespace allows you to organize your blob containers in folders and directories, allowing you to define POSIX-compatible permissions and role-based access control (RBAC) in your container.  
  
You should not use the cool access tier as default or use premium performance. You can use the cool access tier to reduce storage costs, and premium performance to improve data access performance. These features are not related to folder or directory access control in Azure Storage.

**Traducción**

Para implementar la seguridad a nivel de carpetas y directorios en Azure Storage, debe habilitar el espacio de nombres jerárquico. Habilitar el espacio de nombres jerárquico le permite organizar sus contenedores de blobs en carpetas y directorios, lo que le permite definir permisos compatibles con POSIX y control de acceso basado en roles (RBAC) en su contenedor.

No debe usar el nivel de acceso fresco como predeterminado ni usar el rendimiento superior. Puede usar el nivel de acceso esporádico para reducir los costos de almacenamiento y el rendimiento superior para mejorar el rendimiento del acceso a los datos. Estas características no están relacionadas con el control de acceso a carpetas o directorios en Azure Storage.

**References**  
  
[Access control lists (ACLs) in Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control)  
  
[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)  
  
[Storage account overview](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-performance-tiers)

1. D

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.

**Traducción**

Debe seleccionar Azure File Sync. El servicio Azure File Sync conecta un servidor local con copias en caché de archivos con datos de Azure File Storage. Esto permitirá a los usuarios acceder a los archivos a través de las copias en caché dentro de su red local, lo que mejora el rendimiento.

No debe seleccionar AzCopy. Esta utilidad permite a los usuarios cargar archivos en Azure File Storage.

No debe seleccionar la cuenta de almacenamiento de Azure. Una cuenta de almacenamiento es un grupo compartido de almacenamiento. Puede usarlo para crear recursos compartidos de archivos de Azure o contenedores de Blob Storage.

No debe seleccionar la autenticación AD de Azure Files. Esto permite a los usuarios autenticarse a través del inicio de sesión único (SSO) cuando acceden a un recurso compartido de archivos de Azure y a un directorio, tal como lo harían con las aplicaciones de Office 365. No habría ninguna mejora en el rendimiento del servicio de archivos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should recommend a container. Containers act like folders in a file system under your Blob Storage account. They are used to logically group related blob files together. For instance, those with different extensions. You can create an unlimited number of containers, and each container can store an unlimited number of blobs.  
  
You should not recommend a storage account. Storage accounts are your unique namespace in Azure, containers live under them to logically group different types of files together.  
  
You should not recommend a blob. Blobs (or binary large objects) are individual files. Blobs are added to containers, which act like directories under your storage account.  
  
You should not recommend an archive tier. The archive tier is one of the three tiers (hot, cool, and archive) which governs how quickly you can retrieve blobs. The archive tier is the slowest retrieval (and therefore the cheapest), which is intended for blobs that you will not need to access frequently.

**Traducción**

Deberías recomendar un contenedor. Los contenedores actúan como carpetas en un sistema de archivos bajo su cuenta de Blob Storage. Se utilizan para agrupar lógicamente archivos blob relacionados. Por ejemplo, aquellos con diferentes extensiones. Puede crear una cantidad ilimitada de contenedores y cada contenedor puede almacenar una cantidad ilimitada de blobs.

No debe recomendar una cuenta de almacenamiento. Las cuentas de almacenamiento son su espacio de nombres único en Azure, los contenedores viven debajo de ellas para agrupar lógicamente diferentes tipos de archivos.

No deberías recomendar un blob. Los blobs (u objetos binarios grandes) son archivos individuales. Los blobs se agregan a los contenedores, que actúan como directorios en su cuenta de almacenamiento.

No debe recomendar un nivel de archivo. El nivel de archivo es uno de los tres niveles (interactivo, esporádico y de archivo) que determina la rapidez con la que puede recuperar blobs. El nivel de archivo es la recuperación más lenta (y, por lo tanto, la más económica), y está diseñado para blobs a los que no necesitará acceder con frecuencia.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

Microsoft recomienda que cualquier nuevo proyecto de datos creado desde cero use la API Core (SQL). La API Core (SQL) se utiliza para almacenar datos semiestructurados en un documento y utiliza un lenguaje de consulta similar a SQL para manipular los datos almacenados en los documentos. Esto incluye aplicaciones con datos clave/valor. Este tipo de aplicación también es compatible con Table API, pero Core (SQL) API se recomienda como la mejor solución porque proporciona una indexación mejorada y una experiencia de consulta más rica.

Al crear una nueva aplicación que analiza información detallada de relaciones para datos no relacionales, debe usar la API de Gremlin. Este es uno de los pocos casos en los que la API Core (SQL) no se recomienda como la mejor solución. La declaración describía una base de datos de gráficos a la que se adapta específicamente la API de Gremlin.

Al mover datos de aplicaciones a la nube que usan documentos semiestructurados para almacenar datos, debe usar la API Core (SQL). La API Core (SQL) le brinda la capacidad de crear, consultar y actualizar documentos de datos.

Debe usar la API de Cassandra cuando mueva datos de formato de familia de columnas a la nube para admitir una aplicación existente. Microsoft sugiere limitar el uso de Cassandra API para admitir datos existentes, como cuando se mueven datos a la nube. La API de Cassandra está diseñada específicamente para admitir datos de familias de columnas.

Ninguna de las aplicaciones de datos debe usar Azure Table API o MongoDB API. Microsoft recomienda que solo se use cuando se admite una aplicación de datos existente, como cuando se mueve un almacén de tablas de Azure o MongoDB a 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)

1. D

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

An Azure Cosmos account is topmost in the resource hierarchy. You must have this before you can create an Azure Cosmos DB database instance. You can then create API-specific containers, such as tables, collections, or graphs. You create items, the entities for which you are storing data, inside a container. Examples include documents, nodes, edges, or rows.  
  
There is no blob in the Cosmos DB storage hierarchy. Azure Blob is another storage option in Azure. A storage blob has its own hierarchy of a storage account, container (called a container), and then a blob. Items being stored are contained in the blob, adding a fifth layer to that hierarchy.

**Traducción**

Una cuenta de Azure Cosmos ocupa el primer lugar en la jerarquía de recursos. Debe tener esto antes de poder crear una instancia de base de datos de Azure Cosmos DB. A continuación, puede crear contenedores específicos de API, como tablas, colecciones o gráficos. Crea elementos, las entidades para las que está almacenando datos, dentro de un contenedor. Los ejemplos incluyen documentos, nodos, bordes o filas.

No hay ningún blob en la jerarquía de almacenamiento de Cosmos DB. Azure Blob es otra opción de almacenamiento en Azure. Un blob de almacenamiento tiene su propia jerarquía de una cuenta de almacenamiento, un contenedor (denominado contenedor) y luego un blob. Los elementos que se almacenan están contenidos en el blob, lo que agrega una quinta capa a esa jerarquía.

**References**  
  
[Quickstart: Azure Cosmos DB SQL API client library for .NET](https://docs.microsoft.com/en-us/azure/cosmos-db/create-sql-api-dotnet)  
  
[Welcome to Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

You should use the Cassandra API to store columnar data. This API is compatible with Apache Cassandra databases, which are column-family databases used to store columnar data consisting of row identifiers and a group of information stored in a column. Each group of information is stored in independent data structures named keyspaces.  
  
You should use the Gremlin API to store graph data. This API is compatible with Gremlin, which is a graph database that stores nodes and edges used for complex relationships among entities.  
  
You should use the Core API to store JSON documents. This API is used for document-based databases that store semi-structured data in JSON format. A document usually contains all data from an entity, and each document can have different fields of data.  
  
You should not use the Table API. You can use the Table API to store key-value data organized as rows and columns, forming a table in which the number of columns may vary according to each row. This API is compatible with Azure Table storage.

**Traducción**

Debe utilizar la API de Cassandra para almacenar datos en columnas. Esta API es compatible con las bases de datos de Apache Cassandra, que son bases de datos de familias de columnas que se utilizan para almacenar datos en columnas que consisten en identificadores de fila y un grupo de información almacenada en una columna. Cada grupo de información se almacena en estructuras de datos independientes denominadas espacios de claves.

Debe usar la API de Gremlin para almacenar datos de gráficos. Esta API es compatible con Gremlin, que es una base de datos de gráficos que almacena nodos y bordes utilizados para relaciones complejas entre entidades.

Debe utilizar la API principal para almacenar documentos JSON. Esta API se utiliza para bases de datos basadas en documentos que almacenan datos semiestructurados en formato JSON. Un documento normalmente contiene todos los datos de una entidad, y cada documento puede tener diferentes campos de datos.

No debe usar Table API. Puede utilizar Table API para almacenar datos clave-valor organizados en filas y columnas, formando una tabla en la que el número de columnas puede variar según cada fila. Esta API es compatible con Azure Table Storage.

**References**  
  
[Identify Azure Cosmos DB APIs](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/3-cosmos-db-apis)  
  
[Choose an API in Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/choose-api)  
  
[Introduction to the Azure Cosmos DB Cassandra API](https://docs.microsoft.com/en-us/azure/cosmos-db/cassandra-introduction)  
  
[Introduction to Gremlin API in Azure Cosmos DB](https://docs.microsoft.com/en-us/azure/cosmos-db/graph-introduction)  
  
[Introduction to Azure Cosmos DB: Table API](https://docs.microsoft.com/en-us/azure/cosmos-db/table-introduction)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Cosmos DB Table API supports multi-region writes and read replicas. You can configure read replicas in a Cosmos DB account to multiple regions, including support to create multi-region writes.  
  
Azure Table storage supports multi-region reads replicas only. You can configure read replicas in Azure Table storage by configuring the storage account to use Read-access geo-redundant storage (RA-GRS) redundancy. This enables a readable replica in a secondary region. However, you cannot write data in the secondary region.

