Question 1:
What type of data is JavaScript Object Notation (JSON) format data an example of?

- Semi-structured
- Structured
- Unstructured
- Relational

Correct Answer: Semi-structured

Explanation
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 follows:

```
{
"ID": "1",
"Name": "John Doe",
"Telephone": [
{ "Home": "1-999-9999999" },
{ "Business": "1-888-8888888" },
{ "Cell": "1-555-55555555" }
],
"Address": [
{ "Home": [
{ "StreetAddress": "121 Some Street" },
{ "City": "Some City" },
{ "State": "MO" },
{ "Zip": "63601" }
] },
{ "Business": [
{ "StreetAddress": "87 Some Building" },
{ "City": "Another City" },
{ "State": "MO" },
{ "Zip": "63121" }
] }
]
}
```

Curly brackets are used to enclose the document and any subdocuments. Azure Cosmos DB is the most common storage solution for this type of data.
Unstructured data includes files like video or audio files with no schema structure. This type of data is usually stored in Azure Blob storage.
Structured data and relational data refer to the same data structure with data in a highly normalized format and stored in multiple related tables. The most common storage solution is some type of SQL database management system.
References
Identify the need for data solutions
Understand data store models
Non-relational data and NoSQL

Question 2:
Which of the following statements are true?

A. Each document in a document database typically contains all data for a single entity.

B. Documents in a document database use the same data schema for all documents.

C. Documents in a document database support relationships enforced between documents.

- A
- A and B
- B and C
- A and C
- B
- C

Correct Answer: A

Explanation
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.
References
Understand data store models
Non-relational data and NoSQL
Identify the need for data solutions

Question 3:
Your company is writing an application that will receive real-time data with variable data fields from multiple sources. The application should be able to write the data to a data store as quickly as possible. Data will be keyed with an incremental index, and it will be treated as transparent by the application.
Which type of data store should you use?

- Column-family
- Table
- Key/value
- Graph

Correct Answer: Key/Value

Explanation

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 store, data is organized based on a partition key and a row key. The partition key identifies the partition in which the data is stored, and the row uniquely identifies the row within the partition.
You should not use a graph data store. A graph data store is designed to support extensive, complex relationships between entities. This helps to make it easier to perform complex relation analysis.

References

Understand data store models
Non-relational data and NoSQL
Identify the need for data solutions

Question 4:
You work as a data engineer.
Which two data stores types are non-relational? Each correct answer presents a complete solution.

- Graph Database
- Document Database
- SQL Database
- Azure Database for MariaDB

Correct Answer: Graph Database and Document Database

Explanation

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 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.

References

Describe types of non-relational and NoSQL databases
Understand data store models
Relational vs. NoSQL data

What is Azure Database for MariaDB?

Question 5:
Our company application requires you to maintain user preference.
Which type of non-relation data store should you use?

- Column-family database
- Document database
- Key-value store
- Graph database

Correct Answer: Key-value store