What’s A Database?



Probably this what comes into your mind when you hear the term “database”.

It’s a structured system to put your data in that imposes rules upon that data, and the rules are yours, because the importance of these problems changes based on your needs. Maybe your problem is the *size*, while someone else has a smaller amount of data where the *sensitivity*is a high concern.

It’s the things you can’t see that are going on in the background; the security, the enforced integrity of the data, the ability to get to it fast and get to it reliably, the robustness; serving lots of people at the same time and even correctly survive crashes and hardware issues without corrupting the data.

And that’s what we need to do here; understand how to describe our structure and define those rules, so all these invisible things will actually happen.

Database Management System (DBMS)

We often mistakenly say our database is *Oracle, MySQL, SQL Server, MongoDB*. But, they aren’t databases, they are database management systems (DBMS).

The DBMS is the software that would be installed on your personal computer or on a server, then you would use it to manage one or more database.

The database has your actual data and the rules about that data, while the DBMS is the program that surrounds and manages your actual data, and it enforces the rules you specified on your data. The rules for example could be the type of the data, like integer or string, or the relationship between them.



Database Management System (DBMS)

In practice it’s very common to have multiple databases. The database that deals with your order and customer information might be completely independent from you database that deals with human resource information. And in many organizations, you don’t just have multiple databases but multiple DBMS. Sometimes it’s because one DBMS is better at something than the other.

There are different DBMS, and they are categorized under:

* Relational Database Management Systems
* Hierarchical Database Systems
* Network Database Systems
* Object-Oriented Database Systems
* NoSQL Database Systems

We are going to focus on the relational database management systems (RDBMS). And here’s *Why? …*

* They are the most common used one.
* The principles we are going to discuss here are usable across all of them.
* If you know you are going to jump into NoSQL databases, most of the introductions assumes you already understand relation database concepts and will use these concepts to explain what’s offered by NoSQL databases.

RDBMS are like Oracle, MySQL, SQL Server, SQLite, DB2, …etc.