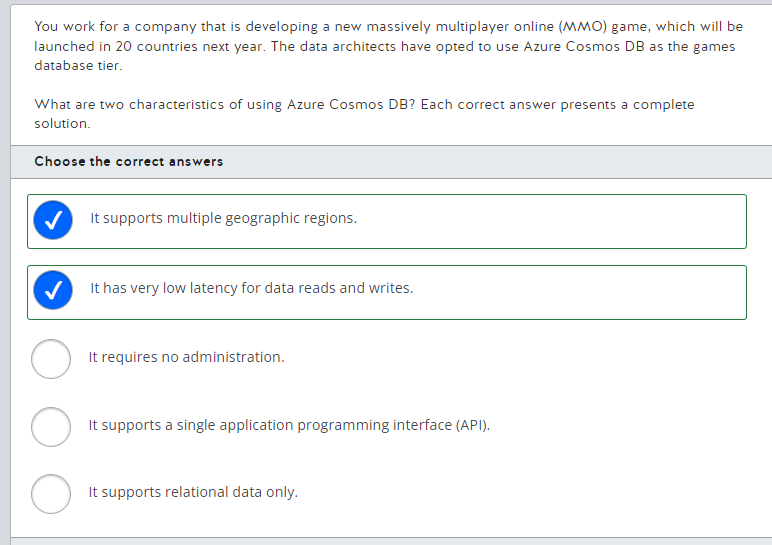
**Traducción**

Cosmos DB Table API admite réplicas de escritura y lectura en varias regiones. Puede configurar réplicas de lectura en una cuenta de Cosmos DB para varias regiones, incluida la compatibilidad para crear escrituras en varias regiones.

Azure Table Storage solo admite réplicas de lectura de varias regiones. Puede configurar réplicas de lectura en Azure Table Storage configurando la cuenta de almacenamiento para usar la redundancia de almacenamiento con redundancia geográfica con acceso de lectura (RA-GRS). Esto habilita una réplica legible en una región secundaria. Sin embargo, no puede escribir datos en la región secundaria.

**References**  
  
[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)  
  
[Introduction to Azure Cosmos DB: Table API](https://docs.microsoft.com/en-us/azure/cosmos-db/table-introduction)  
  
[Azure Storage redundancy](https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy)

1. D



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.

**Traducción**

Azure Cosmos DB tiene una latencia muy baja para lecturas y escrituras de datos, y es muy rápido y elásticamente escalable. Admitirá las lecturas y escrituras de datos muy rápidas requeridas para un juego en línea. Halo 5: Guardians usa Azure Cosmos DB como su nivel de base de datos.

Azure Cosmos DB admite varias regiones geográficas. Puede habilitar las escrituras en varias regiones, agregando las regiones de Azure de su elección a su cuenta de Cosmos DB para que los usuarios distribuidos globalmente puedan trabajar con datos en su réplica local.

Azure Cosmos DB es una aplicación de plataforma como servicio (PaaS), por lo que los administradores de la base de datos necesitan realizar alguna administración con este modelo.

Azure Cosmos DB es una base de datos NoSQL, lo que significa que está diseñada para admitir datos semiestructurados, como los que usa un juego en línea.

Azure Cosmos DB admite varias API, como Cassandra, Gremin y 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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

La API Core (SQL) permite a los desarrolladores trabajar con datos de Cosmos DB mediante la sintaxis SQL.

Cassandra API es compatible con Apache Cassandra, una base de datos estructurada de familias de columnas. Admite la sintaxis SQL para permitir que los desarrolladores manipulen y recuperen datos.

La API de Gremlin permite a los desarrolladores trabajar con datos gráficos. Es compatible con la sintaxis gráfica.

Table API permite a los desarrolladores trabajar con datos clave-valor en Cosmos DB. Admite solicitudes basadas en un espacio de nombres, de forma similar a la recuperación de datos de Azure Blob Storage.

La API de MongoDB es compatible con MongoDB Query Language (MQL), que está orientado a objetos.

**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)

1. D

Tabla

Descripción generada automáticamente

Azure Cosmos DB allows simpler queries to retrieve data than a relational database. Azure Cosmos DB is a NoSQL database that supports storing data in a denormalized way. Denormalization reduces the number of tables within a data model because all attributes from an entity are stored together rather than in different tables. With fewer tables, data read (or retrieval) queries become less complex due to the reduced table joins needed.  
  
Azure Cosmos DB does not reduce data duplication through normalization. As a NoSQL database, Azure Cosmos DB supports the denormalization of data, which increases the need for data duplication but also reduces the complexity of the schema, and, therefore, the queries needed to manipulate it. Normalization is the process of reducing data duplication by creating additional reference tables, which are supported by relational databases.  
  
Azure Cosmos DB supports applications with a global user base. Azure Cosmos DB supports multi-region writes, allowing users to work with a replicated set of data in their own region.

**Traducción**

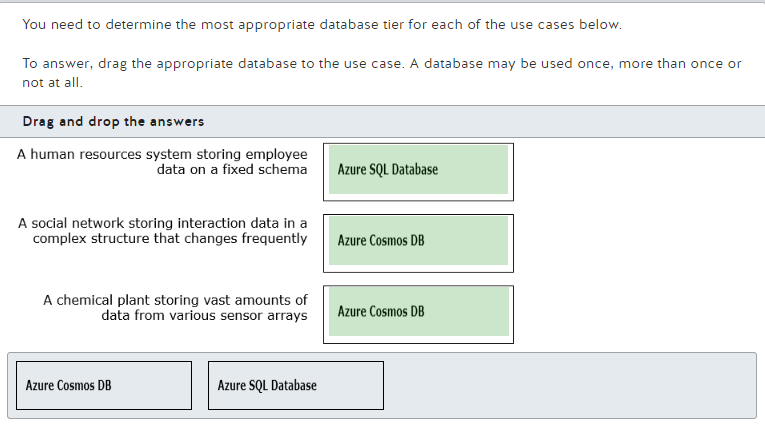
Azure Cosmos DB permite consultas más simples para recuperar datos que una base de datos relacional. Azure Cosmos DB es una base de datos NoSQL que admite el almacenamiento de datos de forma no normalizada. La desnormalización reduce el número de tablas dentro de un modelo de datos porque todos los atributos de una entidad se almacenan juntos en lugar de en tablas diferentes. Con menos tablas, las consultas de lectura (o recuperación) de datos se vuelven menos complejas debido a la necesidad de uniones de tablas reducidas.

Azure Cosmos DB no reduce la duplicación de datos mediante la normalización. Como base de datos NoSQL, Azure Cosmos DB admite la desnormalización de datos, lo que aumenta la necesidad de duplicación de datos, pero también reduce la complejidad del esquema y, por lo tanto, las consultas necesarias para manipularlo. La normalización es el proceso de reducir la duplicación de datos mediante la creación de tablas de referencia adicionales, que son compatibles con bases de datos relacionales.

Azure Cosmos DB admite aplicaciones con una base de usuarios global. Azure Cosmos DB admite escrituras en varias regiones, lo que permite a los usuarios trabajar con un conjunto de datos replicado en su propia región.

**References**  
  
[Common Azure Cosmos DB use cases](https://docs.microsoft.com/en-us/azure/cosmos-db/use-cases)  
  
[Describe Azure Cosmos DB](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/2-describe-azure-cosmos-db)

1. D



Azure SQL database is appropriate for the human resources system. As Azure SQL database uses a fixed schema, meaning the data will not change over time, a relational database like Azure SQL is a good choice.  
  
Azure Cosmos DB is appropriate for the social network. Representing complex relation structures between the entities is best stored in a NoSQL database, specifically in a graph database.  
  
Azure Cosmos DB is appropriate for the chemical plant. Azure Cosmos DB is very low-latency and highly scalable, allowing it to ingest and process huge amounts of data. This makes it perfect for IoT (internet of things) scenarios like sensor data. 

**Traducción**

La base de datos Azure SQL es adecuada para el sistema de recursos humanos. Como la base de datos Azure SQL usa un esquema fijo, lo que significa que los datos no cambiarán con el tiempo, una base de datos relacional como Azure SQL es una buena opción.

Azure Cosmos DB es apropiado para la red social. La representación de estructuras de relaciones complejas entre las entidades se almacena mejor en una base de datos NoSQL, específicamente en una base de datos de gráficos.

Azure Cosmos DB es apropiado para la planta química. Azure Cosmos DB tiene una latencia muy baja y es altamente escalable, lo que le permite ingerir y procesar grandes cantidades de datos. Esto lo hace perfecto para escenarios de IoT (Internet de las cosas), como datos de sensores.

**References**  
  
[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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

The query refers to the Gremlin API. Gremlin syntax includes functions to operate on nodes (instances of data entities) and edges (relationships between nodes), enabling users to navigate around the complex graph structure. The g. in the example query stands for graph. The example statement will retrieve person vertices in descending order of their first names.  
  
The query does not refer to the Core API, also called SQL API. SQL syntax uses keywords in capital letters, spaces, and quotes for string values. An example would be as follows:   
  
SELECT FirstName, LastName  
FROM PERSON  
WHERE IsChild = true  
  
The above query would return the first and last name columns of all rows on the person table marked as children.  
  
The query does not refer to the Table API. This API allows the manipulation of key/value pair data. An example query would be:   
  
https://<mytableendpoint>/People(PartitionKey='Harp',RowKey='Walter')  
  
The above query would retrieve an entity from the People table, by filtering on the PartitionKey Harp and the RowKey Walter.  
  
The query does not refer to the Cassandra API. Query syntax for this API looks similar to that used for the Core API.  
  
The query does not refer to the MongoDB API. This API uses an object-oriented syntax for queries. An example would be:  
  
db.people.find( { "isChild" : true })  
  
The above query would retrieve people that are children.

