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**

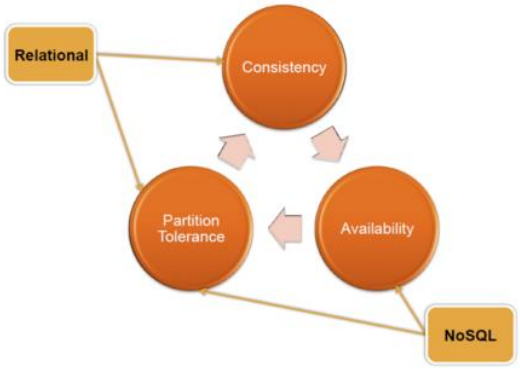
Explain Pros & Cons in using a NoSQL database like MongoDB as your data store, compared to a traditional Relational SQL Database like MySQL.

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 how databases like MongoDB and redis would be classified in the NoSQL world

MongoDB - Document

A document data model is a storage of documents, where each document is a complex data structure, usually represented in JSON. In a document model, you can retrieve whole documents or partial data of a document, update documents etc. And in contrast to key/value databases, document databases can see structure within the aggregate.
A document database has no schema (just like the other NoSQL databases), however you will need to use some kind of schema, in our case in the form of mongoose.

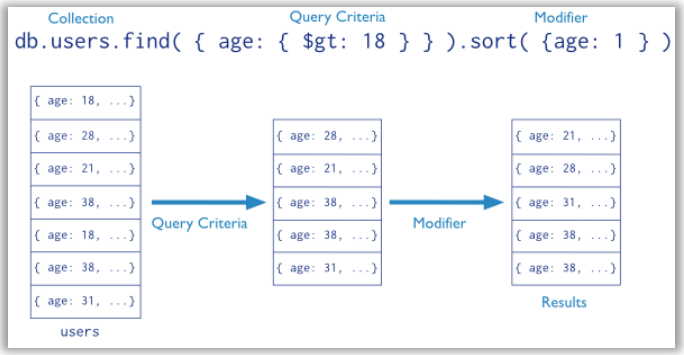
redis – Key/Value

Key/value store means that you have a key which is linked to a certain value in the database. The database doesn't know what the value is and it doesn't care. It could be an image, a complex document etc.
So Key-value databases like redis is like a hashmap but persistent in the disc.
It is also possible to store metadata about the value, which means that the difference between key-value databases and document databases isn't

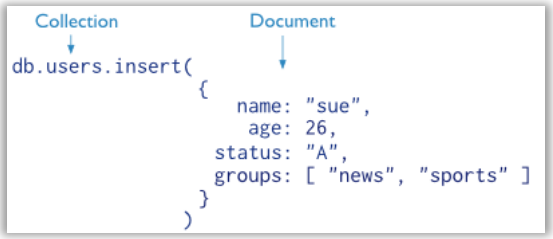
Explain, using relevant examples, the strategy for querying MongoDB (all CRUD operations)

In MongoDB a query targets a specific collection of documents. Queries specify criteria, or conditions, that identify the documents that MongoDB returns to the clients. A query may include a *projection* that specifies the fields from the matching documents to return. You can optionally modify queries to impose limits, skips, and sort orders.

Retrieving data:



Inserting data:



Demonstrate, using a REST-API, how to perform all CRUD operations on a MongoDB

Example: see MongoCrud.js