Data Models

Data models are logical representation and diagrammatic view of the database. It gives us an idea and layout how data are stored, maintain, access, update and how data are connected with each other in the system.

Some of the most widely used data models are:--

1. Entity- relationship model
2. Relational model
3. Hierarchical mode
4. Network mode
5. Object- oriented data model
6. Object-relational data model

Entity- relationship model:--this model represents real world problems and also the relationship among different real world entities. ER diagram has three major components

Entity: -- Entity represents real world things or objects such as person, place, event etc.

Attribute: -- attributes are the real world property. Attributes are the characteristics of the entity such as a teacher is an entity which have attributes (property) such as teacher_id, teacher_name etc

Relationship: -- relationship represent how two entity are related in real – world. Such as a teacher works for a department. So both teacher and department have relationship.

There are 4 types of relationship or mapping which are

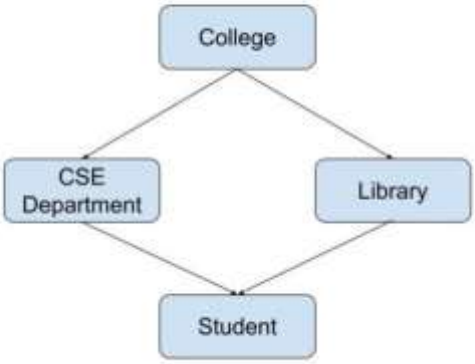
1. One to one
2. One to many
3. Many to one
4. Many to many

Relational model:--in relational model data are maintained into two dimensional tables. All the data are maintain in the forms of row and columns.

Hierarchal model: -- hierarchal model is the first data model of DBMS. This model looks like a tree structure but not a binary tree. It has a single root node and from there child nodes are added and expand. One child can have only one parent but apparent can have more than one child.

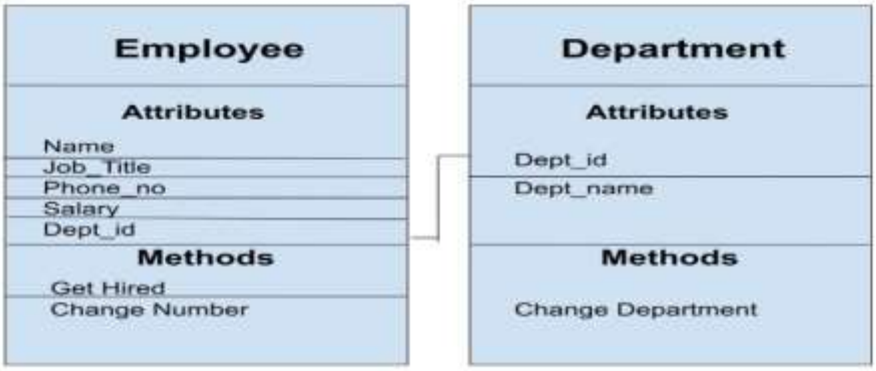
Ex are: -- site map of a website, food recipes etc.

Network model: -- we can say network model is the extension of hierarchal model. Only difference is that in network model a child node can have more than one parent.



Network Model

Object- oriented data model: -- in this model real- world problem are represent in object form and make relationship between them. Here real world things are considered as objects. We can store audio, video in database using this model which cannot be done by ER model (however we can store audio and video using ER model but it recommend not to use it for such operations) we use links to relate the objects.



Object_Oriented_Model

Schema in DBMA

In single word Schema is the design of a database.

Schema is a blue print or architectures of a database. It is a abstract design which represent storage of the data in database and also organization and relationship of the data table. It is a layout or structure how we going to store the data in database

Developers create schema before actually design the database so they will know what necessary component it require and how they are related to each other.

There are three types of schema present in DBMS

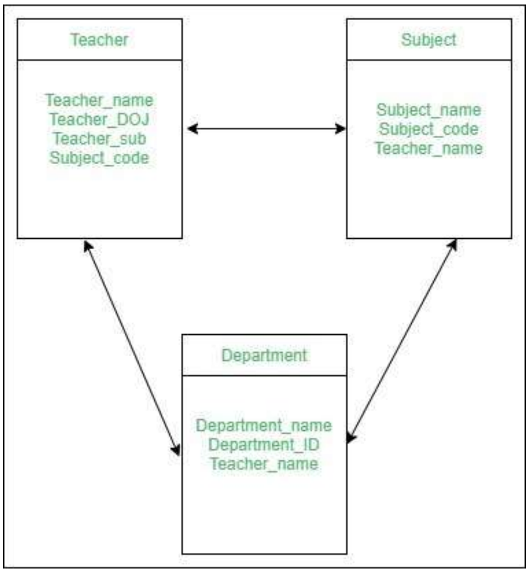
1. Logical schema
2. Physical schema
3. View schema

Logical Schema:--in logical schema describe the database for the community of users. It hides the physical structure and internal implementations of the database. It represents data type, tables, relationship, attributes, entity etc.

Physical Schema: -- physical schema describe how actually data are stored in the system. We can say that it is the actual code which is used to create the structure the data.

View Schema: -- view schema describe external view for different end users with the access of specific part of the database.

The following fig is the example of a achema



Instance of a database:--instance is the collection of the data or information of a database from a particular time period. It can change depending CRUD operations.

Example –
Let’s say a table teacher in our database whose name is School, suppose the table has 50 records so the instance of the database has 50 records for now and tomorrow we are going to add another fifty records so tomorrow the instance have total 100 records. This is called an instance.

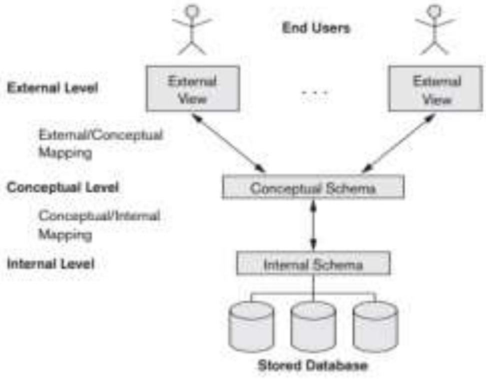
3-Tier Architecture in DBMS

or

Three-Schema Architecture

This is architecture of database. Which contain three independent layers.

Figure 2.2
The three-schema architecture.



Three layers are:--

1. Internal level
2. Conceptual level
3. External level

Internal level has physical schema which describe physical storage of the data

Conceptual level has logical schema which describe the logical structure of the data. It hide physical schema an internal implementation

External level has view schema. It describes a part of the database for some specific group of users.

Data Independence: -- data dependency defines changing in one level or schema should not affect the other levels or schema.

Any change of data in physical level should not affect conceptual or external level. We may change the conceptual level schema to expand the database by adding new row and columns but this should not affect the external schema. End users no need to worry about the changes even they don't know about that changes.

ACID property

ACID

1. Atomicity
2. Consistency
3. Isolation
4. Durability

Atomicity states that all transections must execute at once or none. Transection should not execute partially. Between transection is something wrong happen then entire transection must roll back to its initial condition and also the database.

Consistency state that if any change made in database then that change should remain preserved. Database need to remain consistency after and before transection. If database was in a consistent state before transection then it must be remain consistent after transection.

Isolation state that change in one state doesn't affect the other state. It state that second operation on a database should begin after the first operation execute completely. Any change occurs due to a transaction will not affect the other transection until change is committed in memory.

The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.