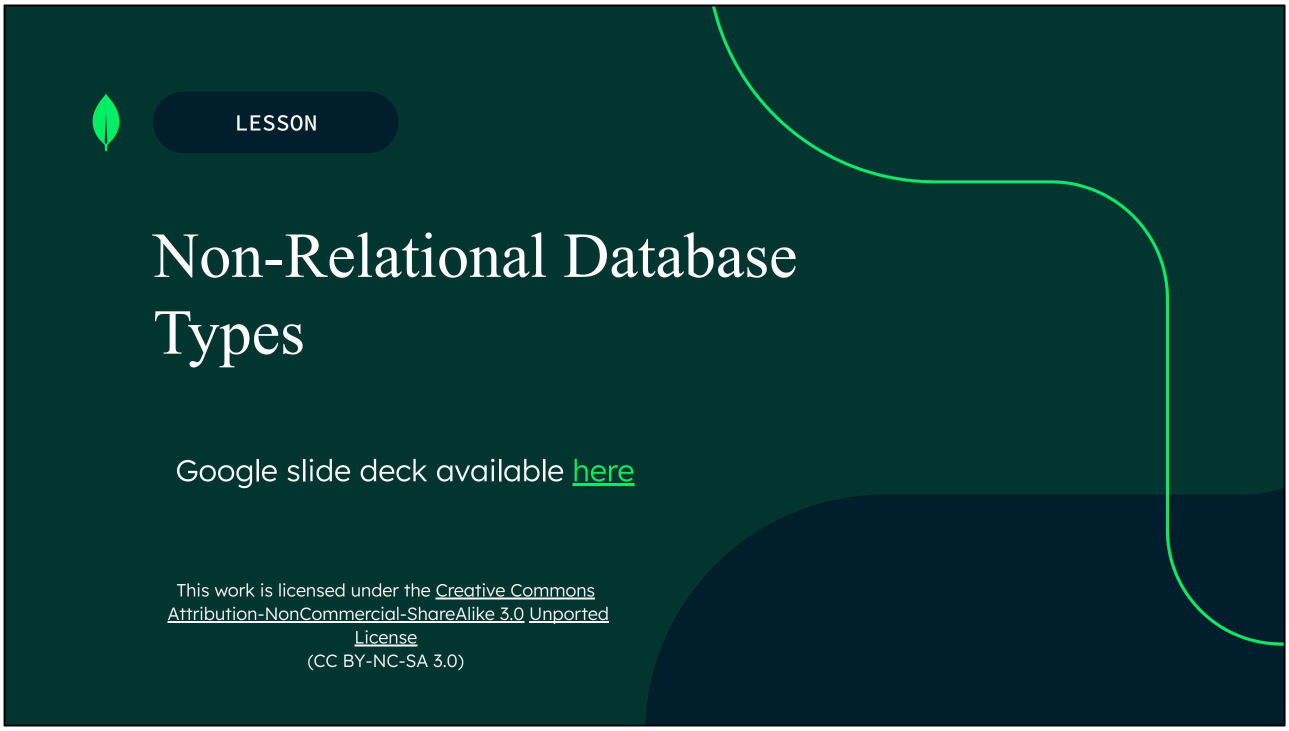
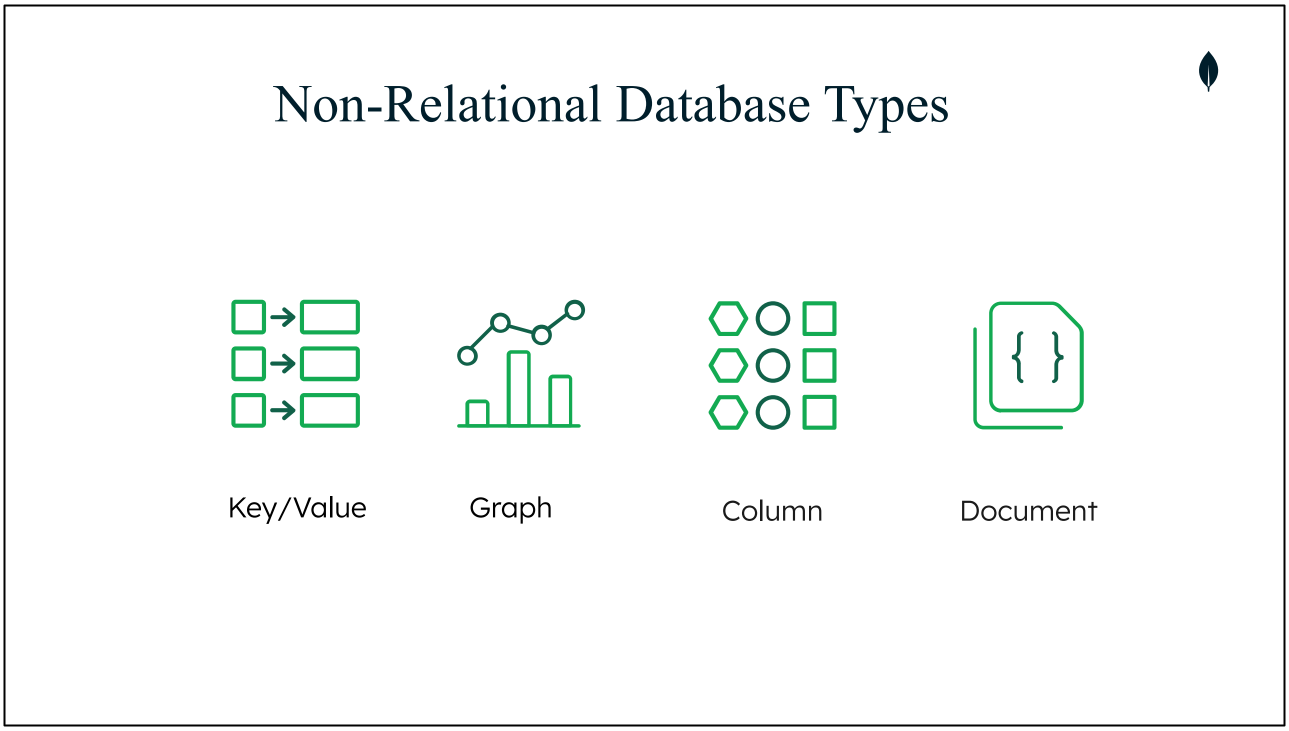


