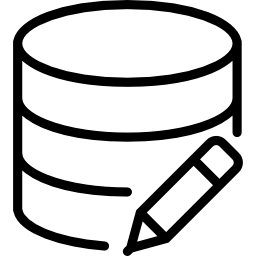
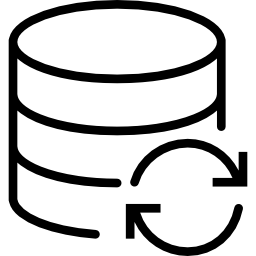
Database Research

# Relation- vs document-oriented databases

VS



There are two main types of databases. The first type, **relational databases,** present data in tables and rows. These tables are connected on a mathematical basis, the relational theory. Most relational databases use Structured Query Language or **SQL** to allow creation and modification of its content.

The second type are **document-oriented databases**. This type represents data in collections of documents, using data interchange languages such as **JSON** and **XML**.

There exist multiple examples of database programs of both types.

* Popular examples of Relation-oriented databases are **MySQL, Microsoft SQL Server and SQLite.**
* Popular examples of Document-oriented databases are **MongoDB, Couchbase and Clusterpoint.**

The database needed for our project will be used for storing **account information**. Moreover, it will be used to store the different **events and restaurants for each manager**, along with the information, statistics, menus and orders for the corresponding restaurants/events.

We have researched the two most popular programs of each type and documented the differences below.

## MongoDB

MongoDB is the most widely used document-oriented database program at this moment. It does not use a traditional table-based structure, linked by relations. Instead, it uses documents with dynamic schemas in a custom BSON format.

**Replication**: For redundancy, same sets of data on multiple servers -> if one is down another will be elected as primary data. (MongoDB documentation)

**Load balancing**: For systems with large data sets and/or high usage it might be interesting to distribute this load over multiple systems. MongoDB provides a sharding system to allow for horizontal scaling. (MongoDB documentation)

**Availability:**MongoDB is available for free under open licensing.

## NoSQL //Maybe combine with part on MongoDB//

Why use NoSQL:

* Simpler design approach than SQL.
* Easier to scale horizontally.
* Certain operations are faster or more flexible in NoSQL

But:

* No joins across tables.
* Lack of standardized interface.
* Use of low-level query languages.
* More difficult to transition to from an existing relational database.

## MySQL (//need more information about why someone would use, or not use, SQL)

MySQL is an open-source relational database management system

## Conclusion

//don’t forget to add all used sources//