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

Characteristics of NoSQL:

- **Non-relational**
  Which means that we don't divide the database into different tables of atomic values using normalization, as we do with SQL.
- **Mostly open-source**
  Almost all of what is characterized as NoSQL databases are open-source. There are a few exceptions, but they aren't relevant to us at the moment.
- **Cluster-friendly**
  Since there are no relations, NoSQL databases are easily scalable.
- **21st century web**
  All NoSQL databases come from the 21st century web culture.
- **Schema-less**

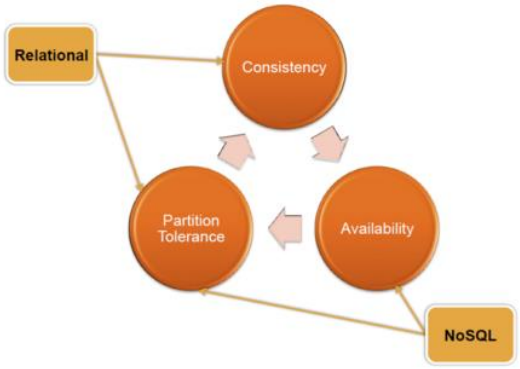
Explain Pros & Cons in using a NoSQL database like MongoDB as your data store, compared to a traditional Relational SQL Database like MySQL.

Pros:

- Simplicity of design
- Horizontal scaling
- Finer control over availability

Cons:

- No structure in relation to real-world entities
- Does not offer **ACID** guarantees:
  - **A**tomicity
  - **C**onsistency
  - **I**solation
  - **D**urability



Additionally the CAP theorem (model on the right) states:

It's theoretically impossible to have all 3 requirements met, so a combination of 2 must be chosen and this is usually the deciding factor in what technology is used.

Consistency:
All the servers in the system will have the same data so anyone using the system will get the same copy regardless of which server answers their request.

Availability:
The system will always respond to a request (even if it's not the latest data or consistent across the system or just a message saying the system isn't working).

Partition Tolerance
The system continues to operate as a whole even if Individual servers fail or can't be reached.

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

- Real data normally has some kind of structure, which we would like to mimic.
- Real data normally has types.
- To handle data as objects easier.

So it basically boils down to **being able to do more with less work**.

Explain the benefit from using Mongoos and provide an example involving all CRUD operations.

Mongoose is an object modeling tool for MongoDB and Node.js, somehow similar to a ORM tool as we know.
The reason why there are benefits of using Mongoose, is that it provides a straight-forward, schema-based solution for modeling your application data and includes, out of the box:

- Schemas
- Built-in type casting
- Validation (also include with plain MongoDB as of v. 3.2)
- Query building
- Business logic hooks (middleware)

This all makes it quicker and easier to make a database than just using MongoDB.
Another reason is, that Mongoose allows us to have access the MongoDB commands for CRUD simply and easily.

Futhermore Mongoose offers more functionality (one of them is being an ODM (Object Document Mapping)).

Mongoose also adds in some necessary data manipulation functions and it works as a layer to be used for future additions (plugins) , rather than messing with the source of MongoDB itself.

With mongoose, you also got update, delete, validation framework, support for custom middleware etc.