LESSON

Non-Relational Database Types

Google slide deck available here



Non-Relational Database Types

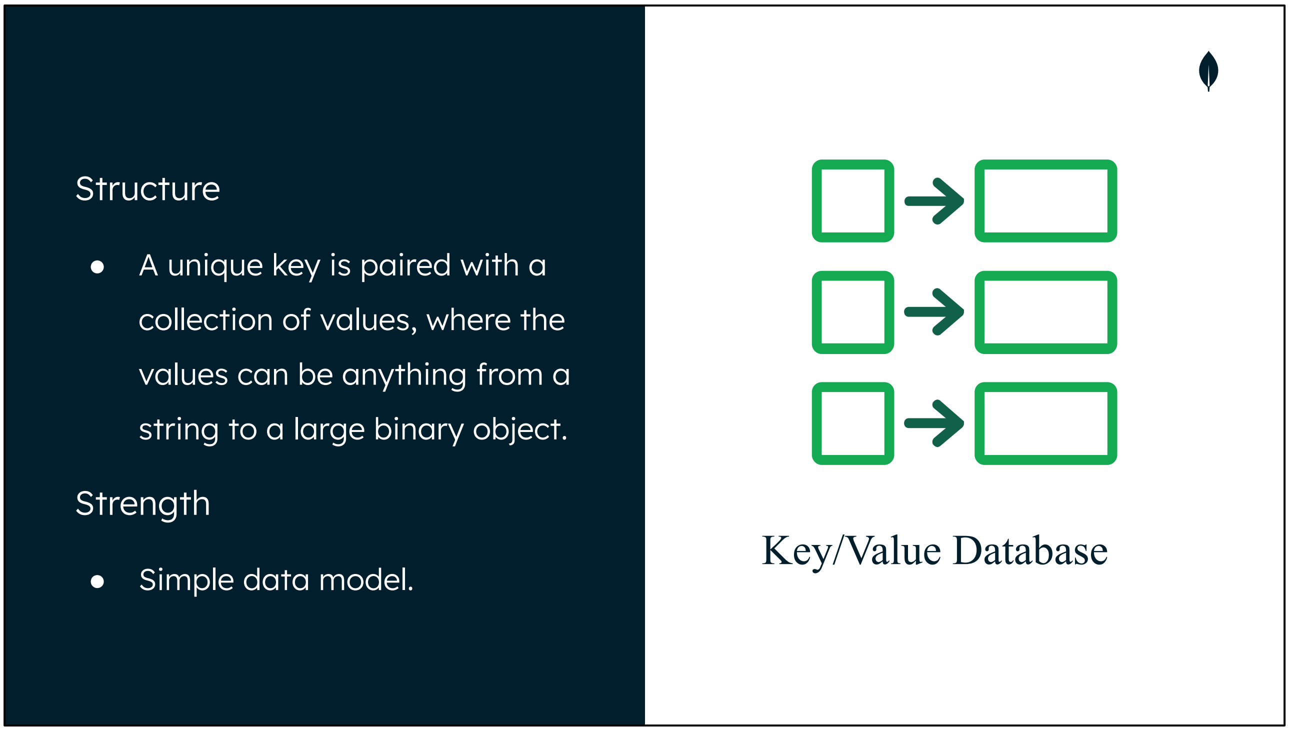
Key/Value

Graph

Column

Document

There are four main types of non-relational databases: key/value, graph, column, and document, and we'll investigate each in this lesson.



Structure

- A unique key is paired with a collection of values, where the values can be anything from a string to a large binary object.

Strength

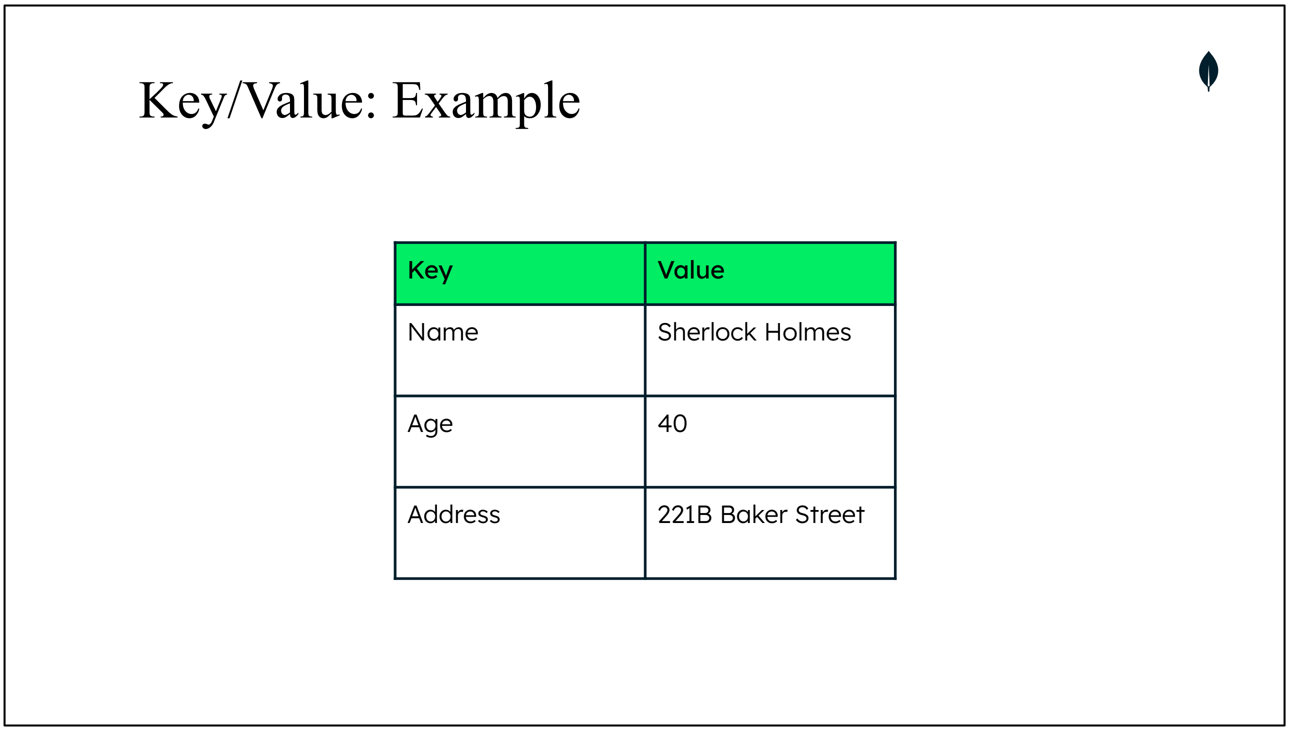
- Simple data model.

Key/Value Database

Key-value databases use a very simple schema: a unique key is paired with a collection of values, where the values can be anything from a string to a large binary object.

One way that databases using this structure gain in performance is that there are no complex queries. The system knows on which server the data is located and sends the request to just that server.

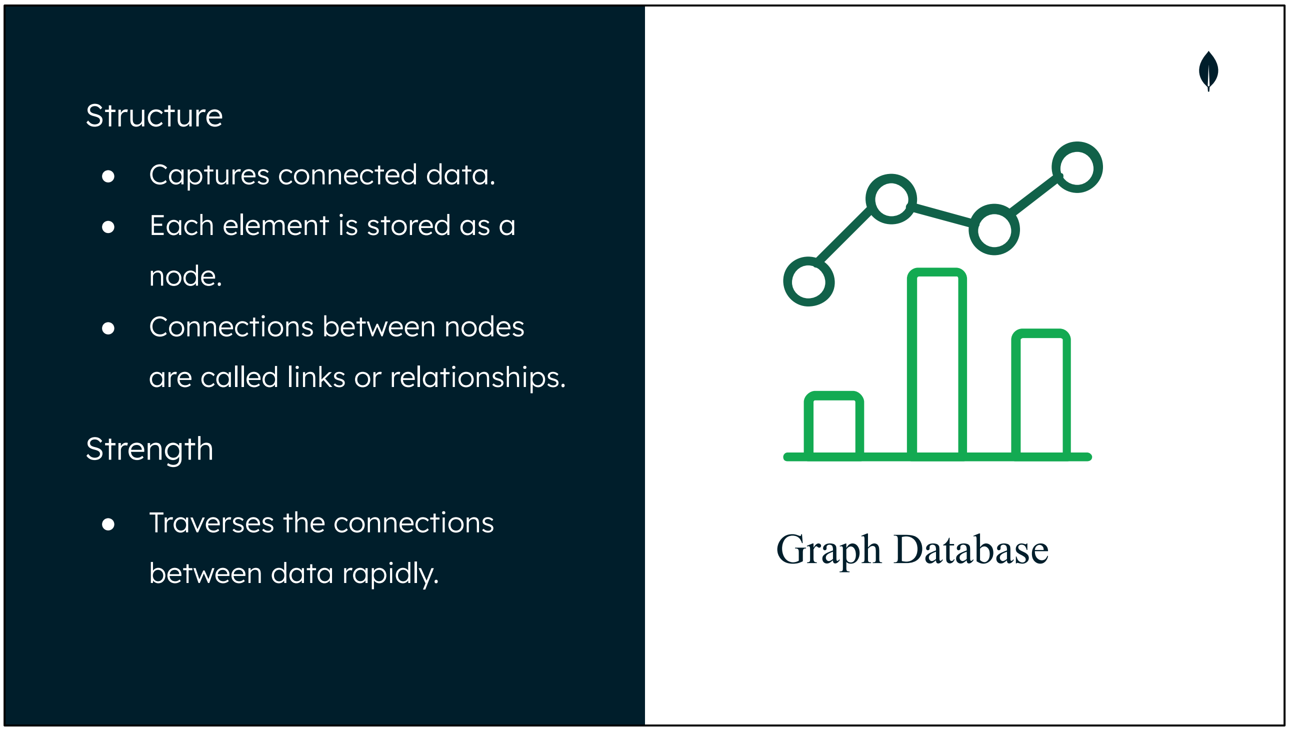
Example: Redis is one of the most popular examples of a key/value store database. .