**Traducción**

La consulta hace referencia a la API de Gremlin. La sintaxis de Gremlin incluye funciones para operar en nodos (instancias de entidades de datos) y bordes (relaciones entre nodos), lo que permite a los usuarios navegar por la compleja estructura gráfica. el g. en la consulta de ejemplo significa gráfico. La instrucción de ejemplo recuperará los vértices de las personas en orden descendente de sus nombres.

La consulta no hace referencia a la API principal, también llamada API de SQL. La sintaxis SQL utiliza palabras clave en mayúsculas, espacios y comillas para valores de cadena. Un ejemplo sería el siguiente:

SELECCIONE Nombre, Apellido

DE PERSONA

DONDE es el niño = verdadero

La consulta anterior devolvería las columnas de nombre y apellido de todas las filas de la tabla de personas marcadas como secundarias.

La consulta no hace referencia a Table API. Esta API permite la manipulación de datos de pares clave/valor. Un ejemplo de consulta sería:

https://<mytableendpoint>/People(PartitionKey='Harp',RowKey='Walter')

La consulta anterior recuperaría una entidad de la tabla People, filtrando PartitionKey Harp y RowKey Walter.

La consulta no hace referencia a la API de Cassandra. La sintaxis de consulta para esta API es similar a la utilizada para la API principal.

La consulta no hace referencia a la API de MongoDB. Esta API utiliza una sintaxis orientada a objetos para las consultas. Un ejemplo sería:

db.personas.find( { "esNiño" : verdadero })

La consulta anterior recuperaría personas que son niños.

**References**  
  
[Identify Azure Cosmos DB APIs](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/3-cosmos-db-apis)  
  
[Query data by using Azure Cosmos DB's API for MongoDB](https://docs.microsoft.com/en-us/azure/cosmos-db/mongodb/tutorial-query-mongodb)  
  
[Tutorial: Query Azure Cosmos DB Gremlin API by using Gremlin](https://docs.microsoft.com/en-us/azure/cosmos-db/graph/tutorial-query-graph)[Tutorial: Query Azure Cosmos DB by using the Table API](https://docs.microsoft.com/en-us/azure/cosmos-db/table/tutorial-query-table)  
  
[Tutorial: Query Azure Cosmos DB by using the SQL API](https://docs.microsoft.com/en-us/azure/cosmos-db/sql/tutorial-query-sql-api)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

You should use the Azure Cosmos DB Core API or Table API to implement a Key-value store. Key-value stores are highly optimized for simple lookup scenarios. Azure Cosmos DB provides Table API for key-value store implementation. Although Core API implements a document-based database to store JSON-like semi-structured data, Microsoft recommends using the Core API for new applications, where you can use a document with an item ID and a single value field to represent a key-value store.  
  
You should use Azure Cosmos DB Cassandra API to implement a column family database. A column family database stores data in rows and columns. It is suitable to represent weather or time-series activity. Azure Cosmos DB provides Cassandra API for column family database implementation. You cannot use Cosmos DB Core API to represent a column family database.  
  
You should use Azure Cosmos DB Gremlin API to implement a graph database. A graph database stores information in the form of edges and nodes. It is used to represent complex relationships such as social interactivity. Azure Cosmos DB provides Gremlin API for graph database implementation. You cannot use Cosmos DB Core API to represent a graph database.

**Traducción**

Debe usar Azure Cosmos DB Core API o Table API para implementar un almacén de clave-valor. Los almacenes de clave-valor están altamente optimizados para escenarios de búsqueda simples. Azure Cosmos DB proporciona Table API para la implementación del almacén de clave-valor. Aunque Core API implementa una base de datos basada en documentos para almacenar datos semiestructurados similares a JSON, Microsoft recomienda usar Core API para nuevas aplicaciones, donde puede usar un documento con un ID de elemento y un campo de valor único para representar un valor clave. Tienda.

Debe usar la API Cassandra de Azure Cosmos DB para implementar una base de datos de familia de columnas. Una base de datos de familia de columnas almacena datos en filas y columnas. Es adecuado para representar el clima o la actividad de series temporales. Azure Cosmos DB proporciona Cassandra API para la implementación de bases de datos de familias de columnas. No puede usar Cosmos DB Core API para representar una base de datos de familia de columnas.

Debe usar la API Gremlin de Azure Cosmos DB para implementar una base de datos de gráficos. Una base de datos de gráficos almacena información en forma de aristas y nodos. Se utiliza para representar relaciones complejas como la interactividad social. Azure Cosmos DB proporciona la API de Gremlin para la implementación de bases de datos de gráficos. No puede usar Cosmos DB Core API para representar una base de datos de gráficos.

**References**  
  
[Explore Azure Storage for non-relational data](https://docs.microsoft.com/en-us/learn/modules/explore-provision-deploy-non-relational-data-services-azure/)  
  
[Identify Azure Cosmos DB APIs](https://docs.microsoft.com/en-us/learn/modules/explore-non-relational-data-stores-azure/3-cosmos-db-apis)  
  
[Understand data store models](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview)  
  
[Select an Azure data store for your application](https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-decision-tree)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

The most appropriate use case for Azure Synapse Analytics is to perform very complex queries and aggregations on a large amount of relational data. You can provision Synapse SQL pools to quickly execute complex queries across multiple computer nodes thanks to the Synapse SQL massively parallel processing (MPP) architecture.  
  
The most appropriate use case for Azure Synapse Analytics is not to create dashboards and data visualizations from tabular data. You should use Power BI for that instead.  
  
The most appropriate use case for Azure Synapse Analytics is not to store massive amounts of unstructured data in a hierarchical structure. You should use Azure Data Lake Storage for that instead.  
  
The most appropriate use case for Azure Synapse Analytics is not to serve as data storage for online transactional processing (OLTP) workloads. You should instead use a relational database service like Azure SQL Database.

**Traducción**

El caso de uso más apropiado para Azure Synapse Analytics es realizar consultas y agregaciones muy complejas en una gran cantidad de datos relacionales. Puede aprovisionar grupos de Synapse SQL para ejecutar rápidamente consultas complejas en varios nodos informáticos gracias a la arquitectura de procesamiento paralelo masivo (MPP) de Synapse SQL.

El caso de uso más apropiado para Azure Synapse Analytics no es crear paneles y visualizaciones de datos a partir de datos tabulares. Debería usar Power BI para eso en su lugar.

El caso de uso más apropiado para Azure Synapse Analytics no es almacenar grandes cantidades de datos no estructurados en una estructura jerárquica. En su lugar, debe usar Azure Data Lake Storage para eso.

El caso de uso más apropiado para Azure Synapse Analytics no es servir como almacenamiento de datos para cargas de trabajo de procesamiento transaccional en línea (OLTP). En su lugar, debe usar un servicio de base de datos relacional como Azure SQL Database.

**References**  
  
[What is Azure Synapse Analytics?](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is)  
  
[Dedicated SQL pool (formerly SQL DW) architecture in Azure Synapse Analytics](https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/massively-parallel-processing-mpp-architecture)  
  
[Connect to Analysis Services tabular data in Power BI Desktop](https://docs.microsoft.com/en-us/power-bi/connect-data/desktop-analysis-services-tabular-data)  
  
[Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Azure HDInsight is a big data processing service which is used to provision and manage a cluster of open-source analytics solutions such as Apache Spark, Hadoop, and Kafka.  
  
Azure Databricks is a complete platform for big data processing, streaming, and machine learning, which is optimized for the Microsoft Azure cloud services platform and built on top of Apache Spark.  
  
Azure Analysis Services is a service used to build multidimensional or tabular models used by online analytical processing (OLAP) queries. You can combine data from multiple sources, such as Azure Synapse Analytics, Azure Data Lake Store, Azure Cosmos DB, and others to build the tabular models.

**Traducción**

Azure HDInsight es un servicio de procesamiento de macrodatos que se utiliza para aprovisionar y administrar un clúster de soluciones de análisis de código abierto, como Apache Spark, Hadoop y Kafka.

Azure Databricks es una plataforma completa para el procesamiento, la transmisión y el aprendizaje automático de big data, que está optimizada para la plataforma de servicios en la nube de Microsoft Azure y construida sobre Apache Spark.

Azure Analysis Services es un servicio que se utiliza para crear modelos tabulares o multidimensionales utilizados por consultas de procesamiento analítico en línea (OLAP). Puede combinar datos de varios orígenes, como Azure Synapse Analytics, Azure Data Lake Store, Azure Cosmos DB y otros, para crear modelos tabulares.

**References**  
  
[What is Azure HDInsight?](https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-overview)  
  
[What is Azure Databricks?](https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks)  
  
[What is Azure Analysis Services?](https://docs.microsoft.com/en-us/azure/analysis-services/analysis-services-overview)  
  
[Explore analytical data stores](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/4-analytical-data-stores)

1. D

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.

**Traducción**

Azure Databricks es una plataforma de análisis basada en Apache Spark. Azure Databricks es una plataforma completa para el procesamiento, la transmisión y el aprendizaje automático de macrodatos optimizada para la plataforma de servicios en la nube de Microsoft Azure y construida sobre Apache Spark.

Azure Databricks puede manejar el procesamiento por lotes y secuencias. También puede realizar el procesamiento de datos en tiempo real y la transmisión de eventos desde Azure Event Hubs con Azure Databricks.

Azure Databricks proporciona un espacio de trabajo interactivo para la exploración y visualización de datos. Azure Databricks proporciona un espacio de trabajo para la colaboración entre científicos de datos, ingenieros de datos y analistas de negocios. Puede ejecutar cuadernos en R, Python, Scala o SQL e interactuar con los datos muy rápidamente.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Azure Data Factory is used to ingest data from data sources. You can ingest data from both relational data and non-structured data from multiple sources with Azure Data Factory.  
  
Azure Data Lake Storage is used to store raw data. You can store raw, unstructured data, such as text files, logs, and images, in order to allow these to be processed quickly at a later stage.  
  
