## Explain, generally, what is meant by a NoSQL database

A NoSQL database is a database where it’s possible to store and retrieve data without the usage of SQL statements. This is in contrast to a relational database where every query made is done by using SQL query calls. The need for NoSQL databases came due to the fact that normal relational databases where not designed to cope with scale and agility challenges that many developers face today.

Some of the different kind of NoSQL databases that exist are:

* **Document-oriented databases:** These pair each key with a complex data structure also known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.
* **Graph stores:** These are used to store information about networks of data, such as social connections.
* **Key-value stores:** These are the simplest NoSQL databases. Every item is stored as an attribute name (a key), together with its value.
* **Wide-column stores:** These are used for large datasets, and store columns of data together instead of in rows.

Another difference between NoSQL databases and relational databases is that the latter uses schemas to be defined before data is added. A NoSQL database allow the insertion of data without a schema which makes it easy to make application changes in real-time without worrying about service interruptions. That said it’s also possible to make schemas for NoSQL databases with the use of node modules. A popular node module for this is mongoose.

## Explain Pros & Cons in using a NoSQL database like MongoDB as your data store, compared to a traditional Relational SQL Database like MySQL

**Pros:**

* **Flexible Data Model:** Can store and combine any type of data.
* **Dynamically updated schemas:** As NoSQL databases don’t require schemas, this means that if requirements for the database changes the “schema” can be updated without any downtime (taking the database offline as would be needed with a relational database)
* **Elastic Scalability:** NoSQL databases scale out on low cost, commodity hardware, allowing for almost unlimited growth
* **High Performance:** Faster queries, inserts and updates

**Cons:**

* **Might lose referential integrity guarantees:** “Eventual consistency” is a term used in the NoSQL universe, meaning that an update or insert on one server may not be visible on other servers immediately.
* **No universal language:** There is no universal language for NoSQL databases like SQL

## Explain reasons to add a layer like Mongoose, on top on of a schema-less database like MongoDB

The main reason in my opinion to add Mongoose to MongoDB is so you can validate data (making sure that the data that is being added or updated to the database fits the requirements of what should be allowed). This is done by making a schema for the database that specifies how your data is supposed to look, and what type is allowed in different fields.

Another reason is that if you are used to working with relational database, it will make the databases look more alike, and create models so it looks like you are working objects instead of pure data.

## Explain the benefits from using Mongoose, and provide an example involving all CRUD operations

Benefits of using Mongoose:

* Validate data
* Making the database structure more ORM like

See RestAPI for an example of how to use Mongoose.

## Explain, using a relevant example, a full MEAN application including relevant test cases to test the REST-API