Key/Value: Example

Key	Value
Name	Sherlock Holmes
Age	40
Address	221B Baker Street

As the simplest of the non-relational databases, key/value is exactly as it sounds, data is organized based on an identifying key and its corresponding value. This simplicity makes it beneficial for large datasets, but not when complex relationships are at play.



Structure

- Captures connected data.
- Each element is stored as a node.
- Connections between nodes are called links or relationships.

Strength

- Traverses the connections between data rapidly.

Graph Database

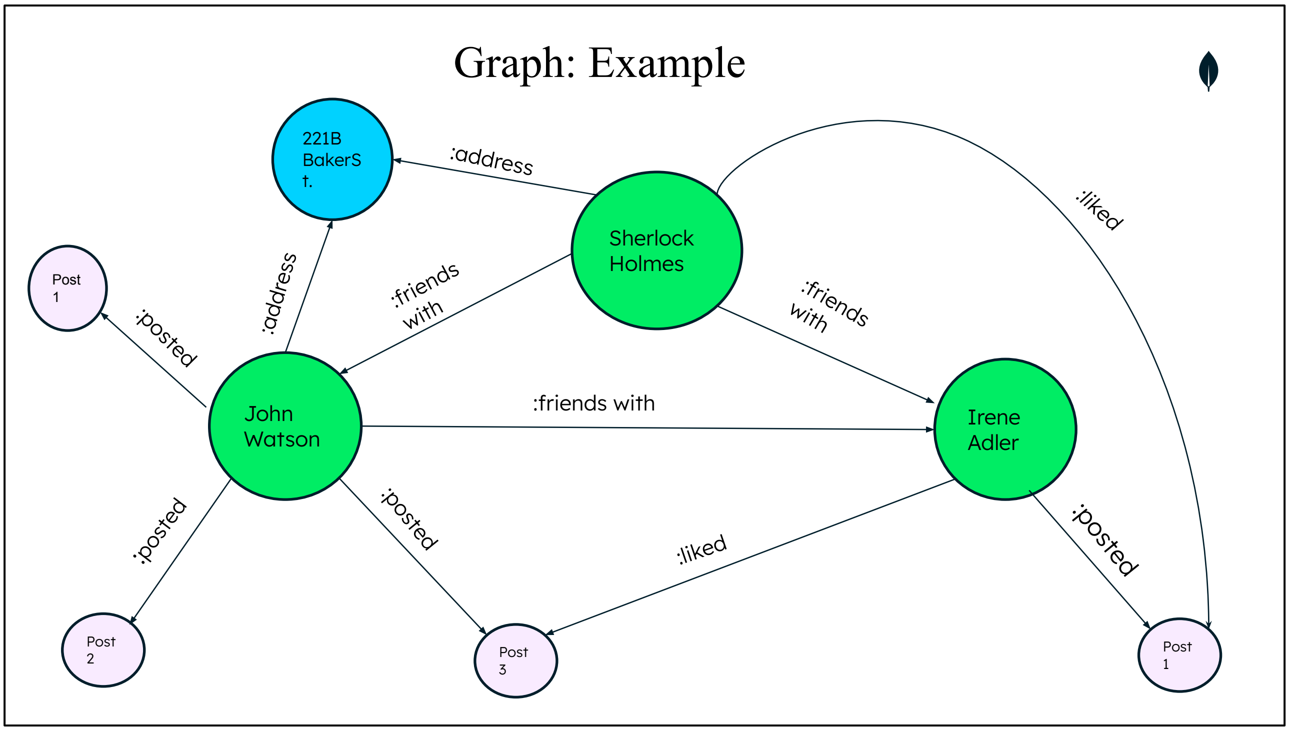
Graph databases are another type within the non-relational family of databases. A popular example of a graph database being used today is Neo4J.

They have been designed to deal with problems around relationships with and focuses on connected data. This type of model does capture the richness of the relationships, however many problems are not naturally modelled as connected data or relationship problems. This makes this database well matched to these problems but as these are niche, it does not have a wider or broader applicability to other problems.

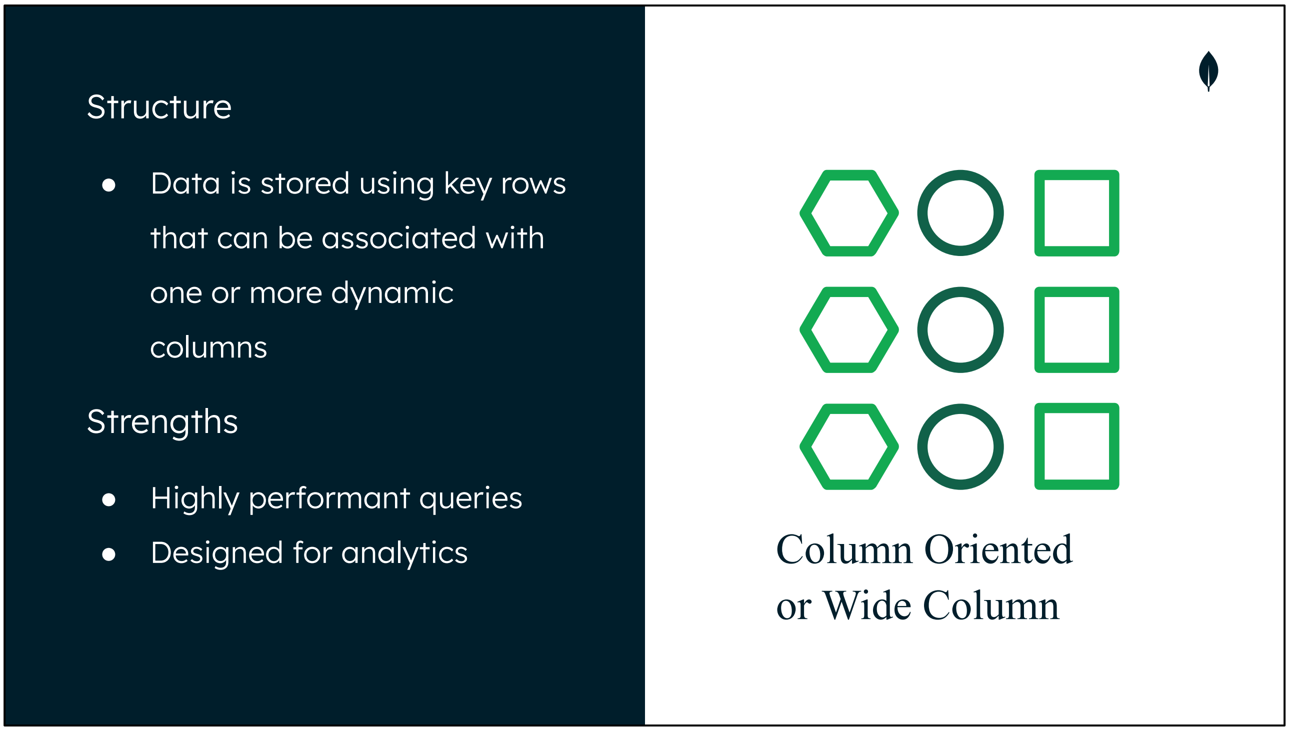
This databases stores information as a collection of nodes and edges, where the edges represent the relationships between the nodes.

Storing the relationships between data means that related data can often be retrieved in a single operation. The concept of relationships between data and this interconnectedness is the key principle behind this type of non-relational database. This approach counters the approach required in SQL/Relational Databases where many joins on several attributes in a number of tables would often be required to retrieve these kinds of relations in the data.

The most popular example of a graph database is Neo4J.



This is a rudimentary example of how data could be stored in a graph database. As you will see, this type of store is useful for social media applications where multiple objects will have multiple relationships or links.



