

Intro to Data Modeling

Tuesday, April 28, 2020 5:07 PM

Databases: A [database](#) is a structured repository or collection of data that is stored and retrieved electronically for use in applications. Data can be stored, updated, or deleted from a database.

Database Management System (DBMS): The software used to access the database by the user and application is the database management system. Check out these few links describing a DBMS in more detail.

Key points about Data Modeling

- **Data Organization:** The organization of the data for your applications is extremely important and makes everyone's life easier.
- **Use cases:** Having a well thought out and organized data model is critical to how that data can later be used. Queries that could have been straightforward and simple might become complicated queries if data modeling isn't well thought out.
- **Starting early:** Thinking and planning ahead will help you be successful. This is not something you want to leave until the last minute.
- **Iterative Process:** Data modeling is not a fixed process. It is iterative as new requirements and data are introduced. Having flexibility will help as new information becomes available.

Advantages of Using a Relational Database

- **Flexibility for writing in SQL queries:** With SQL being the most common database query language.
- **Modeling the data not modeling queries**
- **Ability to do JOINS**
- **Ability to do aggregations and analytics**
- **Secondary Indexes available :** You have the advantage of being able to add another index to help with quick searching.
- **Smaller data volumes:** If you have a smaller data volume (and not big data) you can use a relational database for its simplicity.
- **ACID Transactions:** Allows you to meet a set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, and thus maintain data integrity.
- **Easier to change to business requirements**

When Not to Use a Relational Database

- **Have large amounts of data:** Relational Databases are not distributed databases and because of this they can only scale vertically by adding more storage in the machine itself. You are limited by how much you can scale and how much data you can store on one machine. You cannot add more machines like you can in NoSQL databases.
- **Need to be able to store different data type formats:** Relational databases are not designed to handle unstructured data.

- **Need high throughput -- fast reads:** While ACID transactions bring benefits, they also slow down the process of reading and writing data. If you need very fast reads and writes, using a relational database may not suit your needs.
- **Need a flexible schema:** Flexible schema can allow for columns to be added that do not have to be used by every row, saving disk space.
- **Need high availability:** The fact that relational databases are not distributed (and even when they are, they have a coordinator/worker architecture), they have a single point of failure. When that database goes down, a fail-over to a backup system occurs and takes time.
- **Need horizontal scalability:** Horizontal scalability is the ability to add more machines or nodes to a system to increase performance and space for data.
-