

# NODEJS - DATABASE

ANECO ACADEMY

# DATABASE

## **What is a database?**

A database is information that is set up for easy access, management and updating. Computer databases typically store aggregations of [data](#) records or [files](#) that contain information, such as sales transactions, customer data, financials and product information.

Databases are used for storing, maintaining and accessing any sort of data. They collect information on people, places or things. That information is gathered in one place so that it can be observed and analyzed. Databases can be thought of as an organized collection of information.



# DATABASE

Businesses use data stored in databases to make informed business decisions. Some of the ways organizations use databases include the following:

- **Improve business processes.** Companies collect data about [business processes](#), such as sales, order processing and customer service. They analyze that data to improve these processes, expand their business and grow revenue.
- **Keep track of customers.** Databases often store information about people, such as customers or users. For example, [social media](#) platforms use databases to store user information, such as names, email addresses and user behavior. The data is used to recommend content to users and improve the [user experience](#).
- **Secure personal health information.** Healthcare providers use databases to securely store [personal health data](#) to inform and improve patient care.
- **Store personal data.** Databases can also be used to store personal information. For example, personal [cloud storage](#) is available for individual users to store media, such as photos, in a managed cloud.



# SQL VS NOSQL

## 1. Type –

SQL databases are primarily called as Relational Databases (RDBMS); whereas NoSQL database are primarily called as non-relational or distributed database.

## 2. Language –

SQL databases defines and manipulates data based structured query language (SQL). Seeing from a side this language is extremely powerful. SQL is one of the most versatile and widely-used options available which makes it a safe choice especially for great complex queries. But from other side it can be restrictive. SQL requires you to use predefined schemas to determine the structure of your data before you work with it. Also all of your data must follow the same structure. This can require significant up-front preparation which means that a change in the structure would be both difficult and disruptive to your whole system.

A NoSQL database has dynamic schema for unstructured data. Data is stored in many ways which means it can be document-oriented, column-oriented, graph-based or organized as a KeyValue store. This flexibility means that documents can be created without having defined structure first. Also each document can have its own unique structure. The syntax varies from database to database, and you can add fields as you go.

## SQL VS NOSQL

### 3. Scalability –

In almost all situations SQL databases are vertically scalable. This means that you can increase the load on a single server by increasing things like RAM, CPU or SSD. But on the other hand NoSQL databases are horizontally scalable. This means that you handle more traffic by sharding, or adding more servers in your NoSQL database. It is similar to adding more floors to the same building versus adding more buildings to the neighborhood. Thus NoSQL can ultimately become larger and more powerful, making these databases the preferred choice for large or ever-changing data sets.

### 4. Structure –

SQL databases are table-based on the other hand NoSQL databases are either key-value pairs, document-based, graph databases or wide-column stores. This makes relational SQL databases a better option for applications that require multi-row transactions such as an accounting system or for legacy systems that were built for a relational structure.

### 5. Property followed –

SQL databases follow [ACID properties](#) (Atomicity, Consistency, Isolation and Durability) whereas the NoSQL database follows the Brewers [CAP theorem](#) (Consistency, Availability and Partition tolerance).

### **6. Support –**

Great support is available for all SQL database from their vendors. Also a lot of independent consultations are there who can help you with SQL database for a very large scale deployments but for some NoSQL database you still have to rely on community support and only limited outside experts are available for setting up and deploying your large scale NoSQL deployments.



## SQL

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

These databases have fixed or static or predefined schema

These databases are not suited for hierarchical data storage.

These databases are best suited for complex queries

Vertically Scalable

Follows ACID property

**Examples:** MySQL, PostgreSQL, Oracle, MS-SQL Server etc

## NoSQL

Non-relational or distributed database system.

They have dynamic schema

These databases are best suited for hierarchical data storage.

These databases are not so good for complex queries

Horizontally scalable

Follows CAP (consistency, availability, partition tolerance)

**Examples:** MongoDB, GraphQL, HBase, Neo4j, Cassandra etc



## PEOPLE

```
{
  "Id": 1,
  "FirstName": "Ada",
  "LastName": "Lovelace",
  "Email": "ada.lovelace@gmail.com",
  "Phone": [{
    "Home": "+1.123.456.7890"
  },
  {
    "Work": "+1.111.222.3333"
  }
]
}
```

```
{
  "Id": 2,
  "FirstName": "Grace",
  "LastName": "Hopper",
  "Email": "grace.hopper@gmail.com"
}
```

```
{
  "Id": 3,
  "FirstName": "Kathy",
  "LastName": "Sierra",
  "Email": "kathy.sierra@gmail.com"
}
```



## PERSON

Id	FirstName	LastName	Email
1	Ada	Lovelace	ada.lovelace@gmail.com
2	Grace	Hopper	grace.hopper@gmail.com
3	Kathy	Sierra	kathy.sierra@gmail.com

## PHONE\_NUMBER

PersonId	PhoneId	Phone Number	Type
1	1	+1.123.456.7890	Home
1	2	+1.111.222.3333	Work