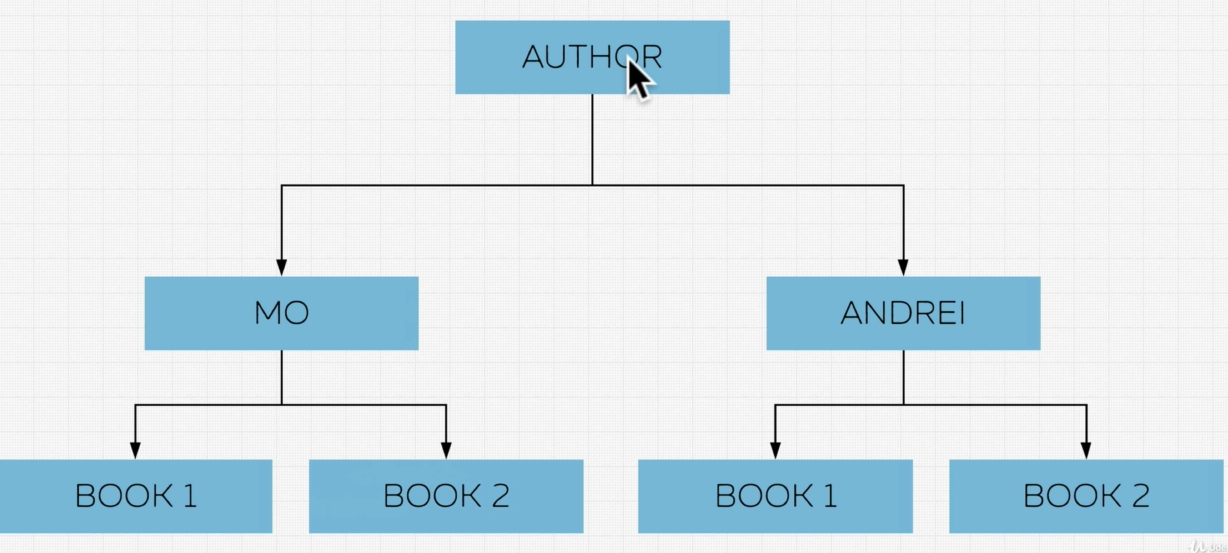
* **Database** is a collection of data that let users store, organize and use data.
* **Database management system (DBMS)** gives us functions/ways to CRUD operation on data.
  1. These follows a specific model.
  2. It manages/stores the data.
  3. It will secure the data with security policy and authentication
  4. It also manages transaction.
* **Types of databases**:

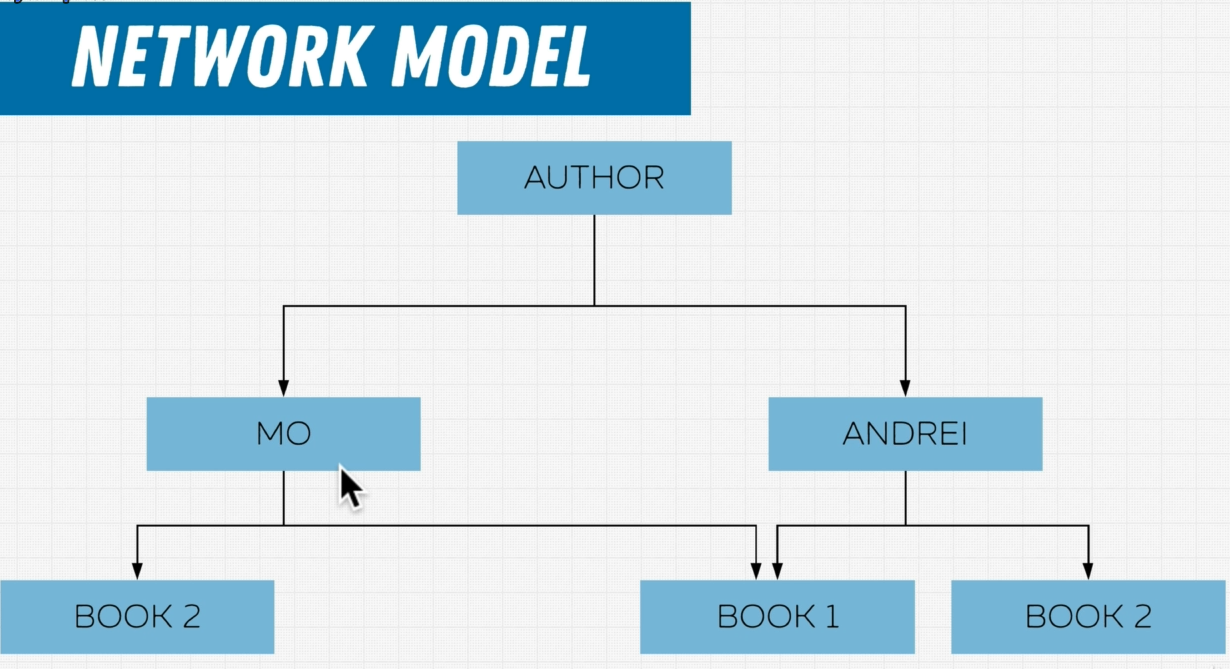
1. Relational - MySQL, PostgreSQL, MSSQL
2. Document - MongoDB
3. Key-Value – Redis, Dynamo-DB
4. Graph – Neo4j,
5. Wide columner

