# What Is a Database?

Databases generally fall into two main categories: SQL and NoSQL. Each type has its own strengths and use cases, which we'll explore in more detail. Let's break down these two categories to understand their unique characteristics and applications.

**por Mayko Silva**

# SQL Databases

## Structured Query Language

SQL databases use a special language to interact with data. For example, to find a birthday, you might use a command like:

```
SELECT * FROM birthdays WHERE person = 'husband';
```

Don't worry if this looks confusing - we'll cover SQL syntax in future lessons!

## Relational Structure

SQL databases are often called relational databases because they're composed of interconnected tables. These tables use predefined schemas, which determine the structure and organization of the data before any information is added.

# NoSQL Databases

## Flexible Structure

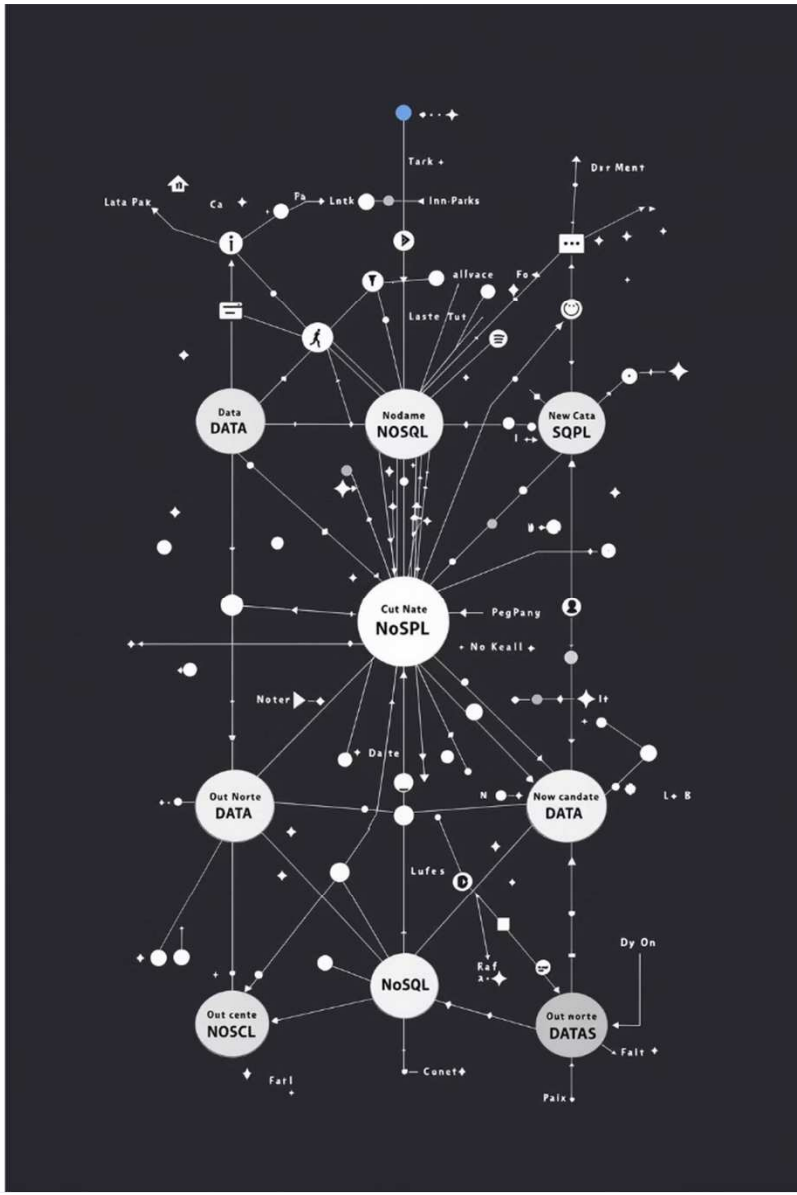
NoSQL databases offer more flexibility when data doesn't fit neatly into predefined structures. They don't require you to define the data structure upfront, allowing for more adaptable data storage and access.

## Scalability

These databases are designed to handle huge amounts of data efficiently, making them ideal for applications that need to scale quickly or deal with varied data types.

## "Not Only SQL"

NoSQL stands for "Not Only SQL," indicating that while these databases differ from traditional SQL databases, they can still incorporate SQL-like query languages in some cases.



# Key Takeaways

1

## Database Basics

A database is an organized place to store data, designed for efficient storage and retrieval of information.

2

## SQL Databases

SQL databases use predefined structures and are excellent for data that fits neatly into tables with clear relationships.

3

## NoSQL Databases

NoSQL databases offer more flexibility and can handle less structured data, making them ideal for certain types of applications and data sets.

In our next lesson, we'll delve deeper into SQL databases and learn more about working with them. For now, consider the different types of data you encounter daily and whether they might be better suited for a SQL or NoSQL database.