Data Base Management Systems

Syllabus

Module 1 :- Conceptual Modeling Introduction

Introduction to Database : purpose of Database Systems, View of data, data models, Database languages, Database users, Various components of overall DBS architecture, Various Concepts of ER models, basics of Relational Model.

Introduction to Database :-

Data :- Data is a single unit of information

(or)

All the single items that are stored in a database, either individually or as a set.

(or)

Data is a collection of raw facts and figures.
Word 'Data' is originated from the word 'datum' that means 'single piece of information'.

For example :-

If we want your name, age, height etc that is related to you, so this data helps to create some information about you.

Database :- It is a collection of interrelated data.

(or)

A database is a location where we store data that data should a systematic manner, this store data will manage by the database usess.

(or)

A database is an organized collection of data, so that it can be easily accessed and managed.

You can organize data into tables, rows, columns and index it to make it easier to find relevant information.

DBMS :- Database Management Systems

It is a Software application, it help users to access the database easily where users can easily store, modify and extract infosmation from a database as per requirement.

(or)

A DBMS is a software which is used to store and manage the database.

For example :-

Mysql, Oracle, IBMDB2, microsoft sql Server, Sqlite. are a very popular commercial database which is used in different applications.

Database Management system provides an interface to perform various operations like database creation storing data in it, updating data, creating a table in the database and a lot more.

Example for database :-

The college Database organizes the data about the admin, staff, students and faculty etc. Using the database, you can easily retrieve, insert, & delete the infosmation.

Database System Application :-

→ Database management system contains information about a particular enterprise.

1) Collection of interrelated data
2) Set of programs to access the data
3) An environment that is both convenient and efficient to use.

→ 1. Enter price information :-

Sales : customers, products, purchases.

Accounting : payments, receipts, account balance, assets.

Human Resources : Employee records, salaries, tax deductions

Manufacturing : production, inventory, orders, supply chain.

Online Retails : order tracking, customized recommen-dations.

2. Banking & Finance :- All transactions.

Credit card : Generation of monthly statements

Finance : storing information about holdings & sales

3. Universities :- Registration, grades

4. Airlines :- Reservations, Schedules

5. Telecommunications : Keeping records of calls made, generating monthly bills.

What is management system?

A management system is the way in which an organization manages the interrelated parts of its business in order to achieve its objectives.

It is important because without the existence of some kind of rules and regulations it is not possible to maintain the database.

Purpose of database management systems :-

The dbms is defined as a s/w system that allows the users to define, create & maintain the database & provide control access to the data

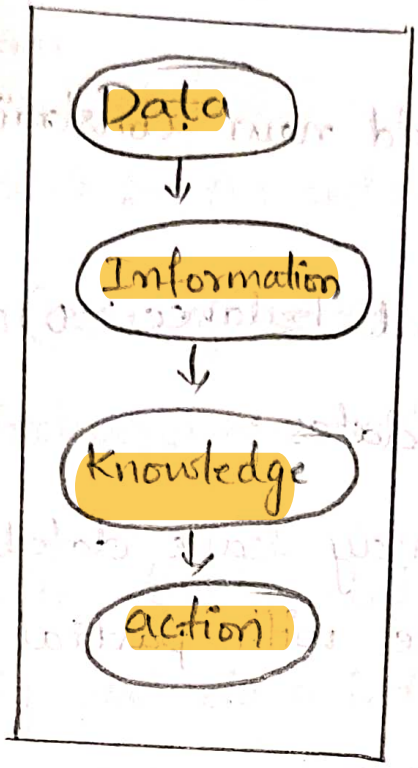
It is a collection of programs used for managing data & it supports different types of users to create, manage, retrieve, update and store information.

Purpose :-

The purpose of dbms is to transfor the following

1) Data into information
2) Information into knowledge
3) Knowledge to the action.

The diagram explains the process to how the transformation of data to information to knowledge to action happens respectively in the DBMS.



In the early days, the database application were built directly on top of the file system.

Drawbacks :-

1) Data redundancy & inconsistency :
   Multiple file formats, duplications of information in different files.
2) Difficulty in accessing data :
   Need to write a new program to carry out each new task.
3) Data Isolation :
   Multiple files and formats.

4) Integrity problems:

Integrity constraints become "buried" in program code rather than being stated explicitly

Hard to add new constraints or change existing ones.

(eg. account balance > 0)

5) Atomicity of updates:

Failures may leave database in an inconsistent state with partial updates carried out

Example:- Transfer of funds from one account to another should either complete or not happen at all or incomplete.

