DBMS Database Models

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the **Relational Model** is the most widely used database model, there are other models too:

- Hierarchical Model

- Network Model

- Entity-relationship Model

- Relational Model

Hierarchical Model

This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The heirarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.
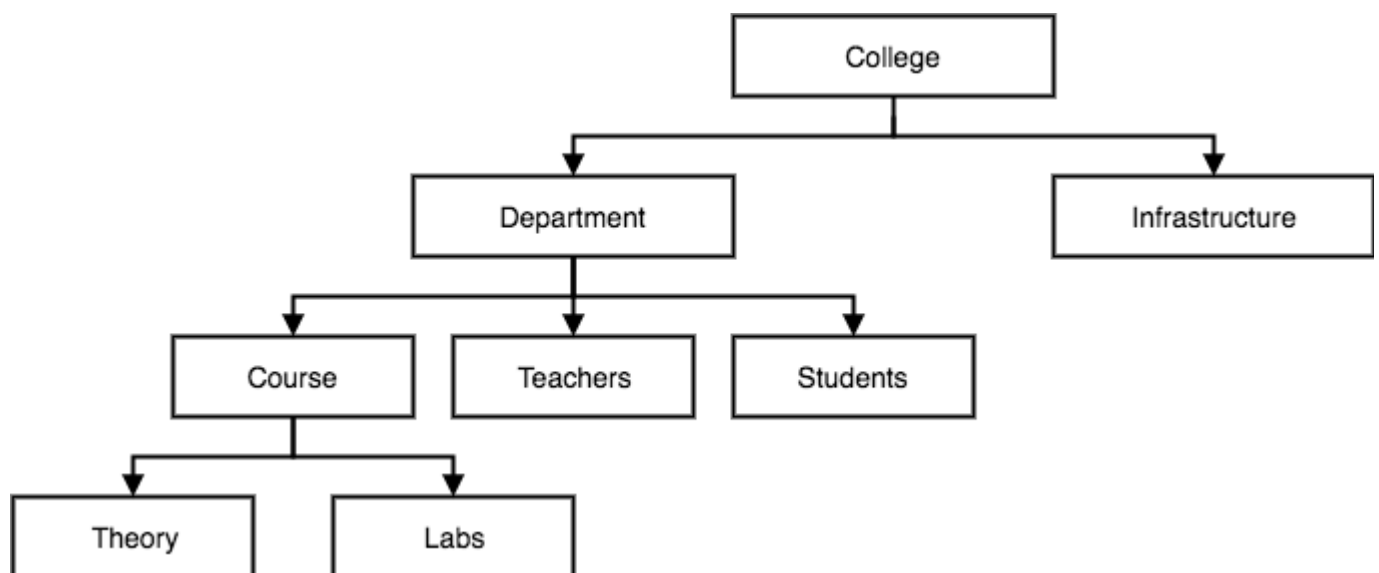
Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

This was the most widely used database model, before Relational Model was introduced.
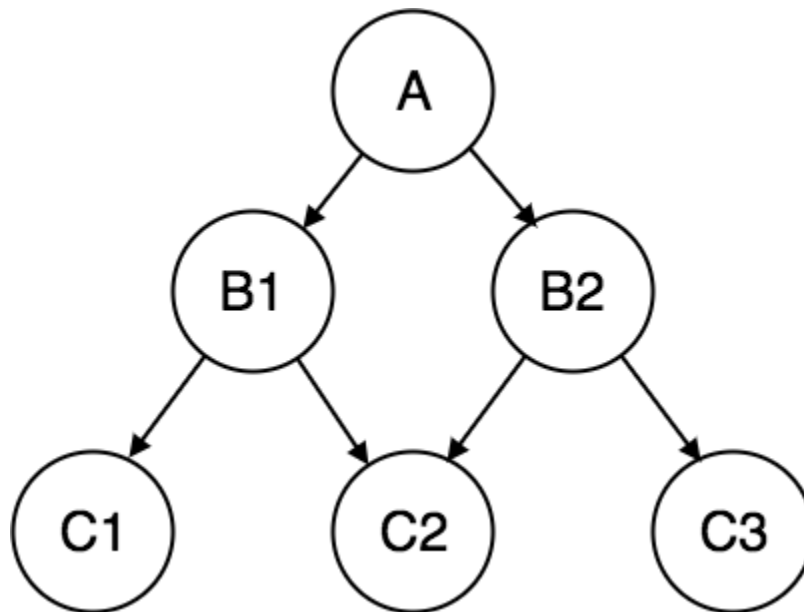
Entity-relationship Model

In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.