Azure Synapse Analytics is used to model and serve data. You can load relational data ingested from Azure Data Factory in Azure Synapse Analytics using Synapse SQL pool, and you can also read unstructured data stored in Azure Data Lake Storage using Polybase. Combining both relational and unstructured data, you can perform complex analytics and serve data for later stages.  
  
Power BI is used to visualize data. You can build interactive reports and dashboards with Power BI, allowing business users to analyze this data and deliver insights throughout your organization.

**Traducción**

Azure Data Factory se usa para ingerir datos de orígenes de datos. Puede ingerir datos tanto de datos relacionales como de datos no estructurados de varios orígenes con Azure Data Factory.

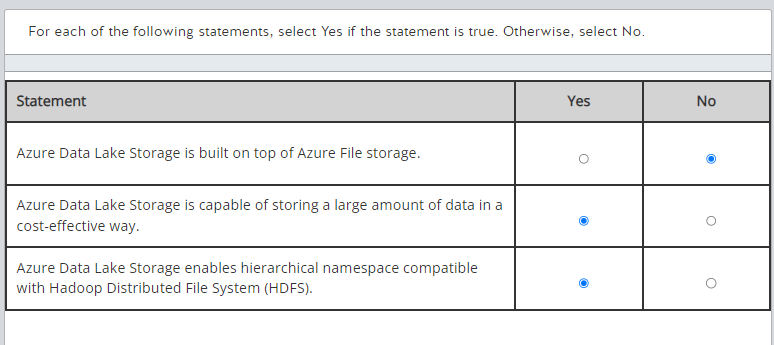
Azure Data Lake Storage se usa para almacenar datos sin procesar. Puede almacenar datos sin estructurar sin procesar, como archivos de texto, registros e imágenes, para permitir que se procesen rápidamente en una etapa posterior.

Azure Synapse Analytics se usa para modelar y servir datos. Puede cargar datos relacionales ingeridos de Azure Data Factory en Azure Synapse Analytics mediante el grupo de SQL de Synapse y también puede leer datos no estructurados almacenados en Azure Data Lake Storage mediante Polybase. Al combinar datos relacionales y no estructurados, puede realizar análisis complejos y servir datos para etapas posteriores.

Power BI se utiliza para visualizar datos. Puede crear informes y paneles interactivos con Power BI, lo que permite a los usuarios comerciales analizar estos datos y brindar información en toda su organización.

**References**  
  
[Describe data warehousing architecture](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/2-describe-warehousing)  
  
[Enterprise data warehouse](https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/modern-data-warehouse)

1. D



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.

**Traducción**

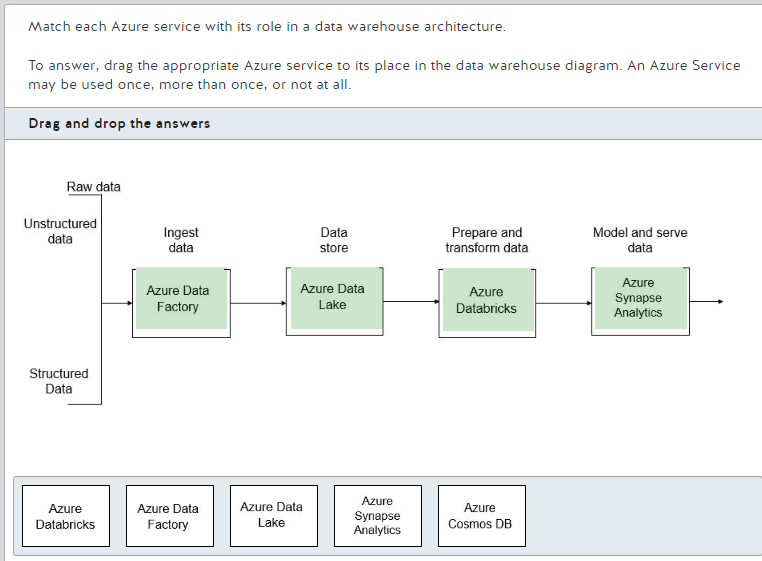
Azure Data Lake Storage no se basa en el almacenamiento de archivos de Azure. Azure Data Lake Storage Gen2 se basa en Azure Blob Storage y combina las características de la generación anterior de Azure Data Lake Storage con Azure Blob Storage.

Azure Data Lake Storage es capaz de almacenar una gran cantidad de datos de forma rentable. Azure Data Lake Storage puede almacenar grandes cantidades de datos, como cientos de terabytes y más, y solo paga por lo que usa. Puede reducir aún más el costo del almacenamiento mediante el uso de funciones como el ciclo de vida del almacenamiento para archivar o mover datos que no se usan con frecuencia a niveles de almacenamiento más económicos.

Azure Data Lake Storage habilita el espacio de nombres jerárquico compatible con el sistema de archivos distribuido de Hadoop (HDFS). Azure Data Lake Storage proporciona una capa para acceder a los datos de Azure Blob Storage como almacenamiento HDFS, incluida la compatibilidad para organizar archivos en directorios y subdirectorios, lo que le permite examinar grandes cantidades de datos rápidamente.

**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)

1. D



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.

**Traducción**

La ingestión de datos es el proceso de combinar sus datos estructurados, semiestructurados y no estructurados en un almacén de datos común. Azure Data Factory está diseñado específicamente para proporcionar compatibilidad integral para operaciones de extracción, transformación y carga (ETL) para la carga de datos del almacén de datos. A menudo, mover datos a un almacén de datos requiere una limpieza y una transformación más agresivas que las que admite Data Factory.

Los datos consolidados se almacenan en Azure Blob Storage a través de Azure Data Lake Storage. Esto le brinda un entorno de almacenamiento flexible para brindar acceso a Azure Databricks para un análisis de datos más intensivo para preparar datos limpios y transformados.

Los conectores nativos le permiten mover datos a escala desde Azure Databricks a Azure Synapse Analytics, que actúa como un centro único para sus datos estructurados. Desde aquí, los datos están disponibles para análisis e informes detallados, incluido el uso de Azure Analysis Services para brindar acceso a los datos a los usuarios finales.

Azure Cosmos DB no forma parte de la infraestructura de almacenamiento de datos de Azure como un almacén de datos no relacional.

**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)

1. D

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.

**Traducción**

Azure Data Factory puede tener varias canalizaciones. Azure Data Factory le permite crear varias canalizaciones de datos.

Pipeline es una agrupación lógica de actividades que realiza una tarea. Puede tener varias actividades en una canalización.

Las actividades en una canalización pueden ejecutarse secuencialmente u operar en paralelo. Una actividad representa un paso en una canalización.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Azure Data Factory to extract data from external sources. Azure Data Factory allows you to extract data from both relational and non-relational data stores.  
  
You should use Azure Data Lake Storage to maintain relational and NoSQL data in a data store. You can load raw data into Azure Data lake storage.  
  
You should use Azure Synapse Analytics to analyze relational and NoSQL data. You can analyze a high volume of structured and unstructured data with Azure Synapse Analytics.

**Traducción**

Debe usar Azure Data Factory para extraer datos de fuentes externas. Azure Data Factory le permite extraer datos de almacenes de datos relacionales y no relacionales.

Debe usar Azure Data Lake Storage para mantener datos relacionales y NoSQL en un almacén de datos. Puede cargar datos sin procesar en el almacenamiento de Azure Data Lake.

Debe usar Azure Synapse Analytics para analizar datos relacionales y NoSQL. Puede analizar un gran volumen de datos estructurados y no estructurados con Azure Synapse Analytics.

**References**  
  
[Describe modern data warehousing](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/2-describe-warehousing)  
  
[What is Azure Data Factory?](https://docs.microsoft.com/en-us/azure/data-factory/introduction)  
  
[Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)  
  
[What is Azure Synapse Analytics?](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

In Azure Data Factory, a pipeline is a logical grouping of activities that define tasks to perform on your data.  
  
A dataset is a representation that maps the data structure inside a pipeline with an external source or destination. Azure Data Factory uses datasets to perform tasks defined in your pipeline activities to move or transform data.  
  
A linked service is used to provide the connection between a data store and Azure Data Factory. A linked service is associated with a dataset and is used to extract or load data in the pipeline.

**Traducción**

En Azure Data Factory, una canalización es una agrupación lógica de actividades que definen tareas para realizar en sus datos.

Un conjunto de datos es una representación que mapea la estructura de datos dentro de una canalización con un origen o destino externo. Azure Data Factory usa conjuntos de datos para realizar tareas definidas en sus actividades de canalización para mover o transformar datos.

Se usa un servicio vinculado para proporcionar la conexión entre un almacén de datos y Azure Data Factory. Un servicio vinculado está asociado con un conjunto de datos y se usa para extraer o cargar datos en la canalización.

**References**  
  
[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)  
  
[Datasets in Azure Data Factory and Azure Synapse Analytics](https://docs.microsoft.com/en-us/azure/data-factory/concepts-datasets-linked-services)  
  
[Linked services in Azure Data Factory and Azure Synapse Analytics](https://docs.microsoft.com/en-us/azure/data-factory/concepts-linked-services)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Azure HDInsight. Azure HDInsight is a big data processing service that is used to provision and manage a cluster of open-source analytics solutions, such as Apache Spark, Hadoop, and Kafka.  
  
You should also use Azure Databricks. Azure Databricks is a complete platform for big data processing, streaming, and machine learning that is optimized for the Microsoft Azure cloud services platform and is built on top of Apache Spark.  
  
You should not use Azure Data Factory. Azure Data Factory is a service that is used to ingest both relational and non-structured data from multiple sources  
  
You should not use Azure Data Lake Storage. Azure Data Lake Storage is a storage service that is built on top of Azure Blob Storage, which is capable of storing large amounts of data in a cost-effective way, and which also provides a compatible Hadoop Distributed File System (HDFS) layer for analytics solutions.

