Relational vs Non-Relational Database

What is SQL?

SQL is a query language. SQL stands for Structured Query Language. Using SQL, you can interact with relational databases.

What is NoSQL?

NoSQL stands for "Not Only SQL". NoSQL refers to non-relational databases. In a non-relational database, you can add data entries that look different than other data entries making non-relational databases very flexible. (MongoDB, n.d.)

What are the key differences between relational and non-relational databases?

Both differ in the use cases they are used for. A brief comparison. (Coursera, 2024)

Feature	Relational Database	Non-Relational Database
Scalability	Vertical Scaling	Horizontal Scaling
Language	SQL	JSON, XML or YAML
Structure	Table based	Document-oriented, key-value pairs or graph structures

But according to AWS (Amazon Web Services, n.d.) it is not the fact if the database is relational or not which decides how well it performs and scales, but they refer to if the database is ACID compliant or not. Here is a brief comparison from AWS.

Quality	ACID	BASE
Scalability	Scales vertically.	Scales horizontally.
Flexibility	Less flexible. Blocks specific records from other applications when processing.	More flexible. Allows multiple applications to update the same record simultaneously.
Performance	Performance decreases when processing large volumes of data.	Capable of handling large, unstructured data with high throughput.
Synchronization	Yes. Adds delay when synchronizing.	No synchronization at the database level.

Should I use Relational or Non-Relational Database for my services?

Currently I have 3 services: **Order Management**, **Product Management**, **Search Engine**. Since I am using the database per service data pattern, each service will have its own database.

Before getting into each service, it is important to mention that scalability and performance play a huge role in my project.

Order Management Service

For the order management it is important to choose a Relational ACID database. Since it is very important that if some action fails the entire process fails. And it is a best practice to use a relational database for this case. Ideally I would choose an SQL ACID database that is able to scale horizontally, like CockroachDB, TiDB, Yugabyte. However, none of these are free to use. CockroachDB does have a 30-day free trial which could be useful.

Since for my project consistency is more important than scalability, I will choose an SQL ACID Database that has low response times and is free. This brings me to **PostgreSQL**. (Silnitsky, 2021)

Product Management Service

This database will receive the most load, therefore its key to choose a database which can handle that and also which is capable of scaling horizontally. It is a best practise to use a BASE-oriented database for products in ecommerce. **MongoDB** is a free BASE-oriented database. (Shopware, 2024)

Search Engine Service

To create a search engine service in ecommerce, the best practice is to use ElasticSearch or Solr in combination with my service. ElasticSearch has a free trial, but Solr is free to use. Therefore, I will use **Solr** for my search engine service (Silnitsky, 2021)