Different entities are related using relationships.

E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.

This model is good to design a database, which can then be turned into tables in relational model(explained below).

Let's take an example, If we have to design a School Database, then **Student** will be an **entity** with **attributes** name, age, address etc. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pincode, city etc, and there will be a relationship between them.
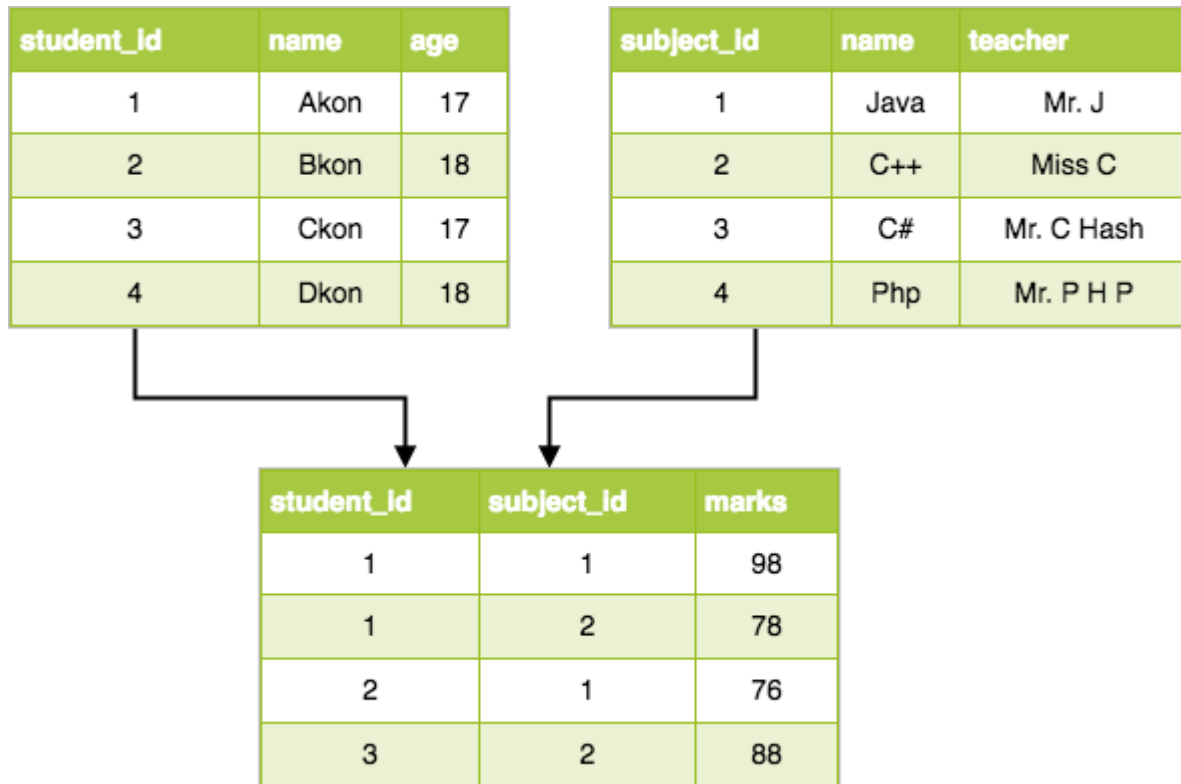
Relationships can also be of different types.
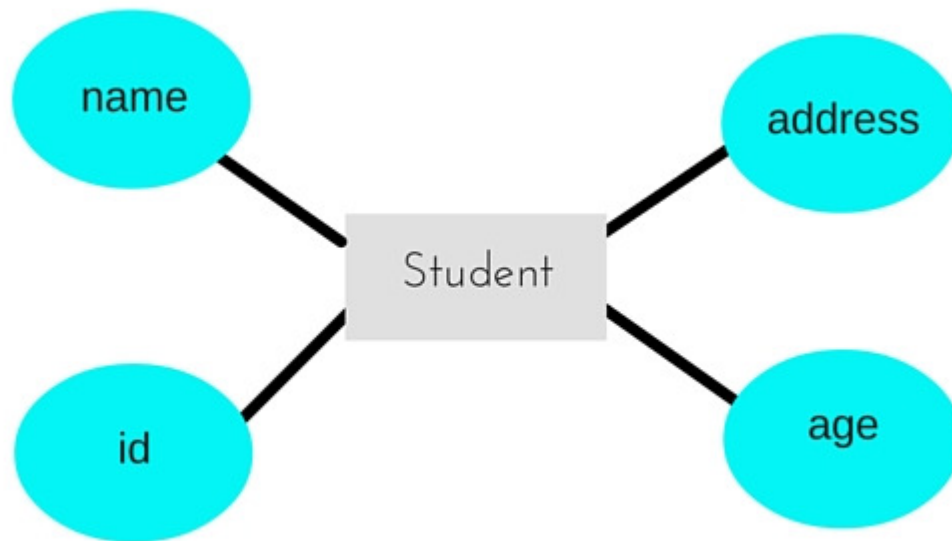
Relational Model

