



## Introduction

You can create a MongoDB database in the following environments:

- [MongoDB Atlas](#): The fully managed service for MongoDB deployments in the cloud
- [MongoDB Enterprise](#): The subscription-based, self-managed version of MongoDB
- [MongoDB Community](#): The source-available, free-to-use, and self-managed version of MongoDB

## Document Database

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

The advantages of using documents are:

- Documents correspond to native data types in many programming languages.
- Embedded documents and arrays reduce need for expensive joins.

- Dynamic schema supports fluent polymorphism.

## Collections/Views/On-Demand Materialized Views

MongoDB stores documents in [collections](#). Collections are analogous to tables in relational databases.

In addition to collections, MongoDB supports:

- Read-only [Views](#)
- [On-Demand Materialized Views](#)

## Key Features

### High Performance

MongoDB provides high performance data persistence. In particular,

- Support for embedded data models reduces I/O activity on database system.
- Indexes support faster queries and can include keys from embedded documents and arrays.

### Query API

The MongoDB Query API supports [read and write operations \(CRUD\)](#) as well as:

- [Data Aggregation](#)
- [Text Search](#) and [Geospatial Queries](#).

### High Availability

MongoDB's replication facility, called [replica set](#), provides:

- *automatic* failover
- data redundancy.

A [replica set](#) is a group of MongoDB servers that maintain the same data set, providing redundancy and increasing data availability.

## Horizontal Scalability

MongoDB provides horizontal scalability as part of its *core* functionality:

- [Sharding](#) distributes data across a cluster of machines.
- Starting in 3.4, MongoDB supports creating [zones](#) of data based on the [shard key](#). In a balanced cluster, MongoDB directs reads and writes covered by a zone only to those shards inside the zone. See the [Zones](#) manual page for more information.

## Support for Multiple Storage Engines

MongoDB supports [multiple storage engines](#):

- [WiredTiger Storage Engine](#) (including support for [Encryption at Rest](#))
- [In-Memory Storage Engine](#).

# DATABASE

A database is **an electronically stored, systematic collection of data**. It can contain any type of data, including words, numbers, images, videos, and files. You can use software called a database management system (DBMS) to store, retrieve, and edit data.

## Database management system(DBMS)

Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

## Data Types:

Databases can hold various kinds of information, including text, numbers, images, videos, and more.

# SQL VS NO- SQL

## STRUCTURED QUERY LANGUAGE(SQL)

Structured query language (SQL) is a programming language for storing and processing information in a relational database.

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

## NO- SQL

NoSQL, also referred to as “not only SQL” or “non-SQL”, is an approach to database design that enables the storage and querying of data outside the traditional structures found in relational databases.

### NoSQL databases are generally classified into four main categories:

1. **Document databases:** These databases store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages.
2. **Key-value stores:** These databases store data as key-value pairs, and are optimized for simple and fast read/write operations.
3. **Column-family stores:** These databases store data as column families, which are sets of columns that are treated as a single entity. They are optimized for fast and efficient querying of large amounts of data.
4. **Graph databases:** These databases store data as nodes and edges, and are designed to handle complex relationships between data.

# Date & Delete

## BASICS OF MongoDB

- MongoDB is a NoSQL Database
- Each Row can be associated with Documents
- Documents can be represented as JSON
- Each table can be called as Collection

## Links for downloading csv and import data

Download the student csv from this [link](#)

Import the data to the collection created [link](#)

### Installation of mongoshell

- Mongo Shell download [link](#)
- Mongo compass download  
<https://www.mongodb.com/docs/compass/current/install/>

## Commands

commands	Expected Output	Notes
show dbs	<pre>admin 40.00 KiB config 72.00 KiB db 128.00 KiB local 40.00 KiB</pre>	All Databases are shown
use db	<pre>switched to db db</pre>	Connect and use db
show collections	<pre>Students</pre>	Show all tables

db.foo.insert({"bar" : "baz"})		Insert a record to collection. Create Collection if not exists
--------------------------------	--	---

## Show dbs

In MongoDB, you can use the show dbs command to **list all databases on a MongoDB server**. This will show you the database name, as well as the size of the database in gigabytes.

## Use db

The use command is **used when there are multiple databases in the SQL and the user or programmer specifically wants to use a particular database**.

Thus, in simple terms, the use statement selects a specific database and then performs operations on it using the inbuilt commands of SQL.

## Show collections

For users with the required access, show collections **lists the non-system collections for the database**. For users without the required access, show collections lists only the collections for which the users has privileges.

Command	Notes
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows

db.foo.remove()	Remove foo table
-----------------	------------------

## Documents

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

The representation of a document varies by programming language, but most languages have a data structure that is a natural fit, such as a map, hash, or dictionary.

## Collections

Collections A collection is a group of documents.

If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

A collection is a grouping of MongoDB documents. Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single database.

## DATATYPES

- String. One of the most basic and widely used data types is the string. ...
- Integer. Numeric values are stored using the integer data type. ...
- Double. Numeric numbers containing 8 bytes (64-bit IEEE 754 floating point) floating-point are stored using the double data type. ...
- Boolean. ...
- Array. ...
- Object. ...
- Date. ...
- Timestamp.

## MongoDB Use Cases

- Artificial Intelligence. Streamline building AI-enriched applications. ...
- Edge Computing. Unlock the benefits of edge computing by simplifying data management. ...
- Internet of Things. Analyze and act on data from the physical world. ...

### Mobile.

- Payments. ...
- Serverless Development. ...
- Single View. ...
- Personalization.