* **SQL-** Structured query language. Used for operating on relational database.
* **Query-** also known as SQL statement.
* **Declarative programming** is a programming paradigm … that expresses the logic of a computation without describing its control flow.
* **Imperative programming** is a programming paradigm that uses statements that change a program’s state
* Codd's Rules for RDBMS: <https://www.w3resource.com/sql/sql-basic/codd-12-rule-relation.php>
* **Database Models**

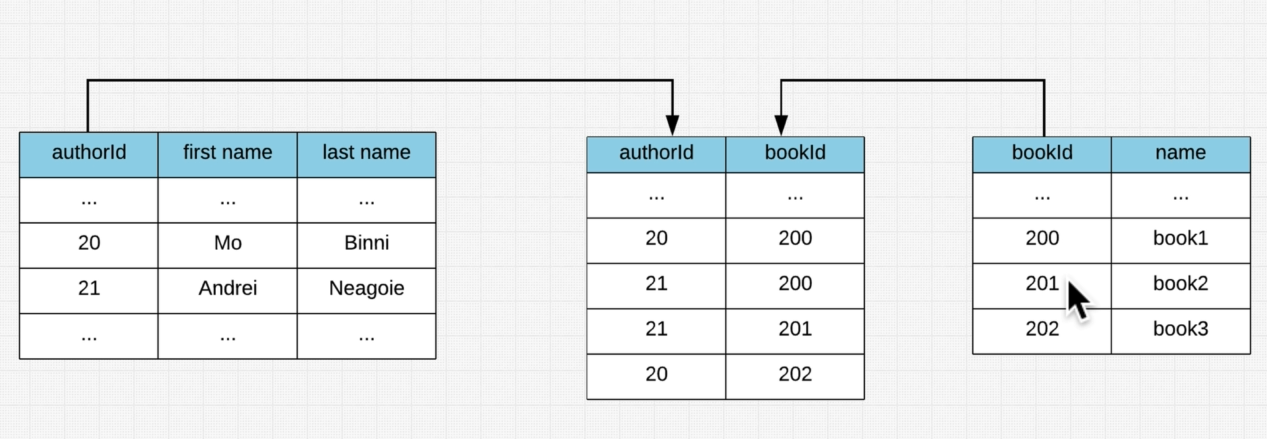
1. Hierarchical model: here relationship between parent children is one-to-many. Parent can have many children, but children can have only one parent.