Column Oriented or Wide Column

Structure

- Data is stored using key rows that can be associated with one or more dynamic columns

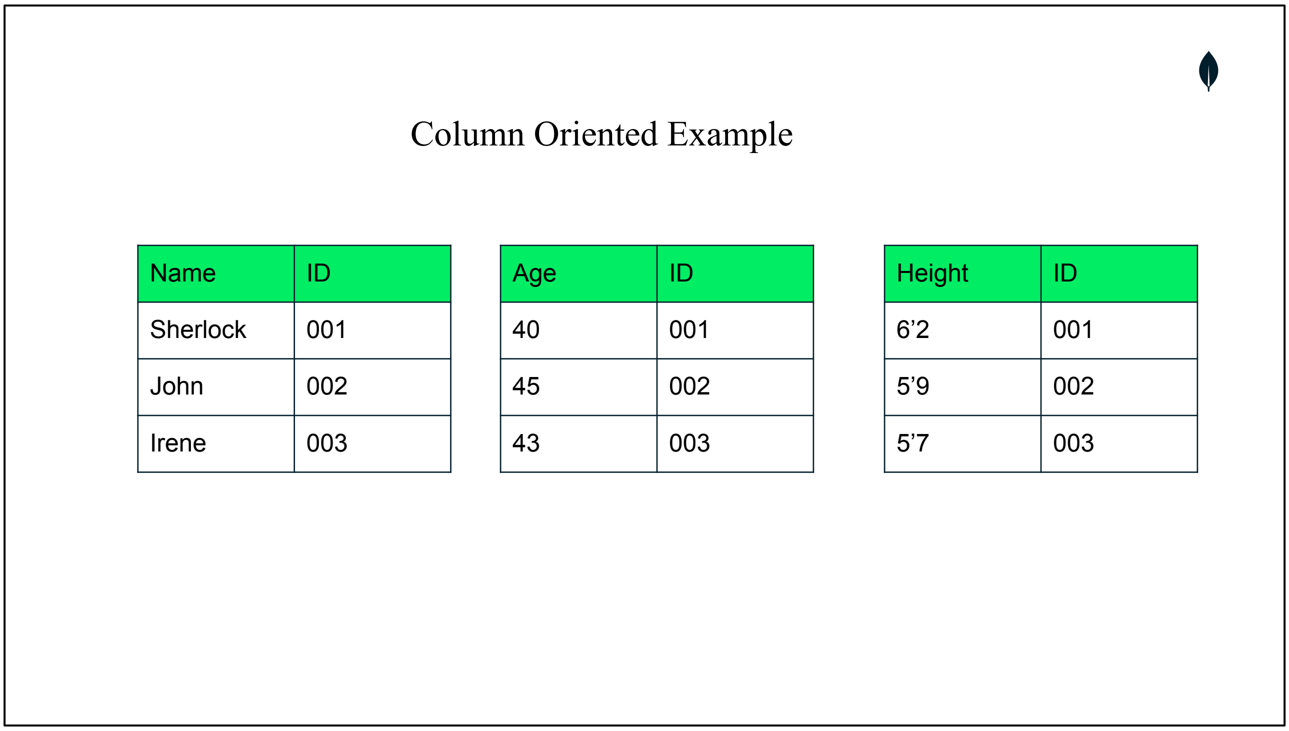
Strengths

- Highly performant queries
- Designed for analytics

A column oriented or wide column non-relational database is primarily designed for analytics. Today, Cassandra is a commonly used column oriented database. The advantage of column versus record/row oriented databases is that column oriented databases return data in columns making the query much more performant as it will not return many irrelevant fields that are not required for the query being serviced. The primary key in a column oriented database is the data / value which is then mapped to row keys. This is the inverse, or opposite, of how the primary key works in a relational database.

The structure of the column data is flexible and can vary from row to row. Associated with storing large amounts of data: billions of rows with millions of columns

This does not necessarily require a separate database and can be implemented as indexes on existing data structures to add this type of functionality to a database.



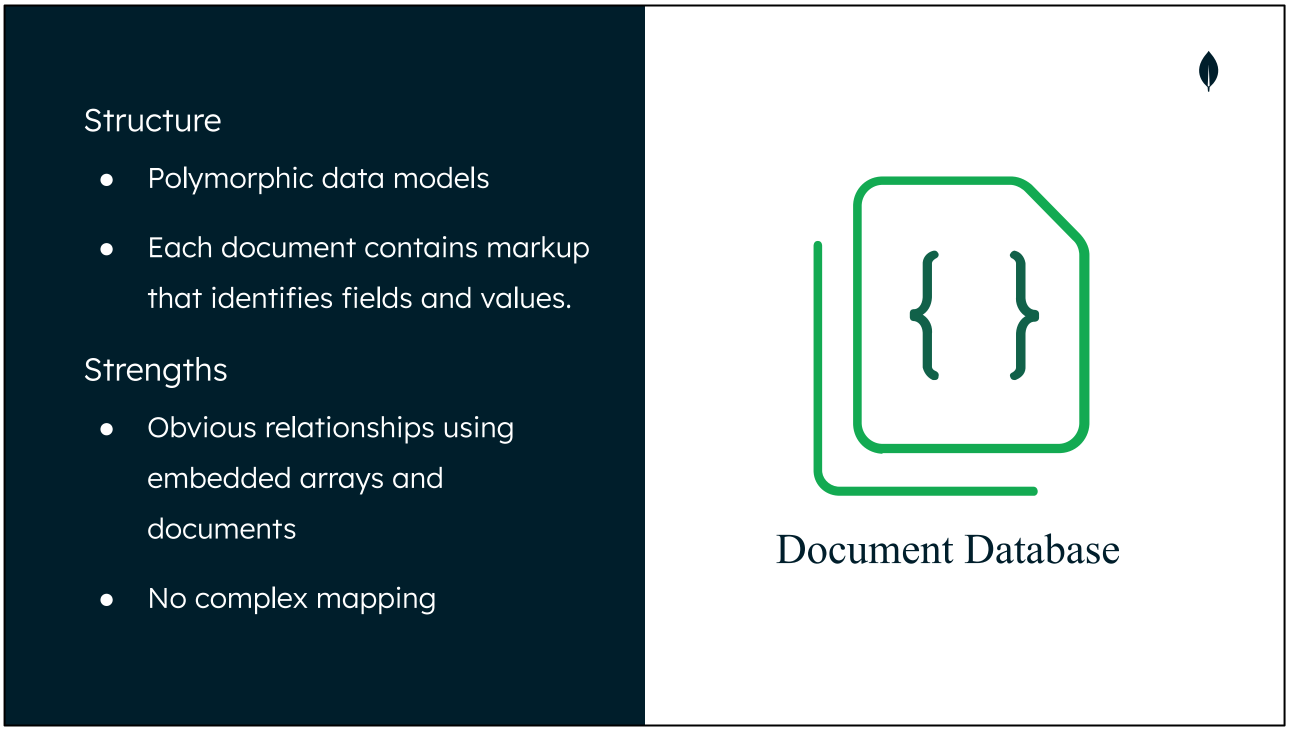
Column Oriented Example

Name	ID
Sherlock	001
John	002
Irene	003

Age	ID
40	001
45	002
43	003

Height	ID
6’2	001
5’9	002
5’7	003

Here we see an example of what data might look like in a column-oriented database. Though at first glance it might seem similar to the known relational tabular format, it is very different in that the data is sorted via column IDs. Therefore the relationships between data are identified via the column key.



Structure

- Polymorphic data models
- Each document contains markup that identifies fields and values.

