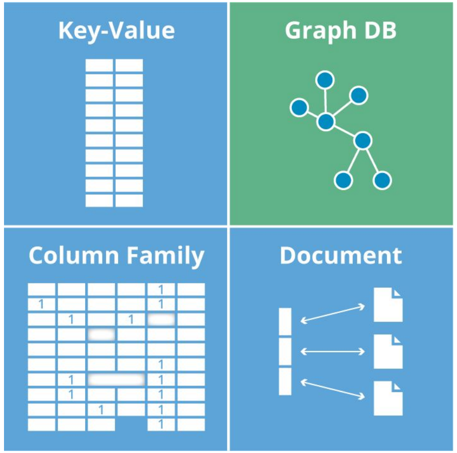


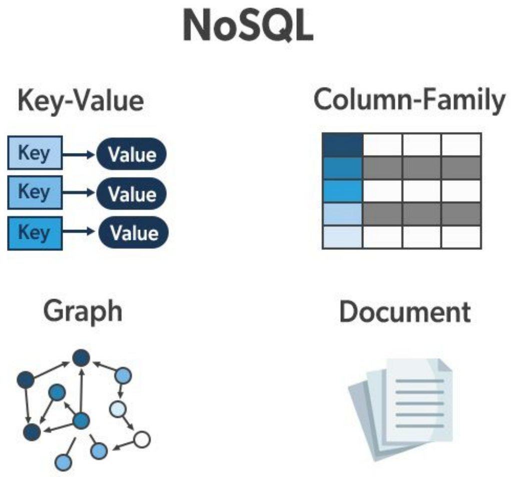
MongoDB®

Introduction to noSQL databases
mongoDB and PyMongo

Feb 26, 2024

Shashank

RESEARCH, INNOVATION & IMPACT
Data Science Institute



- NoSQL databases are non-tabular databases designed to store and retrieve data in ways other than the relational model.
- Types of NoSQL Databases: Document, Key-Value, Wide-Column, and Graph databases.
- Key Features: Scalability, flexibility, high performance, and schema-less data models.
- Use Cases: Big Data applications, real-time web apps, IoT, and more.

Graph Databases: Amazon Neptune, Neo4j

Key value store: Cassandra, Redis, Coherence

Column: Hbase, Big Table, Accumulo

Document-based: Examples – **MongoDB**, CouchDB, Cloudant

SQL vs NoSQL

DBMS	**SQL databases**	**NoSQL databases**
Type	Relational database	Non-relational database
Structure	SQL databases organize and store data by table and fixed columns and rows	NoSQL databases can be key-value, document column-oriented, and graph
Schema	Fixed schema	Dynamic schema
Scalability	Vertical scalability	Horizontal and vertical scalability
Query	Structured Query Language (SQL)	No declarative query language; it depends on the database type

Why It's Useful to Learn NoSQL

Big Data & Scalability: Designed to handle vast amounts of data and scale out easily.

Flexibility: NoSQL databases allow for flexible schema design, which can accommodate the varied and rapidly changing data types.

High Performance: Provides high throughput and low latency, especially for read-heavy applications or where data is accessed in a non-relational manner.

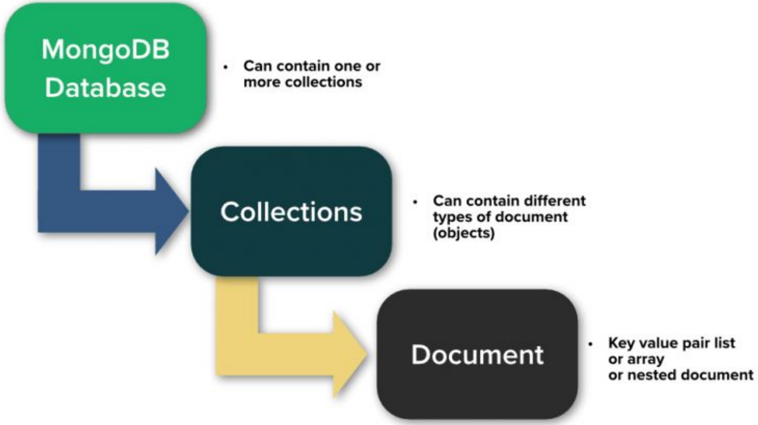
Diverse Data Models: Learning NoSQL exposes you to different data models (document, key-value, graph, wide-column stores).

Understanding MongoDB

MongoDB is a **document-oriented NoSQL database** used for high volume data storage.

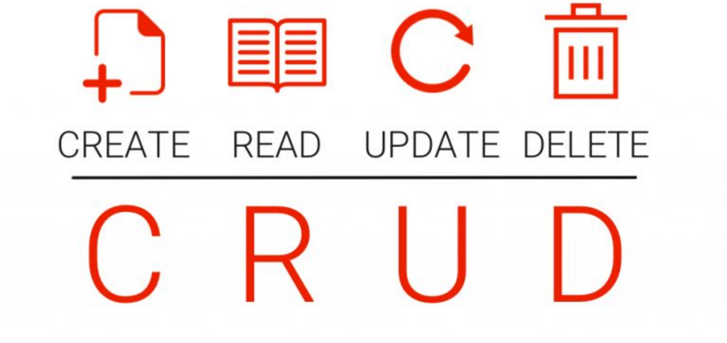
Document Model: Stores data in JSON-like documents with dynamic schemas (BSON), making data integration for certain types of applications easier and faster.

Advantages: Ad-hoc queries, indexing, replication, load balancing, file storage, and aggregation tools.


MongoDB Database
• Can contain one or more collections
Collections
• Can contain different types of document (objects)
Document
• Key value pair list or array or nested document

Introduction to PyMongo

PyMongo is a Python distribution containing tools for working with MongoDB.
Key Features: Easy-to-use interfaces to MongoDB for Python applications.



CRUD Operations: How to create, read, update, and delete documents in MongoDB using PyMongo.

Querying Documents: Techniques for searching and filtering data in MongoDB.

Aggregation Framework: Using PyMongo to perform complex aggregations and data processing.