**Traducción**

Debe usar Azure HDInsight. Azure HDInsight es un servicio de procesamiento de macrodatos que se usa para aprovisionar y administrar un clúster de soluciones de análisis de código abierto, como Apache Spark, Hadoop y Kafka.

También debe usar Azure Databricks. Azure Databricks es una plataforma completa para el procesamiento, la transmisión y el aprendizaje automático de big data que está optimizada para la plataforma de servicios en la nube de Microsoft Azure y se basa en Apache Spark.

No debe usar Azure Data Factory. Azure Data Factory es un servicio que se usa para ingerir datos relacionales y no estructurados de múltiples fuentes

No debe usar Azure Data Lake Storage. Azure Data Lake Storage es un servicio de almacenamiento que se basa en Azure Blob Storage, que es capaz de almacenar grandes cantidades de datos de manera rentable y que también proporciona una capa de Hadoop Distributed File System (HDFS) compatible para análisis soluciones

**References**  
  
[What is Azure HDInsight?](https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-overview)  
  
[What is Azure Databricks?](https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks)  
  
[What is Azure Data Factory?](https://docs.microsoft.com/en-us/azure/data-factory/introduction)  
  
[Introduction to Azure Data Lake Storage Gen2](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)

1. D

Texto, Tabla

Descripción generada automáticamente

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).

**Traducción**

Azure Data Factory puede cargar datos de Azure Blob Storage, Azure Data Lake Storage, Azure Cosmos DB y Azure Synapse Analytics. Incluso puede cargar datos de servicios fuera de Azure, como Amazon S3.

Azure Data Factory puede exportar datos a Azure Data Lake Storage, Azure Synapse Analytics y muchos otros destinos, como Azure SQL Database, Azure Blob Storage y Azure Cosmos DB.

Azure Data Factory puede ejecutar paquetes de SQL Server Integration Service (SSIS) mediante la actividad Ejecutar paquete SSIS. Para usar la actividad Ejecutar paquete SSIS, debe configurar el tiempo de ejecución de integración (IR) de Azure-SSIS.

**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)

1. D

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.

**Traducción**

La tubería está organizando un proceso ETL. ETL significa extraer, transformar, cargar. Esto significa que se realizan algunas operaciones de manipulación sobre los datos antes de llegar a su destino final. Esta operación se llama enriquecimiento de datos (se agregan datos). Dado que la geocodificación se aplica antes de que los datos lleguen al almacén de destino, este proceso es un ejemplo de ETL.

La canalización no está orquestando un proceso ELT. ELT significa extraer, cargar, transformar. Las operaciones de manipulación se realizan in situ en el sistema de destino. Los datos se cargan originalmente en el sistema de destino tal como estaban en el sistema de origen. Dado que la geocodificación se aplica antes de que los datos lleguen al almacén de destino, este proceso es ETL.

La geocodificación de los registros de los empleados es una actividad de canalización. Las actividades son operaciones que se realizan en una canalización de datos.

**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)

1. D

Interfaz de usuario gráfica

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.

**Traducción**

Los almacenes de datos son almacenes de datos relacionales donde el esquema está optimizado para análisis. Se desnormaliza para permitir consultas más rápidas y sencillas. Los esquemas de estrella y copo de nieve son ejemplos; tienen una tabla de hechos central y tablas de dimensiones orbitales (con jerarquías en el caso de copo de nieve).

Los lagos de datos son almacenes de archivos que se utilizan para el acceso de alto rendimiento. Pueden almacenar archivos que contengan datos estructurados, semiestructurados o no estructurados, o incluso una combinación de los tres.

Las casas de lago de datos son un híbrido de un almacén de datos y un lago; almacenan archivos con una capa de almacenamiento relacional abstraída, que se puede consultar mediante SQL.

**References**[Explore analytical data stores](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/4-analytical-data-stores)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Azure Data Lake Storage Gen 2 is a component of Azure Synapse Analytics. This is where data, scripts, and other items are stored within Azure Synapse Analytics. Azure Synapse Analytics is a tool that allows massive and efficient data storage, built-in machine learning, and artificial intelligence (AI) to provide data insights.  
  
Pipelines is also a component of Azure Synapse Analytics. These are used to ingest and transform data within Azure Synapse. They use the same engine as Azure Data Factory.  
  
Workspaces are a component of Azure Synapse Analytics. These are where you work within Synapse Studio.  
  
Power BI is not a component of Azure Synapse Analytics. Power BI is a Power Platform App that allows users to visualize and analyze their data to gain insights. It can connect to Azure Synapse Analytics, but it is a separate app.  
  
Azure SQL Database is not a component of Azure Synapse Analytics. Azure SQL Database is a cloud-managed relational database service. Azure Synapse Analytics uses an Azure Data Lake to store data.

**Traducción**

Azure Data Lake Storage Gen 2 es un componente de Azure Synapse Analytics. Aquí es donde se almacenan los datos, los scripts y otros elementos dentro de Azure Synapse Analytics. Azure Synapse Analytics es una herramienta que permite el almacenamiento de datos masivo y eficiente, el aprendizaje automático integrado y la inteligencia artificial (IA) para proporcionar información sobre los datos.

Pipelines también es un componente de Azure Synapse Analytics. Estos se usan para ingerir y transformar datos dentro de Azure Synapse. Usan el mismo motor que Azure Data Factory.

Los espacios de trabajo son un componente de Azure Synapse Analytics. Aquí es donde trabaja dentro de Synapse Studio.

Power BI no es un componente de Azure Synapse Analytics. Power BI es una aplicación de Power Platform que permite a los usuarios visualizar y analizar sus datos para obtener información. Puede conectarse a Azure Synapse Analytics, pero es una aplicación independiente.

Azure SQL Database no es un componente de Azure Synapse Analytics. Azure SQL Database es un servicio de base de datos relacional administrado en la nube. Azure Synapse Analytics usa Azure Data Lake para almacenar datos.

**References**[Azure SQL Database](https://azure.microsoft.com/en-us/products/azure-sql/database/)  
[Create a data-driven culture with business intelligence for all](https://powerbi.microsoft.com/en-us/)  
  
[Exercise: Explore data analytics in Azure with Azure Synapse Analytics](https://docs.microsoft.com/en-us/learn/modules/examine-components-of-modern-data-warehouse/5-exercise-azure-synapse)  
  
[What is Azure Synapse Analytics?](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

The following two applications are best suited to batch processing:

* Employee payroll processing and generating payroll checks
* Setting inventory stocking levels based on seasonal sales volume

Data for batch processing is collected over time, often from different data sources, and it is processed as a dataset that includes a range of rows or all rows in the dataset. Batch processing is designed to handle processing of large datasets. There is typically a long latency between data collection and data processing.  
  
The following two applications are best suited to stream processing:

* Reporting the number of users and bandwidth user for an online game
* Identifying detected manufacturing errors to automatically reject failing parts

Stream processing is designed for real-time or near real-time data processing, often as a data load process with minimal processing. Data must be able to stream out as quickly as it streams in for processing. Data is either processed as it is generated or in micro-batches of a very few rows with latency of no more than a few milliseconds.

**Traducción**

Las siguientes dos aplicaciones son las más adecuadas para el procesamiento por lotes:

• Procesamiento de nómina de empleados y generación de cheques de nómina

• Establecer niveles de existencias de inventario en función del volumen de ventas estacional

Los datos para el procesamiento por lotes se recopilan a lo largo del tiempo, a menudo de diferentes fuentes de datos, y se procesan como un conjunto de datos que incluye un rango de filas o todas las filas del conjunto de datos. El procesamiento por lotes está diseñado para manejar el procesamiento de grandes conjuntos de datos. Por lo general, hay una larga latencia entre la recopilación y el procesamiento de datos.

Las siguientes dos aplicaciones son las más adecuadas para el procesamiento de secuencias:

• Informar la cantidad de usuarios y el ancho de banda de un juego en línea

• Identificación de errores de fabricación detectados para rechazar automáticamente las piezas defectuosas

El procesamiento de flujo está diseñado para el procesamiento de datos en tiempo real o casi en tiempo real, a menudo como un proceso de carga de datos con un procesamiento mínimo. Los datos deben poder transmitirse tan rápido como llegan para su procesamiento. Los datos se procesan a medida que se generan o en microlotes de muy pocas filas con una latencia de no más de unos pocos milisegundos.

**References**  
  
[Explore core data concepts](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/4-describe-difference)  
  
[Choose a batch processing technology in Azure](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/batch-processing)  
  
[Choose a stream processing technology in Azure](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/stream-processing)  
  
[Real Time vs Batch Processing vs Stream Processing](https://www.bmc.com/blogs/batch-processing-stream-processing-real-time/)

1. D

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Data can be processed from multiple sources by both batch and stream processing. One primary difference is that batch processing can involve a wider variety of sources, including on-premises sources, whereas stream processing would receive data from streaming sources only, and it would most likely be with similar data. For example, a batch process used to analyze customer activity could include data from different databases and text documents in different formats. Data gathering often requires extensive transformation, and the data must then be written to a data store before analysis. Streaming could involve collecting data from multiple Internet of Things (IoT) sensors and writing them to a Table storage.  
  
For data used for transaction processing that requires immediate, consistent postings, stream processing should be used. This is because of a concern about latency. For example, you might insert a time stamp on each incoming entry or make minor formatting changes to the data.  
  
With batch processing, you can process a large quantity of data at once. You can use batch processing to process all the data in a specific dataset, including large datasets with millions or billions of records. With stream processing, you typically only have access to the most recent data, continuously receiving new data as a few records when it is created.