6) Concurrent access by multiple users:

Concurrent accessed needed for performance uncontrolled concurrent accesses can lead to inconsistencies

Example:- Two people reading a balance and updating it at the same time

7) Security problems:

Hard to provide users access to some, but not all data

* Database system offer so many solutions to all these problems.

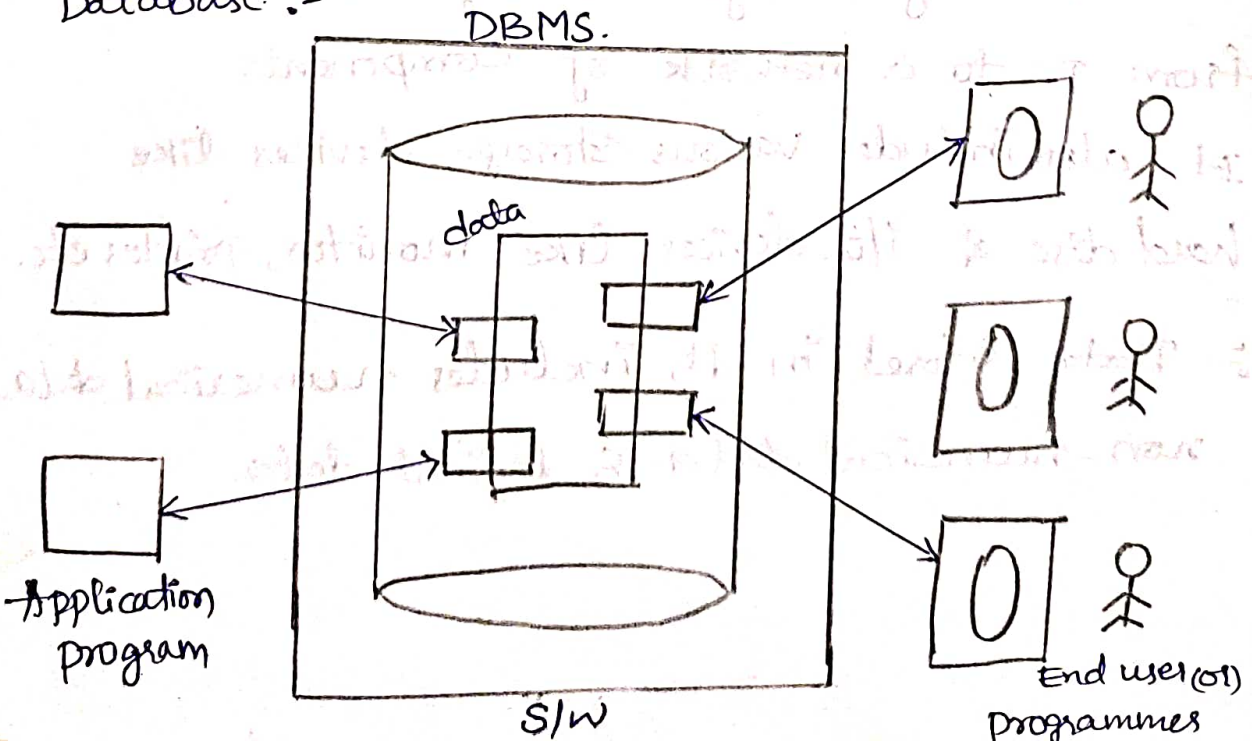
Uses of DBMS :- Advantages

The main uses of DBMS are as follows:-

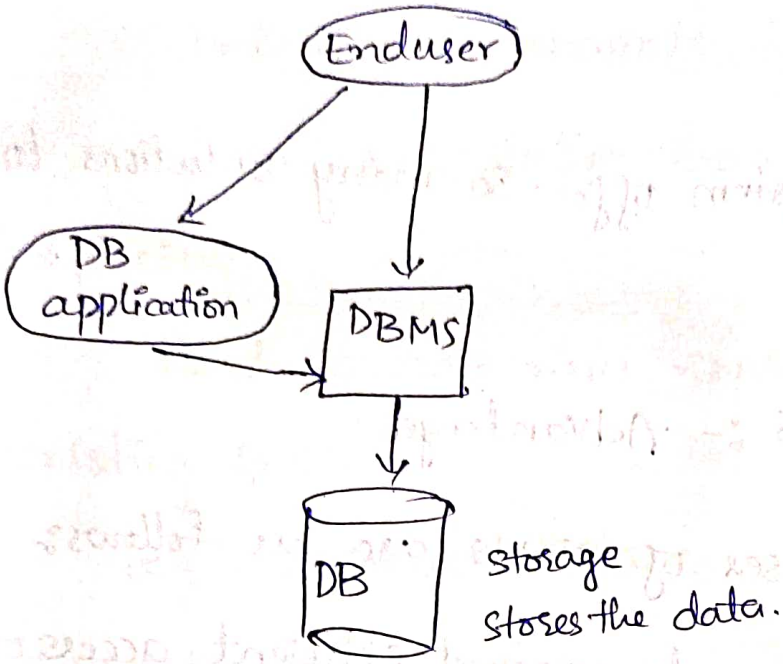
1) Data independence and efficient access of data
2) Application development time reduces.
3) Security and data integrity
4) Uniform data administration
5) Concurrent access & recovery from crashes