Strengths

- Obvious relationships using embedded arrays and documents
- No complex mapping

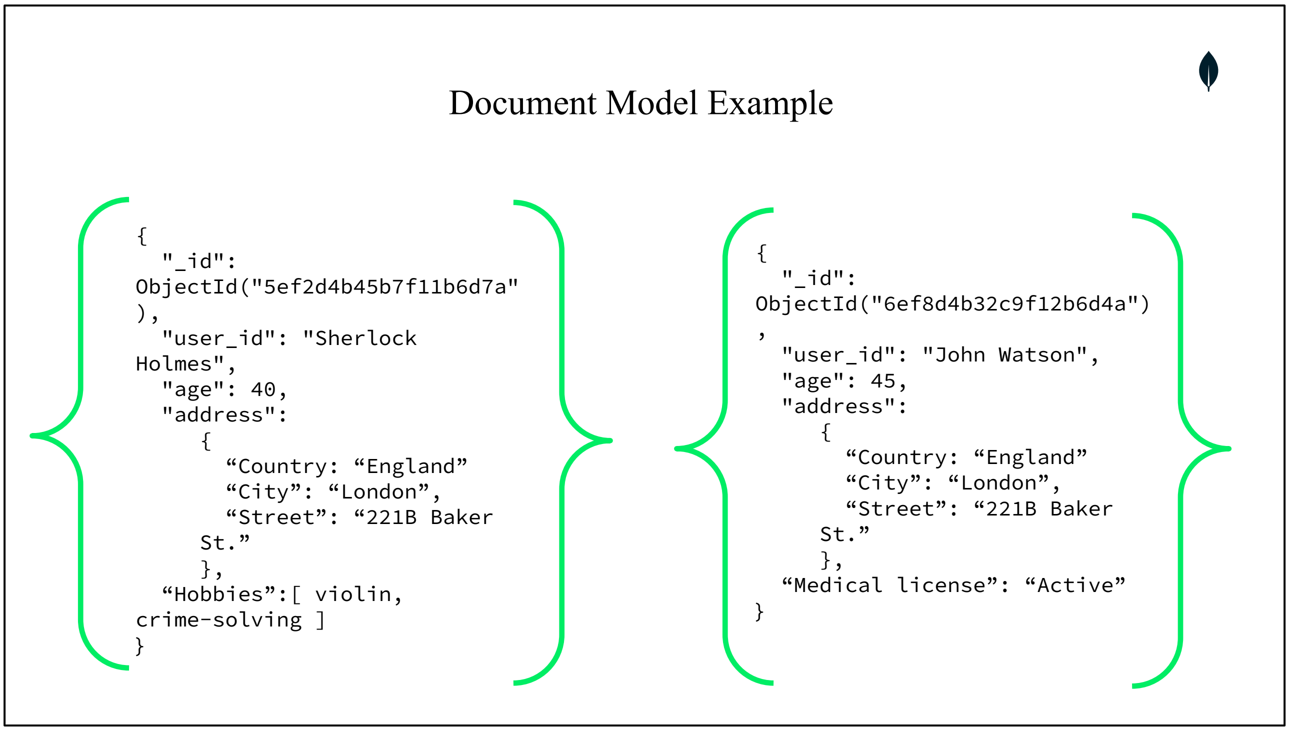
Document Database

Document databases, such as MongoDB, store data in a single document which can have different shapes within the single collection or table that stores the documents.

It provides a clear means of capturing relationship using sub-documents and embedded arrays within a single document.

The document is a close analogy to the object in object oriented programming and provides a clear natural representation of a ‘thing’ and it’s data.

This clear representation often means that there is no requirement for object mapping between the database and the application/programming language. The document is often the exact match for the object that the programmer wishes to use. The flexibility of the document to hold many shapes or multiple parallel schemas at any point in time gives great flexibility for modeling with documents when compared to relational database tables.

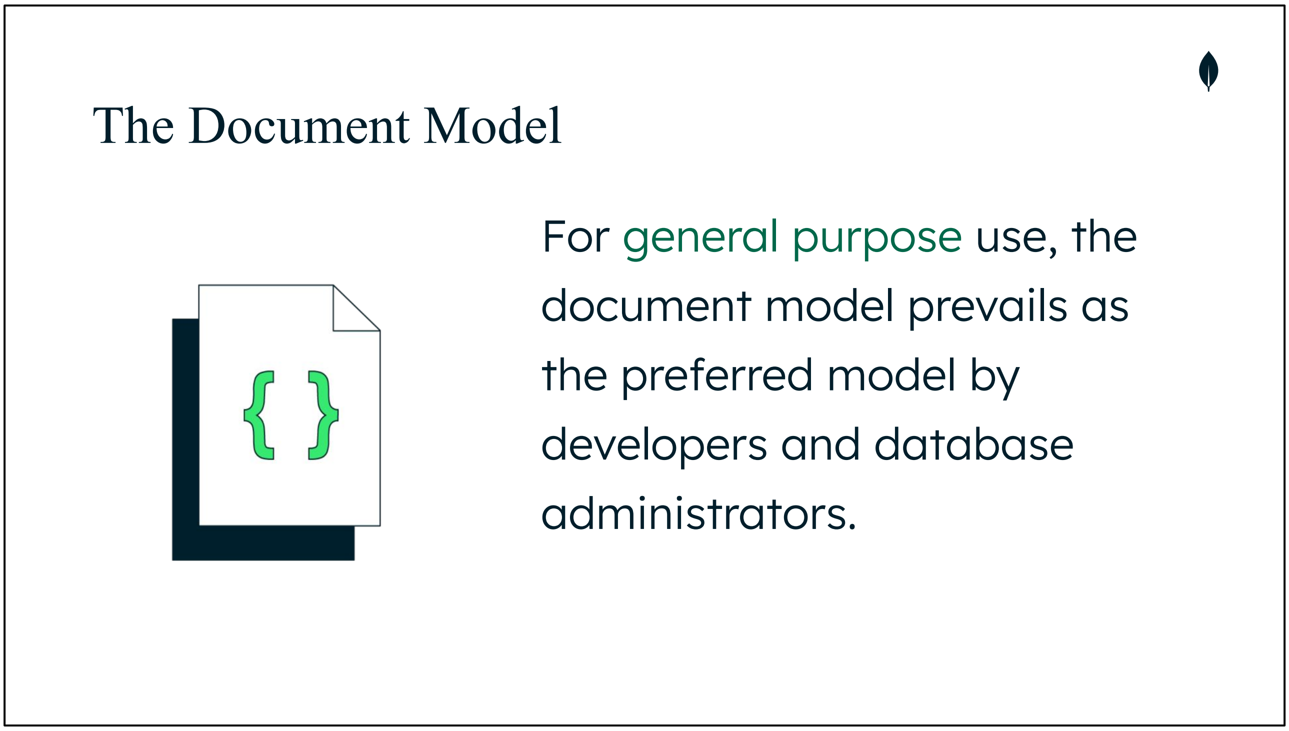


Document Model Example

```
{
  "_id":
ObjectId("5ef2d4b45b7f11b6d7a"
),
  "user_id": "Sherlock
Holmes",
  "age": 40,
  "address":
     {
       “Country: “England”
       “City”: “London”,
       “Street”: “221B Baker
     St.”
     },
  “Hobbies”:[ violin,
crime-solving ]
}
```

```
{
  "_id":
ObjectId("6ef8d4b32c9f12b6d4a")
,
  "user_id": "John Watson",
  "age": 45,
  "address":
     {
       “Country: “England”
       “City”: “London”,
       “Street”: “221B Baker
     St.”
     },
  “Medical license”: “Active”
}
```

The key thing to understand about the document model is that data that is accessed together is stored together. It is also important to note that just because one document has one field does not mean another related document has to have the same field when stored together. We will discuss this more in the lesson when we talk about collections.



The Document Model

For general purpose use, the document model prevails as the preferred model by developers and database administrators.

Due to the fact that the document model implements data structures using programming languages, it is the preferred model by developers as it most closely matches the way they think and work already.