**Traducción**

Los datos se pueden procesar desde múltiples fuentes tanto por lotes como por secuencias. Una diferencia principal es que el procesamiento por lotes puede involucrar una variedad más amplia de fuentes, incluidas las fuentes locales, mientras que el procesamiento de transmisión recibiría datos solo de fuentes de transmisión y lo más probable es que sea con datos similares. Por ejemplo, un proceso por lotes utilizado para analizar la actividad del cliente podría incluir datos de diferentes bases de datos y documentos de texto en diferentes formatos. La recopilación de datos a menudo requiere una gran transformación, y los datos deben escribirse en un almacén de datos antes del análisis. La transmisión podría implicar la recopilación de datos de múltiples sensores de Internet de las cosas (IoT) y escribirlos en un almacenamiento de tabla.

Para los datos usados ​​para el procesamiento de transacciones que requieren publicaciones inmediatas y consistentes, se debe usar el procesamiento continuo. Esto se debe a una preocupación sobre la latencia. Por ejemplo, puede insertar una marca de tiempo en cada entrada entrante o realizar cambios menores en el formato de los datos.

Con el procesamiento por lotes, puede procesar una gran cantidad de datos a la vez. Puede usar el procesamiento por lotes para procesar todos los datos en un conjunto de datos específico, incluidos conjuntos de datos grandes con millones o miles de millones de registros. Con el procesamiento de secuencias, normalmente solo tiene acceso a los datos más recientes y recibe continuamente nuevos datos como algunos registros cuando se crean.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)  
  
[Choose a batch processing technology in Azure](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/batch-processing)  
  
[Choose a stream processing technology in Azure](https://docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/stream-processing)

1. D

Tabla

Descripción generada automáticamente

You can process stream data at the moment it is generated. Stream data should be processed continuously as quickly as it is generated, requiring a short latency measured in milliseconds.  
  
You cannot stream a large set of data at once in stream processing. Stream processing should be used to process individual or a few records of data continuously. You should use batch processing to process large datasets at once.  
  
You cannot use stream data to perform complex analytics. You can use stream processing for simple functions, aggregates, or calculations with individual sets of data. You should use batch processing to perform complex analytics.

**Traducción**

Puede procesar datos de transmisión en el momento en que se generan. Los datos de transmisión deben procesarse continuamente tan rápido como se generan, lo que requiere una latencia corta medida en milisegundos.

No puede transmitir un gran conjunto de datos a la vez en el procesamiento de transmisión. El procesamiento de flujo debe usarse para procesar registros de datos individuales o de unos pocos de forma continua. Debe utilizar el procesamiento por lotes para procesar grandes conjuntos de datos a la vez.

No puede usar datos de transmisión para realizar análisis complejos. Puede usar el procesamiento de secuencias para funciones simples, agregados o cálculos con conjuntos de datos individuales. Debe utilizar el procesamiento por lotes para realizar análisis complejos.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)  
  
[Big data architecture style](https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data)

1. D

Tabla

Descripción generada automáticamente

You can load a large set of data at once using batch processing. You should use batch processing to process large datasets at once, generally scheduling the processing to run when the systems are not being used, such as overnight, or during off-peak hours.  
  
Batch processing does not require near real-time data latency. In batch processing, data is collected over time for periodic processing, based on a schedule or the number of records collected. A long latency is acceptable in batch processing because data is collected over time.  
  
You can perform complex analytics using batch processing, such as aggregates, calculations, or checking data integrity.

**Traducción**

Puede cargar un gran conjunto de datos a la vez mediante el procesamiento por lotes. Debe utilizar el procesamiento por lotes para procesar grandes conjuntos de datos a la vez, generalmente programando el procesamiento para que se ejecute cuando los sistemas no se estén utilizando, como durante la noche o durante las horas de menor actividad.

El procesamiento por lotes no requiere una latencia de datos casi en tiempo real. En el procesamiento por lotes, los datos se recopilan a lo largo del tiempo para el procesamiento periódico, según un cronograma o la cantidad de registros recopilados. Una latencia larga es aceptable en el procesamiento por lotes porque los datos se recopilan con el tiempo.

Puede realizar análisis complejos mediante el procesamiento por lotes, como agregados, cálculos o comprobación de la integridad de los datos.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)  
  
[Big data architecture style](https://docs.microsoft.com/en-us/azure/architecture/guide/architecture-styles/big-data)

1. D

Tabla

Descripción generada automáticamente

You should use batch processing to build a monthly payroll for your company. Batch processing workloads are best suited to collecting a group of data within a scheduled time interval or when a certain amount of data has arrived.  
  
You should use stream processing to monitor temperature sensors on a production line. Streaming processing workloads are best suited to handling continuous data used by time-critical operations.  
  
You should not use stream processing to measure the weekly gas consumption in a neighborhood. This scenario is best suited for a batch processing workload.

**Traducción**

Debe utilizar el procesamiento por lotes para crear una nómina mensual para su empresa. Las cargas de trabajo de procesamiento por lotes se adaptan mejor a la recopilación de un grupo de datos dentro de un intervalo de tiempo programado o cuando ha llegado una cierta cantidad de datos.

Debe usar el procesamiento de flujo para monitorear los sensores de temperatura en una línea de producción. Las cargas de trabajo de procesamiento de transmisión se adaptan mejor al manejo de datos continuos utilizados por operaciones de tiempo crítico.

No debe utilizar el procesamiento de flujo para medir el consumo semanal de gas en un vecindario. Este escenario es el más adecuado para una carga de trabajo de procesamiento por lotes.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use stream processing. You can use streaming processing workloads to handle a continuous stream of data used by time-critical operations.  
  
You should not use batch processing. You can use batch processing workloads to collect a group of data with a scheduled time interval or when a certain amount of data has arrived that is not time-sensitive.  
  
You should not use 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.  
  
You should not use an OLAP workload. You can use OLAP workloads to organize large business databases and perform complex analyses without negatively affecting transactional systems.

**Traducción**

Debe utilizar el procesamiento de secuencias. Puede usar cargas de trabajo de procesamiento de transmisión para manejar un flujo continuo de datos utilizados por operaciones de tiempo crítico.

No debe utilizar el procesamiento por lotes. Puede usar cargas de trabajo de procesamiento por lotes para recopilar un grupo de datos con un intervalo de tiempo programado o cuando llega una cierta cantidad de datos que no son sensibles al tiempo.

No debe usar una carga de trabajo OLTP. Puede usar cargas de trabajo OLTP con sistemas transaccionales utilizados en las operaciones diarias de una organización, como sistemas contables, financieros y otros que requieren una gran coherencia para las transacciones.

No debe usar una carga de trabajo OLAP. Puede usar cargas de trabajo OLAP para organizar grandes bases de datos comerciales y realizar análisis complejos sin afectar negativamente los sistemas transaccionales.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)  
  
[Explore transactional data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/5-transactional-data-processing)  
  
[Explore analytical data processing](https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/6-analytical-processing)  
  
[Online transaction processing (OLTP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-transaction-processing)  
  
[Online analytical processing (OLAP)](https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing)

1. D

Interfaz de usuario gráfica

Descripción generada automáticamente

Batch processing involves processing a dataset on a regular schedule. It is useful for periodic processes containing large amounts of data. Streaming is event driven; data is processed as soon as something happens. It is used when data changes frequently.  
  
For the finance department, you should select batch. As processing is completed periodically (once a month) and monthly data forms a single set, this is an example of a batch process.  
  
For the insurance company, you should also select batch. Sensor data is collected while the car is offline, and then it is uploaded and processed when the car is connected to a wifi network, which is an irregular, periodic process.  
  
For the retail company, you should select streaming. Anonymous visitors are constantly using the website, each creating a micro-batch.  
  
For the delivery company, you should also select streaming. The drivers' locations change constantly, meaning that the delivery company will need a low-latency and will need to process the data in real time.

**Traducción**

El procesamiento por lotes implica procesar un conjunto de datos en un horario regular. Es útil para procesos periódicos que contienen grandes cantidades de datos. La transmisión está impulsada por eventos; los datos se procesan tan pronto como sucede algo. Se utiliza cuando los datos cambian con frecuencia.

Para el departamento de finanzas, debe seleccionar lote. Como el procesamiento se completa periódicamente (una vez al mes) y los datos mensuales forman un solo conjunto, este es un ejemplo de un proceso por lotes.

Para la compañía de seguros, también debe seleccionar lote. Los datos del sensor se recopilan mientras el automóvil está fuera de línea y luego se cargan y procesan cuando el automóvil está conectado a una red wifi, que es un proceso periódico e irregular.

Para la empresa minorista, debe seleccionar la transmisión. Los visitantes anónimos utilizan constantemente el sitio web y cada uno crea un microlote.

Para la empresa de entrega, también debe seleccionar la transmisión. Las ubicaciones de los conductores cambian constantemente, lo que significa que la empresa de entrega necesitará una latencia baja y deberá procesar los datos en tiempo real.

**References**  
  
[Understand batch and stream processing](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/2-batch-stream)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Spark Structured Streaming can be used with Azure Synapse Analytics, Azure Databricks, and Azure HDInsight. Spark Structured Streaming is an open-source library for Apache Spark-based services.  
  
Spark Structured Streaming cannot be used with Azure SQL Database or Power BI. Azure SQL Database is a relational database service, and Power BI is a Power Platform analytics tool.

**Traducción**

Spark Structured Streaming se puede usar con Azure Synapse Analytics, Azure Databricks y Azure HDInsight. Spark Structured Streaming es una biblioteca de código abierto para servicios basados en Apache Spark.

Spark Structured Streaming no se puede usar con Azure SQL Database o Power BI. Azure SQL Database es un servicio de base de datos relacional y Power BI es una herramienta de análisis de Power Platform.