Conti DBMS is a S/W oracle, Mysql, sql, PLsql] db
Application prog c, C++, Java

Database :-



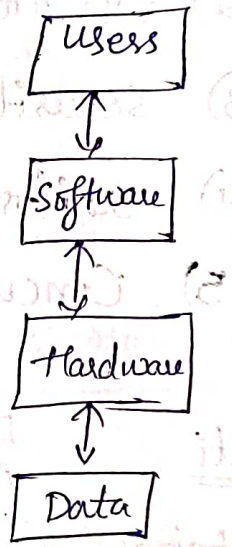
Example :- Database



Components of DBMS :-

Usess :- Application programmers, End uses, db administration

Software :- Controls the organization, Storage, management & retrieval of data in db



Hardware :- Hlw of a sys can range from pc to a network of components. It also include various storage devices like hard disc & i/o devices like monitors, printers etc.

Data :- Data stored in db includes numerical data, non-numerical data & logical data.

Building blocks of database :-

1) columns / fields
2) Rows / tuples / record
3) Tables

→ Why not use dbms :- disadvantage

The overhead cost of using dbms

1) High initial investment in h/w, s/w & training
2) Cost of defining & processing data
3) Overhead for security, Concurrency control, recovery.

Hence, it may be more desirable, to use regular files under

- Simple well defined db application that are not expected to change
- No multiple uses access to data.

View of Data

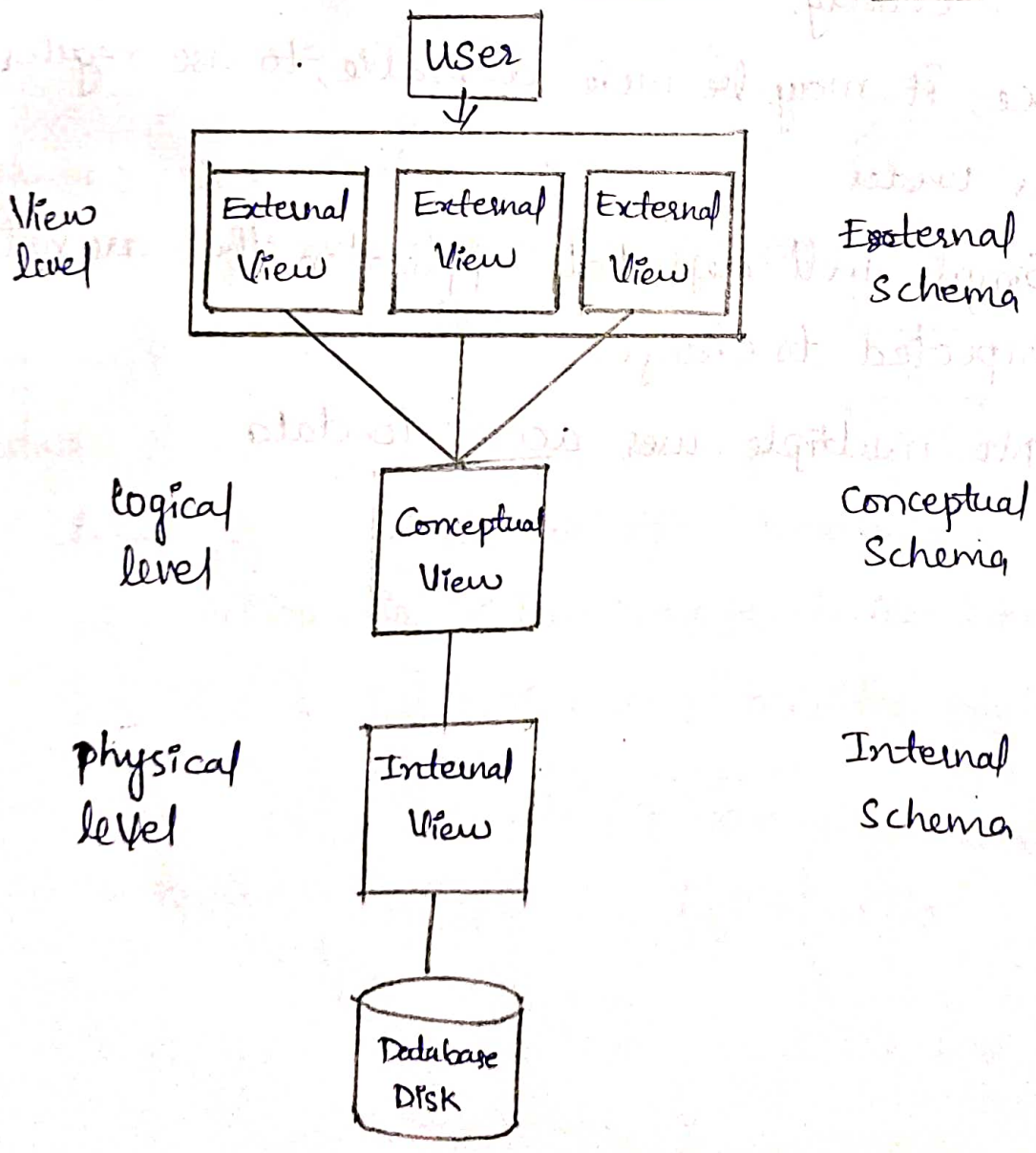
A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored & maintained.