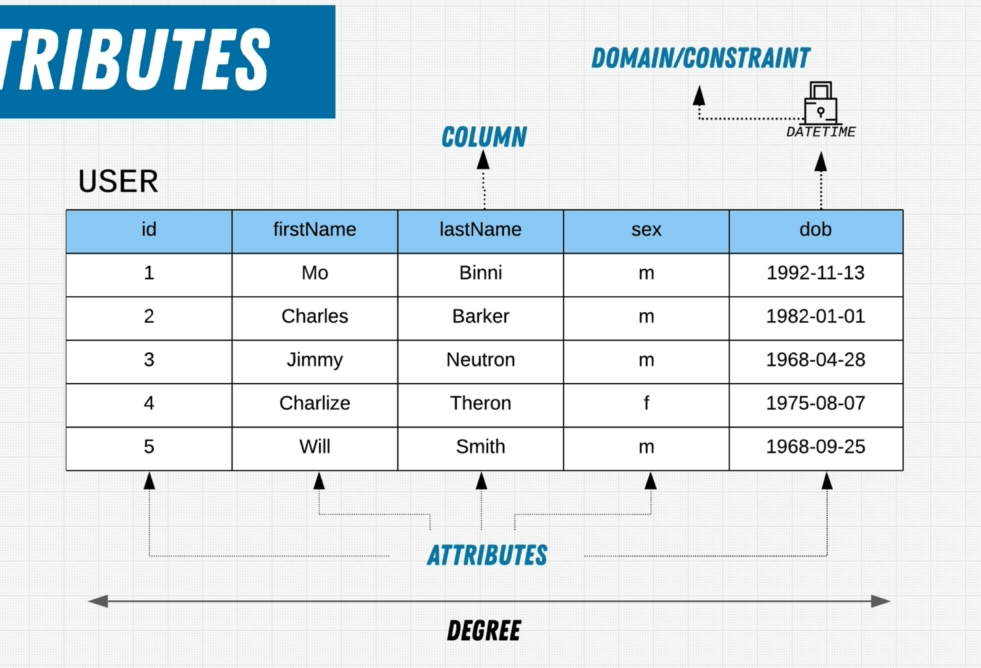
1. Networking model: here relationship between parent children is many-to-many.



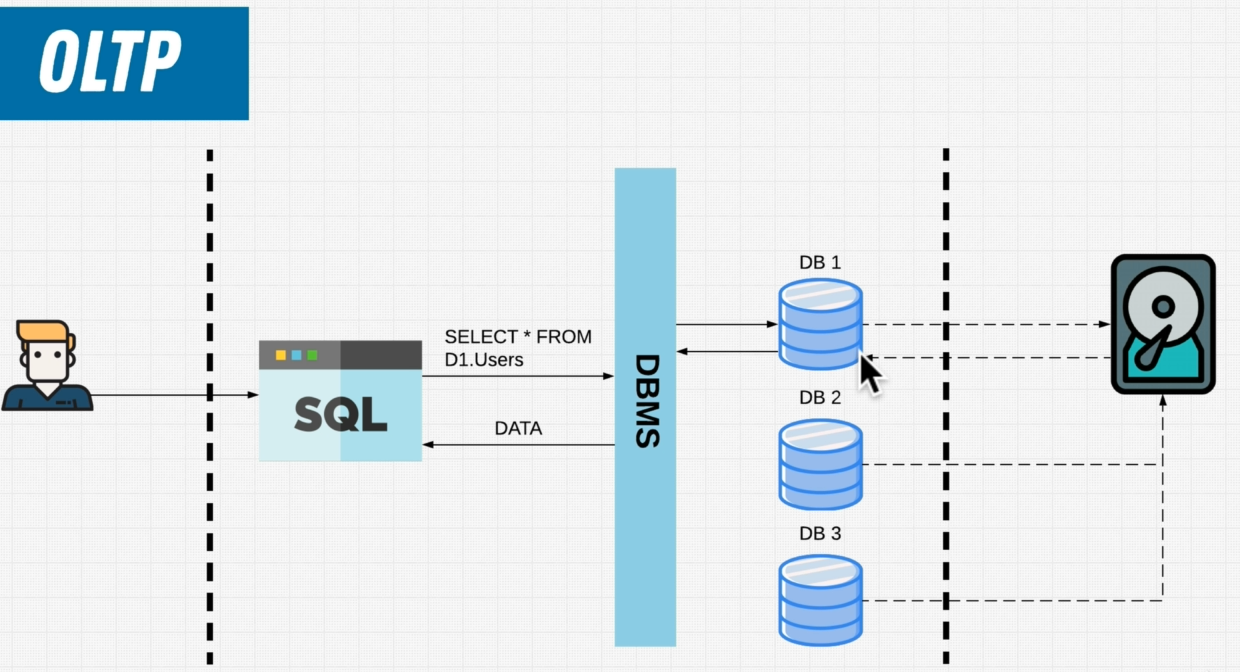
1. Relational model: It follows a tables/relation structure.