**References**[Create a data-driven culture with business intelligence for all](https://powerbi.microsoft.com/en-us/)  
  
[Azure SQL Database](https://azure.microsoft.com/en-us/products/azure-sql/database/)  
[What is Apache Spark Structured Streaming?](https://docs.microsoft.com/en-us/azure/databricks/structured-streaming/)  
  
[Apache Spark in Azure Synapse Analytics](https://docs.microsoft.com/en-us/azure/synapse-analytics/spark/apache-spark-overview)  
  
[Explore common elements of stream processing architecture](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/3-explore-common-elements)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Azure Stream Analytics provides a platform as a service (PaaS) service model.  
  
Azure Stream Analytics is an event processing and analytics engine. It allows users to ingest a continuous amount of streaming data from various sources like Azure Event Hubs or IoT devices, perform operations on top of the streamed data, and finally send the output to a target like a data warehouse.  
  
An example use case would be to use machine learning to apply a predictive score of how likely a car is to need maintenance, based on information ingested from in-car sensors. The data could then be output to Power BI to produce a report alerting users to these insights.  
  
Azure Stream Analytics does not require users to provision their own infrastructure, but users will write their own software to create their own queries to analyze the streaming data.   
  
Software as a service (SaaS) gives users the least level of responsibility of all cloud services; it only requires users to control access to their data and maintain user authentication.  
  
Infrastructure as a service (IaaS) requires users to configure their network and the operating systems of services provided.  
  
On-premises systems are not cloud services; they require users to provide everything required for their solutions from the ground up, including the physical machines they will run on.

**Traducción**

Azure Stream Analytics proporciona un modelo de servicio de plataforma como servicio (PaaS).

Azure Stream Analytics es un motor de análisis y procesamiento de eventos. Permite a los usuarios ingerir una cantidad continua de datos de transmisión de varias fuentes, como Azure Event Hubs o dispositivos IoT, realizar operaciones sobre los datos transmitidos y, finalmente, enviar la salida a un destino como un almacén de datos.

Un ejemplo de caso de uso sería utilizar el aprendizaje automático para aplicar una puntuación predictiva de la probabilidad de que un automóvil necesite mantenimiento, en función de la información recopilada de los sensores del automóvil. Luego, los datos podrían enviarse a Power BI para generar un informe que alerte a los usuarios sobre estos conocimientos.

Azure Stream Analytics no requiere que los usuarios aprovisionen su propia infraestructura, pero los usuarios escribirán su propio software para crear sus propias consultas para analizar los datos de transmisión.

El software como servicio (SaaS) brinda a los usuarios el menor nivel de responsabilidad de todos los servicios en la nube; solo requiere que los usuarios controlen el acceso a sus datos y mantengan la autenticación del usuario.

La infraestructura como servicio (IaaS) requiere que los usuarios configuren su red y los sistemas operativos de los servicios prestados.

Los sistemas locales no son servicios en la nube; requieren que los usuarios proporcionen todo lo necesario para sus soluciones desde cero, incluidas las máquinas físicas en las que se ejecutarán.

**References**[Shared responsibility in the cloud](https://docs.microsoft.com/en-us/azure/security/fundamentals/shared-responsibility)  
  
[Welcome to Azure Stream Analytics](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction)

1. D

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

You should use KQL. Azure Data Explorer uses Kusto Query Language (KQL), an open-source language based on SQL.  
  
You should not use SQL. Structured Query Language (SQL) is used to query relational databases.  
  
You should not use Gremlin. Gremlin is used to query graph databases.  
  
You should not use FetchXML. FetchXML is an XML-based query language used to query Microsoft Dataverse in the Power Platform.

**Traducción**

Deberías usar KQL. Azure Data Explorer usa Kusto Query Language (KQL), un lenguaje de código abierto basado en SQL.

No debe usar SQL. El lenguaje de consulta estructurado (SQL) se utiliza para consultar bases de datos relacionales.

No deberías usar Gremlin. Gremlin se utiliza para consultar bases de datos de gráficos.

No debe usar FetchXML. FetchXML es un lenguaje de consulta basado en XML que se utiliza para consultar Microsoft Dataverse en Power Platform.

**References**[Explore Azure Data Explorer](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-stream-processing/8-data-explorer)  
  
[Kusto Query Language (KQL) overview](https://docs.microsoft.com/en-us/azure/data-explorer/kusto/query/)  
  
[What is Azure Data Explorer?](https://docs.microsoft.com/en-us/azure/data-explorer/data-explorer-overview)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

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.

**Traducción**

Debe utilizar un gráfico de barras. Los gráficos de barras son los gráficos más comunes que se utilizan para visualizar y comparar datos entre categorías. Puede determinar las categorías de productos más vendidas ordenando el campo de ventas brutas.

No debe usar un gráfico de burbujas. Los gráficos de burbujas se utilizan para visualizar tres dimensiones de datos, con los ejes x e y representando dos dimensiones de datos, y la tercera dimensión representada por el tamaño de la burbuja.

No debe usar un gráfico circular o de anillos. Estos gráficos son más adecuados cuando se comparan solo algunas categorías. Cuando tienes más de ocho categorías, la lectura y la interpretación se vuelven bastante difíciles. Los gráficos de anillos son similares a los gráficos circulares, con la diferencia de que el centro está vacío para los gráficos de anillos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Business Intelligence (BI) allows businesses to achieve better decision making by combining technologies, applications, and processes to collect, analyze, and present business information. BI combines techniques like data visualization, reporting, and applications like Power BI to help businesses make decisions based on data.  
  
Data visualization refers to the techniques used to represent information and data in a graphical way by using visual elements like charts, graphs, and other visualization resources. It provides an accessible way to spot and understand trends, outliers, and patterns in data.  
  
Reporting is the process of organizing business data into informational summaries to discover how the organization is performing.  
  
Power BI is a collection of services, apps, and connectors to combine data from a range of sources, build interactive dashboards and reports, and visualize relevant business information. You can use Power BI to create reports and implement data visualization to present business information.

**Traducción**

Business Intelligence (BI) permite a las empresas lograr una mejor toma de decisiones al combinar tecnologías, aplicaciones y procesos para recopilar, analizar y presentar información comercial. BI combina técnicas como la visualización de datos, informes y aplicaciones como Power BI para ayudar a las empresas a tomar decisiones basadas en datos.

La visualización de datos se refiere a las técnicas utilizadas para representar información y datos de forma gráfica mediante el uso de elementos visuales como tablas, gráficos y otros recursos de visualización. Proporciona una forma accesible de detectar y comprender tendencias, valores atípicos y patrones en los datos.

La generación de informes es el proceso de organizar los datos comerciales en resúmenes informativos para descubrir cómo se está desempeñando la organización.

Power BI es una colección de servicios, aplicaciones y conectores para combinar datos de una variedad de fuentes, crear paneles e informes interactivos y visualizar información comercial relevante. Puede usar Power BI para crear informes e implementar la visualización de datos para presentar información comercial.

**References**  
  
[Explore data visualization](https://docs.microsoft.com/en-us/learn/modules/explore-concepts-of-data-analytics/3-explore-data-visualization)  
  
[What is Power BI?](https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

The data model uses the star schema. The star schema is a data modeling technique used in data warehouses that consists of denormalizing the data in fact tables and dimension tables. Denomarlizing helps the data warehouse perform queries in tables with a large volume of rows more quickly.  
  
The data model does not use the snowflake schema. The snowflake schema is a variation of the star schema, where some dimension tables are not fully denormalized. In this scenario, the dimension tables are fully denormalized.  
  
The Investment table is a fact table. A fact table describes business events, represented by numeric measure columns, and linked with dimension key columns that relate to dimension tables. In a fact table, you can store information like sales orders, stock balances, or other business measures.  
  
The Account table is a dimension table. A dimension table describes business entities that are associated with a business event, and it is used to aggregate or filter those events. A dimension table represents entities like products, sales personnel, dates, and is referenced with dimension key columns in the fact tables.

**Traducción**

El modelo de datos utiliza el esquema en estrella. El esquema en estrella es una técnica de modelado de datos utilizada en almacenes de datos que consiste en desnormalizar los datos en tablas de hechos y tablas de dimensiones. La denomarlización ayuda al almacén de datos a realizar consultas en tablas con un gran volumen de filas más rápidamente.

El modelo de datos no utiliza el esquema de copo de nieve. El esquema de copo de nieve es una variación del esquema de estrella, donde algunas tablas de dimensiones no están completamente desnormalizadas. En este escenario, las tablas de dimensiones están completamente desnormalizadas.

La tabla de inversiones es una tabla de hechos. Una tabla de hechos describe eventos comerciales, representados por columnas de medidas numéricas y vinculados con columnas de clave de dimensión que se relacionan con tablas de dimensiones. En una tabla de hechos, puede almacenar información como órdenes de venta, saldos de existencias u otras medidas comerciales.

La tabla Cuenta es una tabla de dimensiones. Una tabla de dimensiones describe las entidades comerciales que están asociadas con un evento comercial y se usa para agregar o filtrar esos eventos. Una tabla de dimensiones representa entidades como productos, personal de ventas, fechas y se hace referencia a ellas con columnas de clave de dimensión en las tablas de hechos.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

El modelo de datos que se muestra en la exposición es un ejemplo del esquema en estrella. En el esquema en estrella, las tablas de hechos están directamente conectadas a las tablas de dimensiones. En este caso, las tablas Cliente y Producto son dimensiones y la tabla Pedido es un hecho.

La exposición no muestra el esquema del copo de nieve. En el esquema de copo de nieve, las dimensiones están interrelacionadas entre sí y una tabla de dimensiones se vincula con la tabla de hechos. Las tablas de dimensiones en el esquema de copo de nieve están muy normalizadas.

No debe utilizar el sistema OLAP. Los sistemas OLAP están diseñados para realizar análisis complejos y proporcionar inteligencia comercial.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

You should use Power BI service to create an app workspace and share its dashboard. Power BI service allows you to create an app workspace and share reports and dashboards. You can create a dashboard from reports in Power BI service.  
  