Abstraction view is a process of hiding unwanted or irrelevant details from the end user

(or)

Abstract view of data refers to data hiding.

Levels of Abstraction in DBMS



View level :- Entire database can view.

This level describes that part of db that is relevant to each user.

Logical level :- What data is stored

This level describes what data is stored in db & relationship among the data.

→ All entities, attributes & their relationship
→ Constraints on the data
→ Security & integrity information

physical level :- How data is stored.

It covers the data structures & file organization

Example :- College Database.

1) External Schema (View)

Course_info (Cid : string, enrollment : integer)

2) Conceptual Schema (logical)

Students (Sid : string, name : string, login : string, age : integer, cgpa : real)

3) physical schema (physical)

→ Relations stored as unordered files.
→ Index on first column of students.

Instances :-

The collection of information stored in the database at a particular moment is called an instance of the database.

At any instant of time the content of the database is called instance. i.e because of performing insert / delete / update operations on a database, the content of the db get changed from time to time after every operation

* frequent change

Schema :-

The overall design of the database is called the database schema

* No frequent change

Eg:- Student

Sid num (5)
Sname char (25)
class char (10)

View schema :- How present the user data

logical schema - How organized data

physical schema :- where data stored in physical stored

Data Models

Data Models :-

Data Models are collection of tools for describing data, data relationships, data semantics, data constraints.

1) Relational Model :-

The relational model uses a collection of tables to represent both the data & the relationships among those data. Each table has multiple columns and each columns has a unique name. Tables are also called known as relations. (Data Table)

Sid	name	login	age	gpa
500	ABC	abc@ds	18	7.5
501	Mani	Mani@123	18	7.8
502	Nani	nani12@	18	8.2

Tuples
Records
Rows

Fields, Attributes, Columns.

2) Entity-Relationship model :

Data is represented as a collection of entities and relationship among the entities

* Entity will be a thing or object in the real world

* Each entity is described by set of attributes.

Example :- customer is entity

custid, cust name, cust street, cust city.

* The set of all entities of same type are called the entity set.

* The set of all relationship of same types is called the relationship set.

* The overall structure of a database can be expressed graphically by E-R diagram.

Where Rectangles - represent entity

Ellipses - represent attributes

Diamonds - relationship among entity

lines - link the entites set of relationship set.

3) Object Based data models :-

(Object-oriented and Object-relational) -

Inherited properties - like Java, python lang.