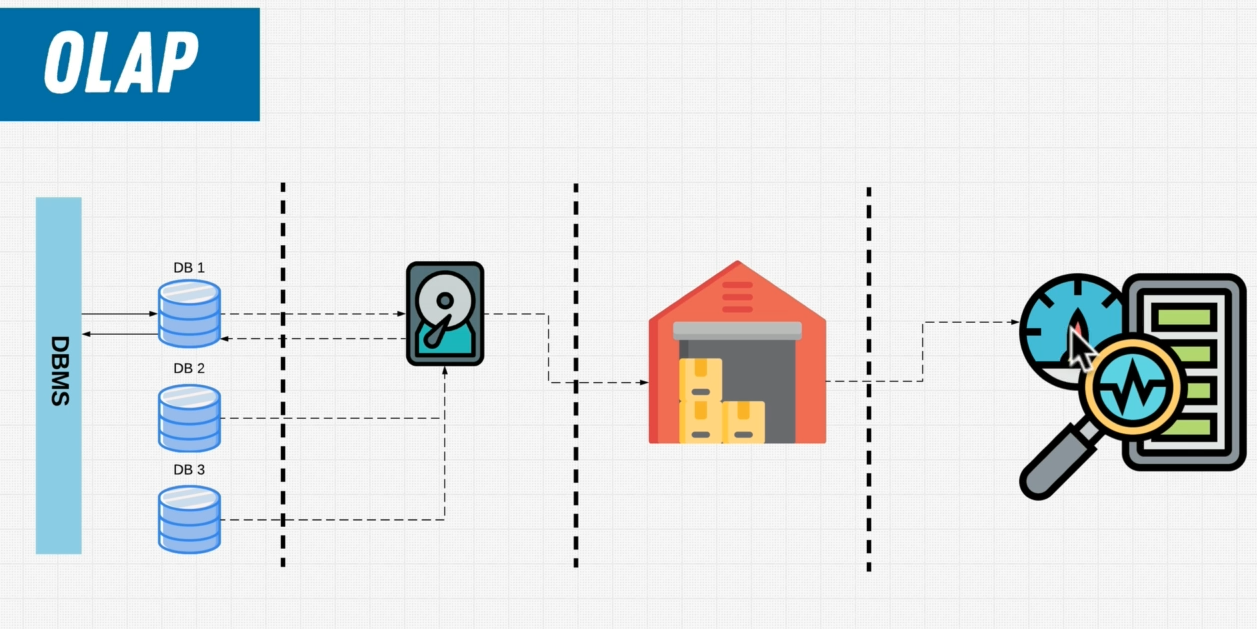
* **Table:** table has a schema
* **Domain/Constraint:** It’s the constraint of the data, it defines what data can go in.
* **Columns/attributes:** Columns that stores specific type of data.
  + **Degree:** Collections of columns are called degree.
* **Rows/Tuples:** One row of data is called tuple, multiple rows of data are called tuples. All row/tuple follow the column constraint.
  + **Cardinality:** Number of rows in a table is called cardinality.



* **Primary key:** It is unique identifier, can’t be null. Each table can have only 1 primary key.
* **Foreign key:** Foreign key is a column (or collection of colums) in a table, that are primary key in another tale.
* Online analytical processing (**OLAP**):



* Online Transactional Processing **(OLTP**):



* **Aggregate function:** It runs against all data, and produce one output. It operates on many records to produce one value. Example: min(), average()
* **Scalar function:** It runs against each individual row independently. Example: concat()