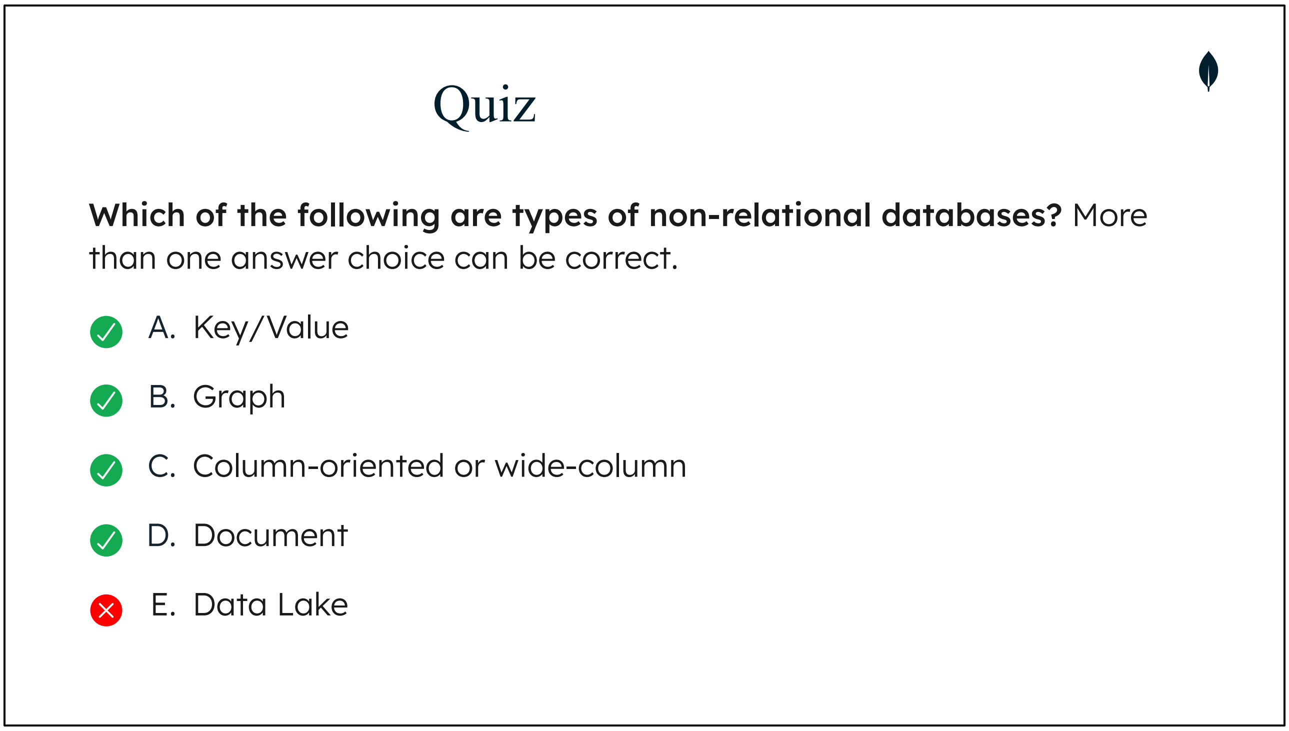
We will take a closer look at the document model and understand how it functions.

Quiz

Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

- A. Key/Value
- B. Graph
- C. Column-oriented or wide-column
- D. Document
- E. Data Lake



Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

✓ A. Key/Value

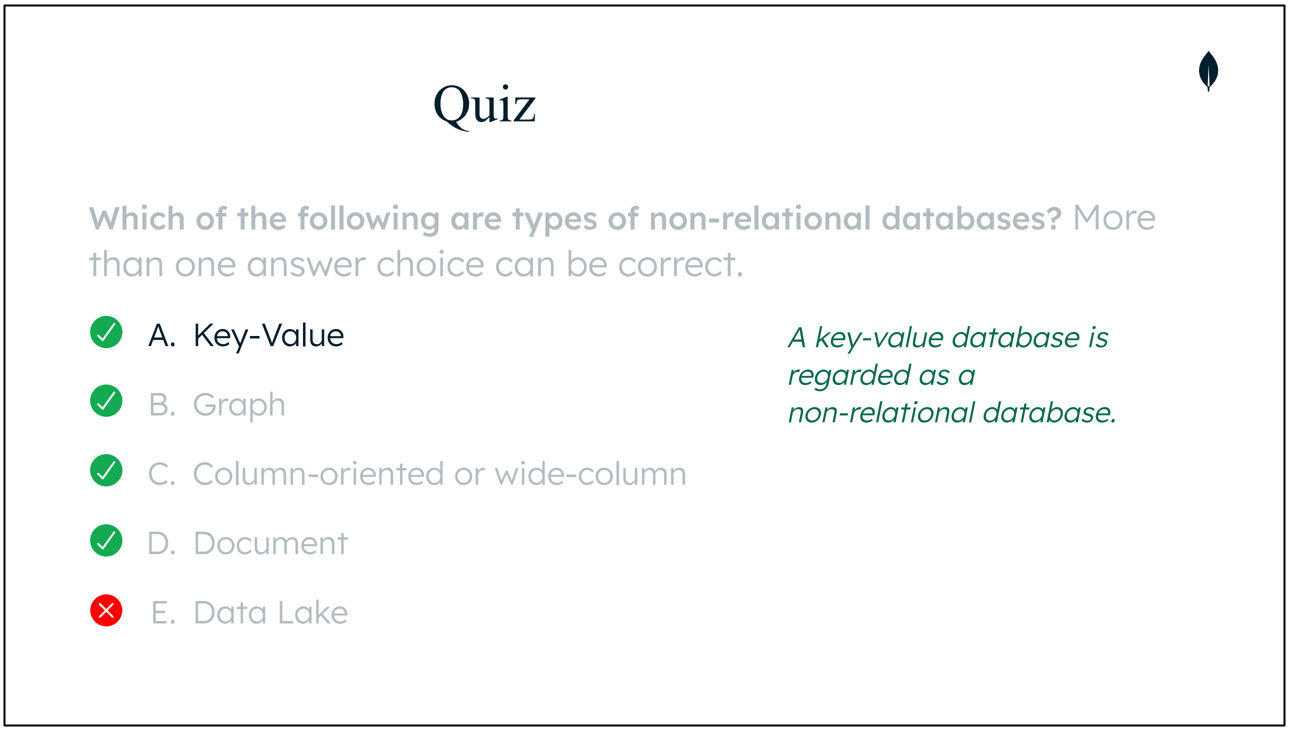
✓ B. Graph

✓ C. Column-oriented or wide-column

✓ D. Document

✗ E. Data Lake

CORRECT: Key/Value - A key/value database is regarded as a non-relational database
CORRECT: Graph - A graph database is regarded as a non-relational database
CORRECT: Column-oriented or wide-column - A column-oriented or wide-column database is regarded as a non-relational database
CORRECT: Document - A document database is regarded as a non-relational database
INCORRECT: Data Lake - A data lake is regarded as a centralised repository of data, both structured and unstructured so it not regarded as non-relation or as relational.



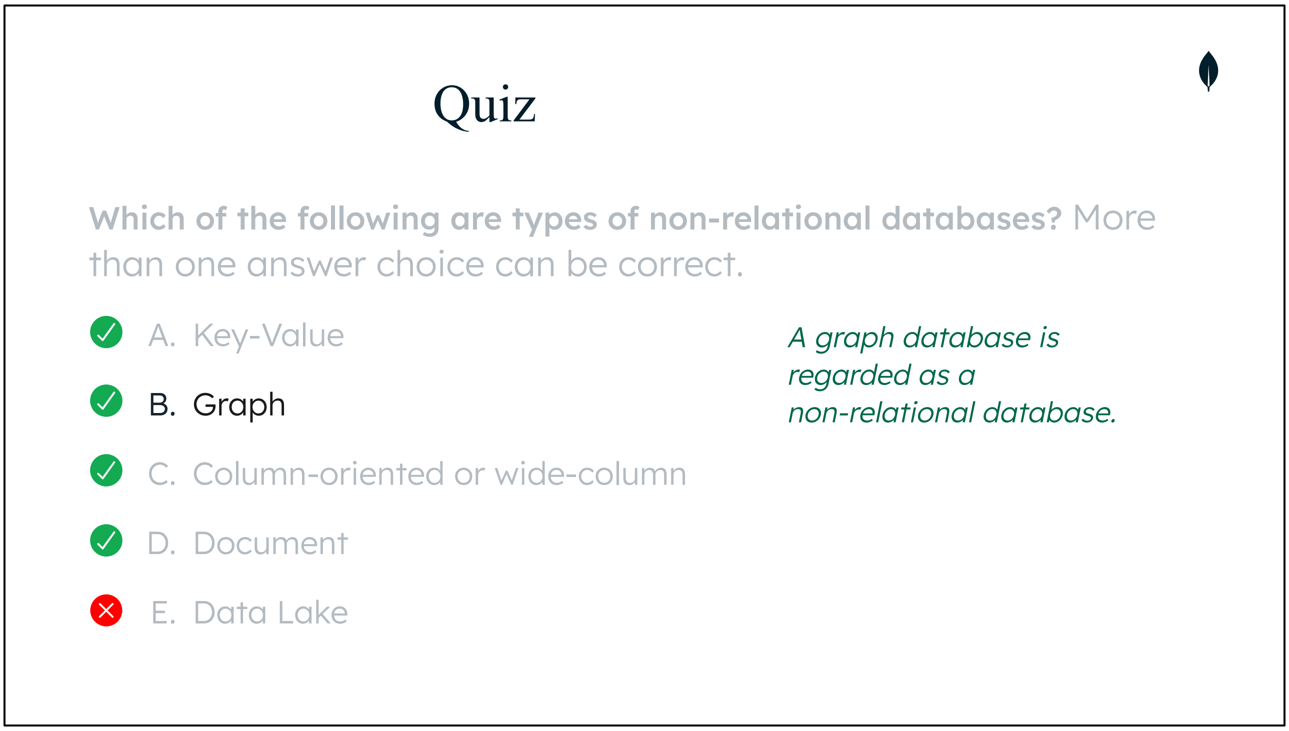
Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