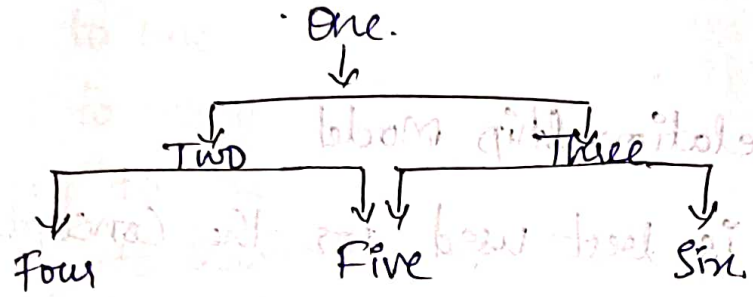
4) Semi-structured data model :-

web pages (HTML, XML)

other older models

5) Network model :-

Collection of records, relationship among data are represented by links, collection of records are organised as an arbitrary graph.



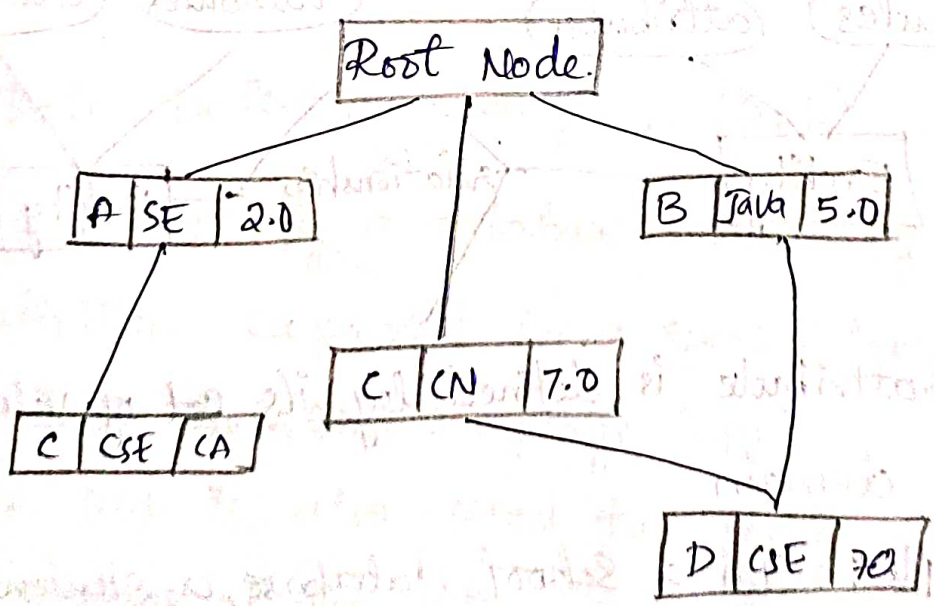
Trees & graphs

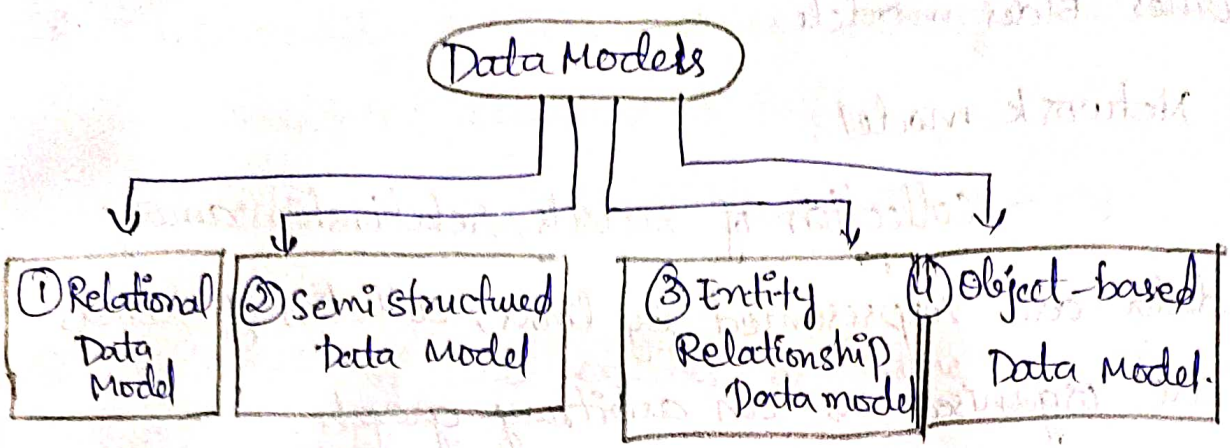
structure like Tree & graphs

6) Hierarchical model :-

It is similar to network model.