You should not use Power BI Desktop to create an app workspace and share its dashboard. Power BI Desktop allows you to create reports and publish them to an app workspace in the Power BI service. You can create dashboards from reports.  
  
You should not use Power BI service and Power BI Desktop to create a workspace and share its dashboard. You only need Power BI service to create a workspace and share its dashboard.  
  
You should not use Power BI mobile app to create a workspace and share its dashboard. Power BI mobile app allows you to view reports and dashboards shared with you.

**Traducción**

Debe usar el servicio Power BI para crear un espacio de trabajo de la aplicación y compartir su tablero. El servicio Power BI le permite crear un espacio de trabajo de la aplicación y compartir informes y paneles. Puede crear un panel a partir de informes en el servicio Power BI.

No debe usar Power BI Desktop para crear un área de trabajo de la aplicación y compartir su tablero. Power BI Desktop le permite crear informes y publicarlos en un área de trabajo de la aplicación en el servicio Power BI. Puede crear paneles a partir de informes.

No debe usar el servicio Power BI y Power BI Desktop para crear un espacio de trabajo y compartir su tablero. Solo necesita el servicio Power BI para crear un espacio de trabajo y compartir su tablero.

No debe usar la aplicación móvil Power BI para crear un espacio de trabajo y compartir su tablero. La aplicación móvil de Power BI le permite ver informes y paneles compartidos con usted.

**References**  
  
[Describe Power BI tools and workflow](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-data-visualization/2-power-bi)  
  
[Create a workspace in Power BI](https://docs.microsoft.com/en-us/power-bi/collaborate-share/service-create-the-new-workspaces)  
  
[Share dashboards with your organization](https://docs.microsoft.com/en-us/learn/modules/explore-data-power-bi/6-share-dashboards)

1. D

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.

**Traducción**

Debe crear un informe en Power BI Desktop. En un flujo de trabajo común, comienza conectándose a orígenes de datos y creando un informe en Power BI Desktop. También puede crear informes sobre el servicio Power BI pero con acceso limitado a las fuentes de datos.

Luego, debe compartir un informe sobre el servicio Power BI. Puede publicar y compartir informes en un área de trabajo del servicio Power BI para que estén disponibles para los usuarios finales.

Finalmente, debe ver e interactuar con los informes en Power BI móvil. Después de compartir un informe en el servicio Power BI, puede ver e interactuar con este informe mediante Power BI móvil. Puede usar el servicio Power BI para interactuar y ver informes con usuarios finales solo con acceso de escritorio.

No debe crear ni compartir un informe en Power BI móvil. Las aplicaciones móviles de Power BI solo se pueden usar para ver e interactuar con informes.

**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)

1. D

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.

**Traducción**

Puede crear informes y paneles en el servicio Power BI. Aunque puede crear informes y paneles básicos en el servicio Power BI, es más común crear informes en Power BI Desktop para una experiencia de diseño completa y para acceder a más fuentes de datos.

Puede compartir y distribuir informes en el servicio Power BI. Puede crear espacios de trabajo en el servicio Power BI para colaborar y compartir sus informes con otros miembros del equipo y su empresa.

No puede diseñar el modelado de datos en el servicio Power BI. Debería usar Power BI Desktop en su lugar para diseñar modelos de datos, como crear columnas personalizadas y administrar relaciones de modelos.

**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)

1. D

Tabla

Descripción generada automáticamente

Dashboards are created within the Power BI service, the SaaS component of Power BI.  
  
Dashboards are a single canvas. They have only one page, unlike Power BI reports, which can have multiple pages.  
  
Dashboards are composed of tiles, which can be created by pinning visuals from Power BI reports, and widgets.

**Traducción**

Los paneles se crean dentro del servicio Power BI, el componente SaaS de Power BI.

Los tableros son un solo lienzo. Tienen una sola página, a diferencia de los informes de Power BI, que pueden tener varias páginas.

Los paneles se componen de mosaicos, que se pueden crear fijando elementos visuales de los informes y widgets de Power BI.

**References**  
  
[What is Power BI?](https://docs.microsoft.com/en-us/learn/modules/introduction-power-bi/2-what-power-bi)  
  
[Describe Power BI tools and workflow](https://docs.microsoft.com/en-us/learn/modules/explore-fundamentals-data-visualization/2-power-bi)

1. D

Interfaz de usuario gráfica, Texto, Aplicación

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.

**Traducción**

Debe seleccionar Preguntas y respuestas. El visual de Preguntas y respuestas brinda a los usuarios un cuadro de pregunta (ingreso de texto) que pueden usar para hacer preguntas en lenguaje natural, así como una lista previamente completada de preguntas sugeridas y recibir respuestas visuales.

No debe seleccionar el gráfico de anillos. Un gráfico de anillos es similar a un gráfico circular; muestra la relación de las partes con un todo.

No debe seleccionar la tabla de indicadores. Estos muestran un indicador de semicírculo que muestra el progreso hacia una meta o KPI.

No debe seleccionar Diagrama de árbol. Estos se componen de rectángulos de colores y se utilizan para representar datos jerárquicos o porciones de partes hacia un todo donde hay demasiadas partes para mostrar adecuadamente en un gráfico circular o de anillos.

**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)

1. D

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.

**Traducción**

Deberías usar Tabla. Los objetos visuales de tabla le permiten mostrar datos en dos dimensiones con columnas y filas, como una hoja de cálculo.

También debe utilizar la tarjeta. Las imágenes de tarjetas muestran uno o más puntos de datos sobre una fila en particular.

No debe usar el gráfico de embudo. Los gráficos de embudo se utilizan para mostrar un proceso con etapas; por ejemplo, un canal de ventas.

No debe usar el gráfico de personas influyentes clave. El gráfico de personas influyentes clave se utiliza para ayudar a los usuarios a comprender las razones detrás de una métrica.

No debe seleccionar Mapa básico. Los mapas básicos le permiten asociar datos con un área geográfica y mostrarlos en una vista de mapa.

**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)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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.

**Traducción**

Debe utilizar el Editor de Power Query. Esta es una herramienta dentro de Power BI Desktop que le permite interactuar con los datos aplicando una serie de transformaciones que se registran como pasos, que se pueden deshacer si es necesario.

No debe usar la vista Informe. Este es el lienzo donde puede arrastrar sus elementos visuales para crear su informe utilizando datos que ya ha importado, transformado y modelado.

No debe usar Publicar en el servicio Power BI. Este botón toma el informe de Power BI que ha creado con Power BI Desktop y lo envía al servicio de Power BI para permitirle compartirlo con colegas.

No debe usar Conectar a datos. Esta función le permite conectarse a fuentes de datos sin procesar, por ejemplo, y la base de datos de Azure SQL. Este es el paso antes de transformar y dar forma a sus datos antes de crear el modelo.

**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)

1. D

Tabla

Descripción generada automáticamente

You can apply role-based security to data. Power BI service is a software as a Service (SaaS) application that allows users to share and collaborate on Power BI reports and dashboards, and also lets administrators assign users to roles for a specific report. These roles are created by the report's author on the Power BI desktop during the data modeling, allowing you to filter data in the visuals that specific users can access and interact with.  
  
You can share dashboards you create with colleagues. Power BI Service allows you to share the dashboards that you create with your colleagues. Dashboards are one-page visual data displays that allow users to interact with and collaborate on data.  
  
You do not need to install Power BI Service on your machine. Power BI service is a SaaS application that you can access via your web browser or a dedicated app. It does not need to be installed locally. Power BI Desktop (used to create Power BI reports) is an application you need to install on your local machine.

**Traducción**

Puede aplicar seguridad basada en funciones a los datos. El servicio Power BI es una aplicación de software como servicio (SaaS) que permite a los usuarios compartir y colaborar en informes y paneles de Power BI, y también permite a los administradores asignar usuarios a roles para un informe específico. Estos roles los crea el autor del informe en el escritorio de Power BI durante el modelado de datos, lo que le permite filtrar datos en las imágenes a las que los usuarios específicos pueden acceder e interactuar.

Puede compartir los tableros que cree con sus colegas. El servicio Power BI le permite compartir los paneles que crea con sus colegas. Los tableros son pantallas de datos visuales de una página que permiten a los usuarios interactuar y colaborar con los datos.

No necesita instalar el servicio Power BI en su máquina. El servicio Power BI es una aplicación SaaS a la que puede acceder a través de su navegador web o una aplicación dedicada. No es necesario instalarlo localmente. Power BI Desktop (utilizado para crear informes de Power BI) es una aplicación que necesita instalar en su máquina local.

**References**  
  
[Comparing Power BI Desktop and the Power BI service](https://docs.microsoft.com/en-us/power-bi/fundamentals/service-service-vs-desktop)  
  
[Basic concepts for designers in the Power BI service](https://docs.microsoft.com/en-us/power-bi/fundamentals/service-basic-concepts)

1. D

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Power BI data models traditionally use star schemas, like data warehouses. They are a simple, denormalized data structure, designed to allow faster queries to retrieve data and increase report performance. In a star schema, there are fact (events or transactions) and dimension tables (business entities). The fact table is the table at the center of a star schema (in this scenario, the Class table). Fact tables usually contain many rows, whereas dimension tables have few rows.

**Traducción**

Los modelos de datos de Power BI tradicionalmente usan esquemas en estrella, como almacenes de datos. Son una estructura de datos simple y desnormalizada, diseñada para permitir consultas más rápidas para recuperar datos y aumentar el rendimiento de los informes. En un esquema en estrella, hay tablas de hechos (eventos o transacciones) y dimensiones (entidades comerciales). La tabla de hechos es la tabla en el centro de un esquema en estrella (en este escenario, la tabla Clase). Las tablas de hechos suelen contener muchas filas, mientras que las tablas de dimensiones tienen pocas filas.

**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)