- ✓ A. Key-Value
- ✓ B. Graph
- ✓ C. Column-oriented or wide-column
- ✓ D. Document
- ✗ E. Data Lake

A key-value database is regarded as a non-relational database.

CORRECT: Key-Value - A key-value database is regarded as a non-relational database



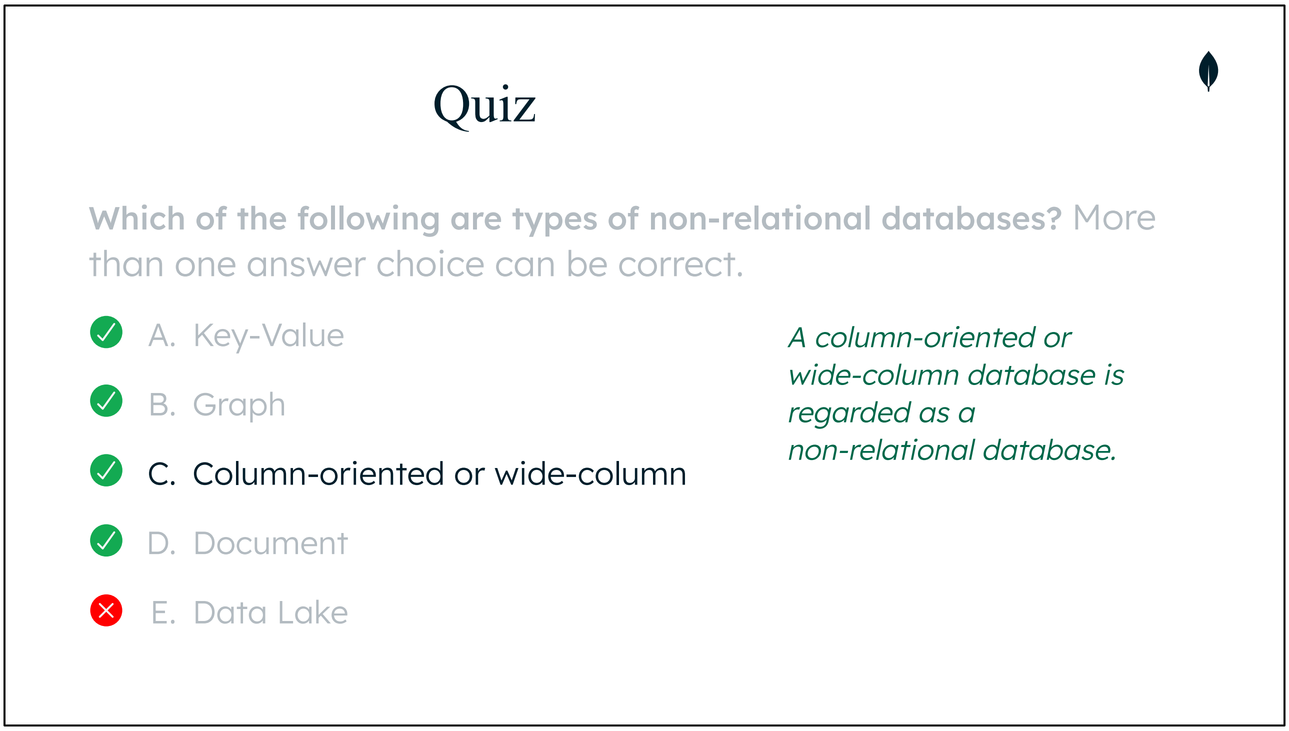
Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

- ✓ A. Key-Value
- ✓ B. Graph
- ✓ C. Column-oriented or wide-column
- ✓ D. Document
- ✗ E. Data Lake

A graph database is regarded as a non-relational database.

CORRECT: Graph - A graph database is regarded as a non-relational database



Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

✓ A. Key-Value

✓ B. Graph

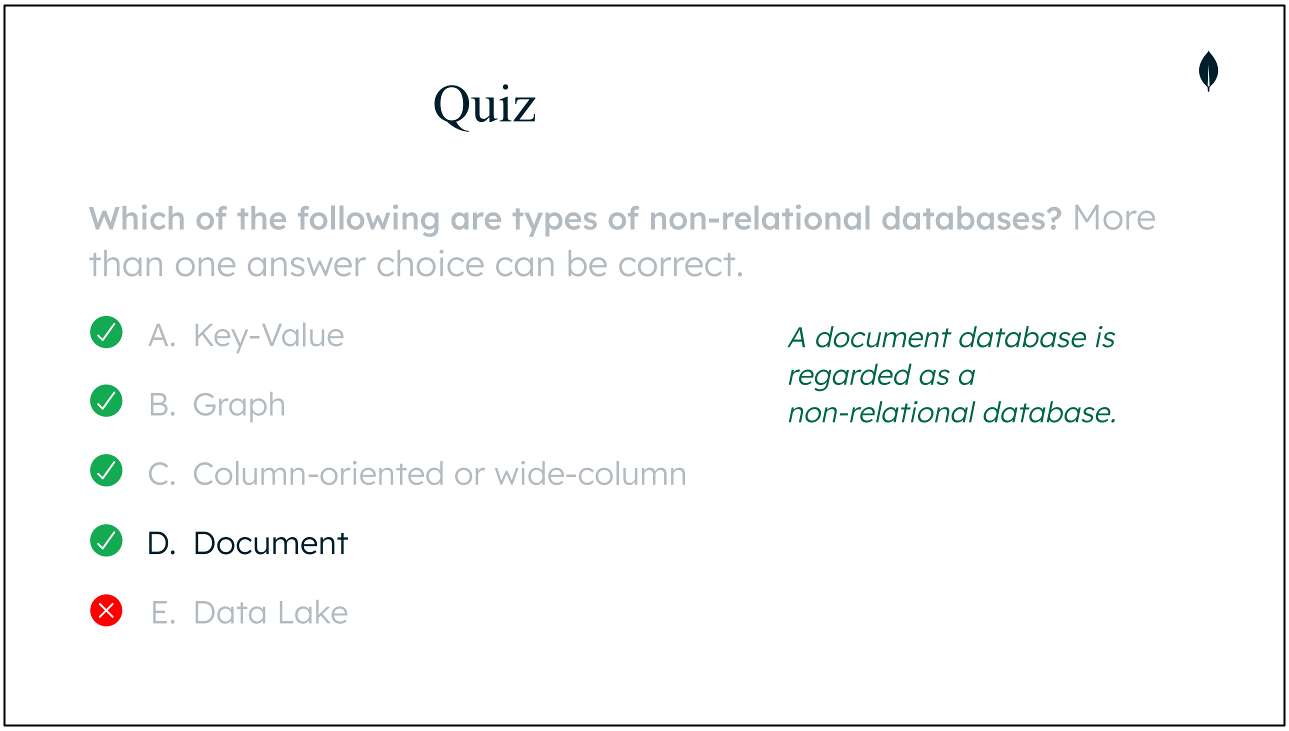
✓ C. Column-oriented or wide-column

✓ D. Document

✗ E. Data Lake

A column-oriented or wide-column database is regarded as a non-relational database.