In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field.

This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.

Hence, tables are also known as **relations** in relational model.

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Database Language

- o A DBMS has appropriate languages and interfaces to express database queries and updates.
- o Database languages can be used to read, store and update the data in the database.

Types of Database Language
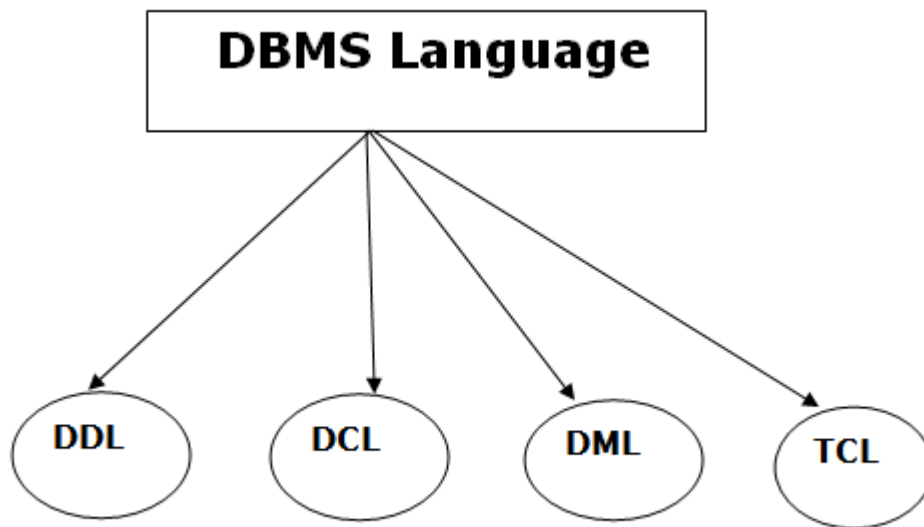
1. Data Definition Language

- o **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
- o It is used to create schema, tables, indexes, constraints, etc. in the database.
- o Using the DDL statements, you can create the skeleton of the database.
- o Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

- o **Create:** It is used to create objects in the database.
- o **Alter:** It is used to alter the structure of the database.
- o **Drop:** It is used to delete objects from the database.
- o **Truncate:** It is used to remove all records from a table.
- o **Rename:** It is used to rename an object.
- o **Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

2. Data Manipulation Language

DML stands for **D**ata **M**anipulation **L**anguage. It is used for accessing and manipulating data in a database. It handles user requests.

Here are some tasks that come under DML:

- o **Select:** It is used to retrieve data from a database.
- o **Insert:** It is used to insert data into a table.
- o **Update:** It is used to update existing data within a table.
- o **Delete:** It is used to delete all records from a table.
- o **Merge:** It performs UPSERT operation, i.e., insert or update operations.
- o **Call:** It is used to call a structured query language or a Java subprogram.
- o **Explain Plan:** It has the parameter of explaining data.
- o **Lock Table:** It controls concurrency.

3. Data Control Language

- o **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
- o The DCL execution is transactional. It also has rollback parameters.

    (But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

- o **Grant:** It is used to give user access privileges to a database.
- o **Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

4. Transaction Control Language

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

- o **Commit:** It is used to save the transaction on the database.
- o **Rollback:** It is used to restore the database to original since the last Commit.