CORRECT: Column-oriented or wide-column - A column-oriented or wide-column database is regarded as a non-relational database



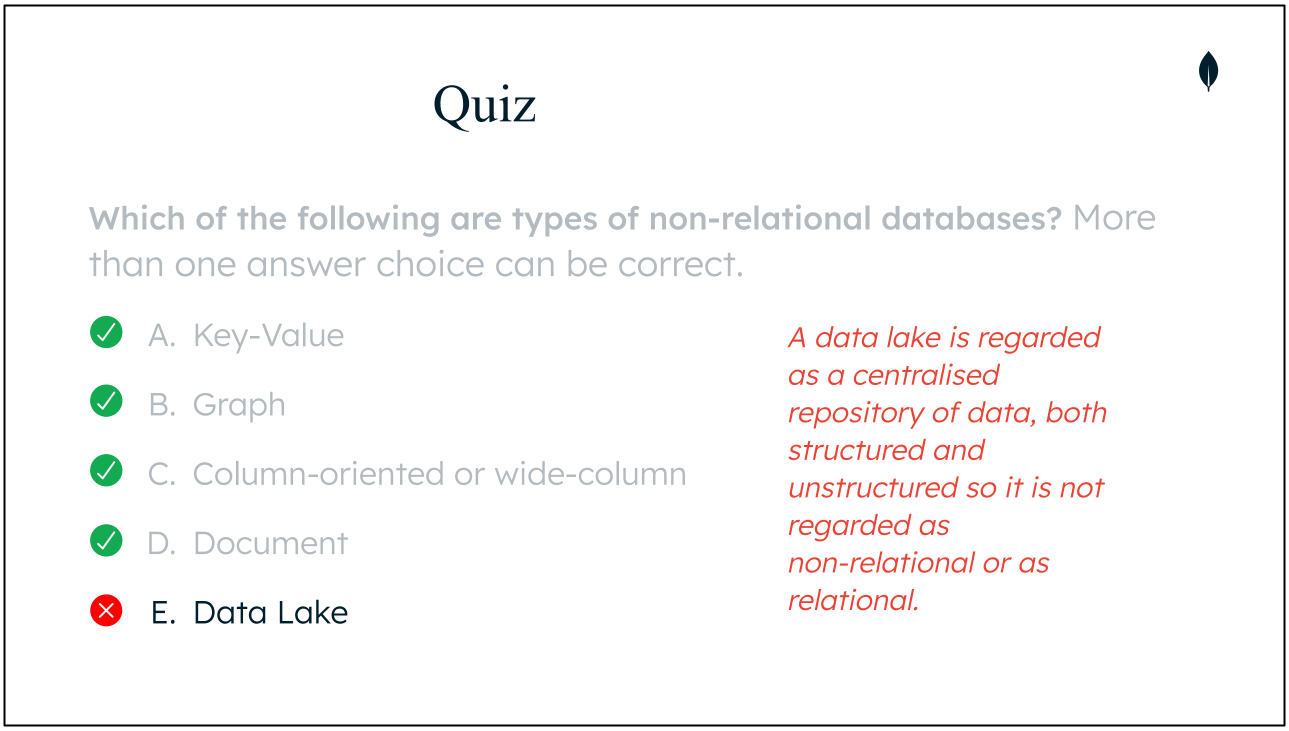
Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

- ✓ A. Key-Value
- ✓ B. Graph
- ✓ C. Column-oriented or wide-column
- ✓ D. Document
- ✗ E. Data Lake

A document database is regarded as a non-relational database.

CORRECT: Document - A document database is regarded as a non-relational database



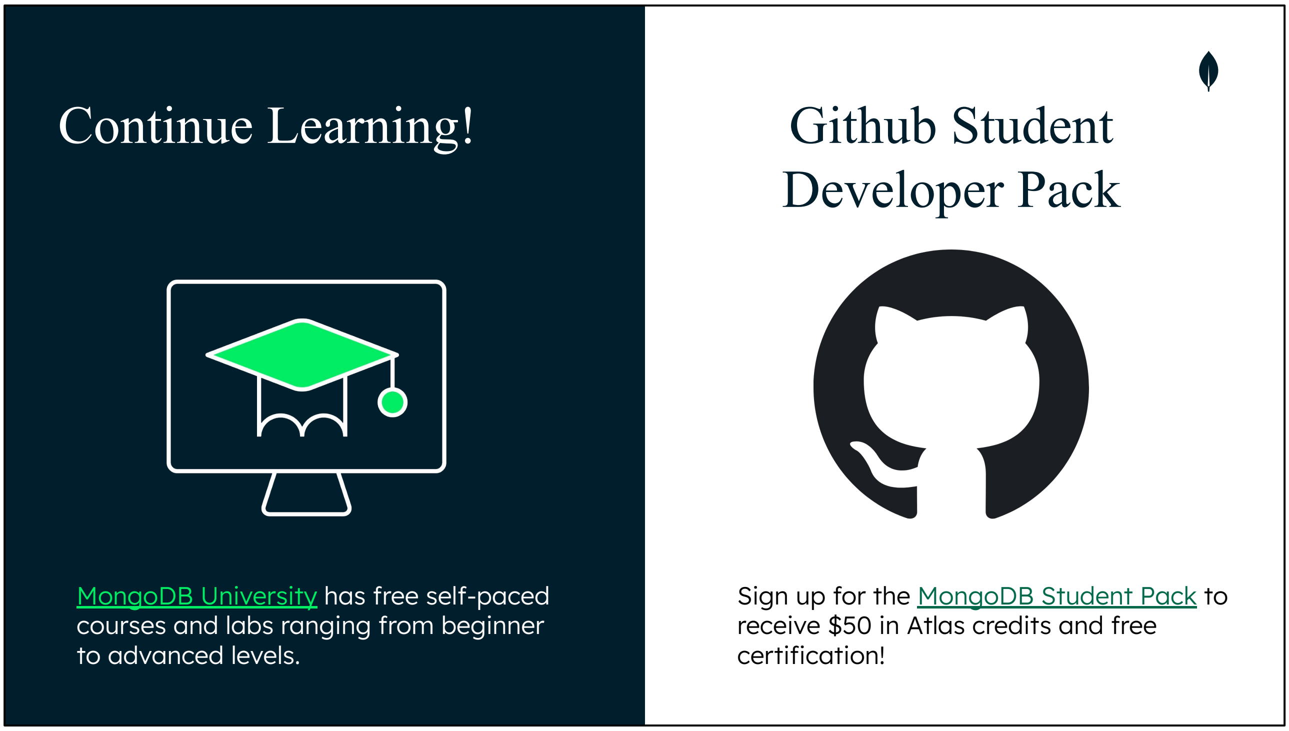
Quiz

Which of the following are types of non-relational databases? More than one answer choice can be correct.

- ✓ A. Key-Value
- ✓ B. Graph
- ✓ C. Column-oriented or wide-column
- ✓ D. Document
- ✗ E. Data Lake

A data lake is regarded as a centralised repository of data, both structured and unstructured so it is not regarded as non-relational or as relational.

INCORRECT: Data Lake - A data lake is regarded as a centralised repository of data, both structured and unstructured so it is not regarded as non-relational or as relational.



Continue Learning!

MongoDB University has free self-paced courses and labs ranging from beginner to advanced levels.

Github Student Developer Pack

Sign up for the MongoDB Student Pack to receive $50 in Atlas credits and free certification!

This concludes the material for this lesson. However, there are many more ways to learn about MongoDB and non-relational databases, and they are all free! Check out MongoDB's University page to find free courses that go into more depth about everything MongoDB and non-relational. For students and educators alike, MongoDB for Academia is here to offer support in many forms. Check out our educator resources and join the Educator Community. Students can receive $50 in Atlas credits and free certification through the Github Student Developer Pack.