* organizes the collection of records as trees, rather than arbitray graphs





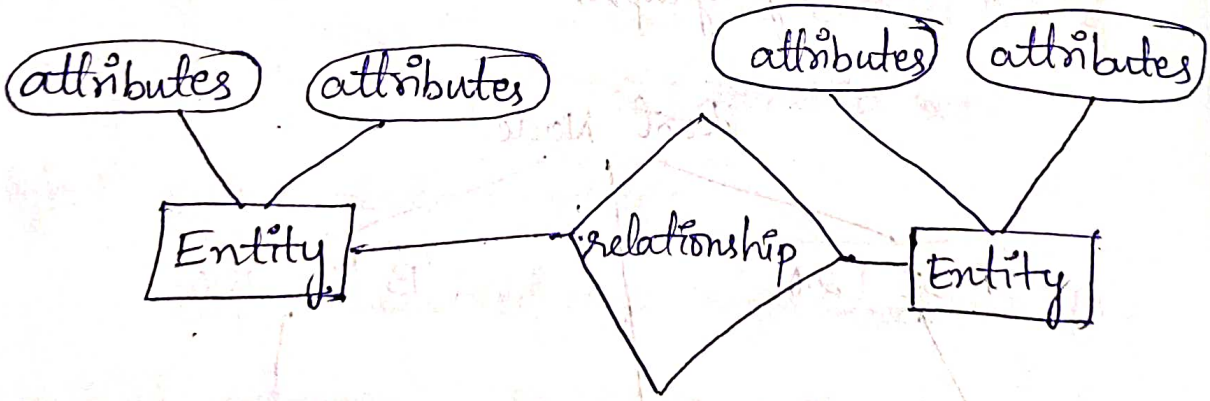
Conti

2) Entity-Relationship Model

→ ER Model is best used for the Conceptual design of a database.

→ ER Model is based on.

1) Entities & their attributes
2) Relationships among entities



→ Every attribute is defined by its set of values called domain.

For example:- In a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc

1) Relationship

The logical association among entities is called relationship. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

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

Database Language

A database provides a DDL to specify the database schema and a DML to express database queries and updates.

1) Data Definition Language (DDL)

We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL)
The DDL is also used to specify additional properties of the data.
SQL provides a rich DDL that allows one to define tables, integrity constraints, assertions, etc...

Example :-

```
Create table account (
    account number char(10),
    branch name char(10),
    balance integer )
```

* language is used to specify the database scheme, by a set of definition.
* DDL statement results in a set of tables. These tables are stored in a special files called data dictionary.
* Data directory is a file that contains "metadata". here metadata is nothing but a data about data.
* DDL contains the storage structure and access method used by the database system.

→ DDL Commands

Create :- used to create new tables

Syntax :- create table tablename
(Attr1 dt1, Attr2 dt2 ---- Attrn dtn);

Alter :- It is used to alter the table.
(To add few columns or changing the column size)

* Alter table tablename ADD (attribute name datatype);

* column size

Alter table tablename modify (attribute name Renamesize);

Drop :- It is used to delete the structure of data permanetly.

Drop table tablename

alter table tablename drop (attr);

Truncate :- It is used to delete all the records of a relation at a time, we can't delete single record from relation using where clause

Truncate table tablename;

Rename :- It is used to rename the table and also particular field name

```
Rename tablename1 to tablename2 ;
```

```
Alter table tablename rename column old
        to column new ;
```

2) Data-Manipulation language (DML)

A data-manipulation language (DML) is a language that enables users to access or manip-ulate data as organized by the appropriate data model.

The types of access are

→ Retrieval of information stored in the db
→ Insertion of new information into the db
→ Deletion of information from the db
→ Modification of information stored in the db

A query is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a query language

→ DML Commands

Enable the users to access or manipulate data that is stored in database

How to get it.

1) procedural DML :-

They required a user to specify what data is needed

2) Non-procedural DML :-

They requires a users to specify what data is needed without specifying how to get it.

Select :- It is used to retrieves the data from the db.

Select * from tablename displays the complete records

Select sid from tablename specific fields

Insert :- It is used to insert the values in to relations

insert into tablename (attributes)
Values (values);

(or)

insert into tablename values (&att1, &att2 ...);

Update :- It is used to modify or update the existing rows columns in the relation

Update tablename set attname = value
where condition;

Delete :- It is used to delete the data from database

Delete table tablename

alter table tablename delete (attname)

3) Data control language :- (DCL)

To control users access in a db related to the security issues

* It allows to restricts the user from accessing data in db
* Grant, Revoke

1) Grant :- It gives users access previleges to the db

Syntax :- Grant <previlege list>
on <relation name or view name>
To <user / role list>;

eg :- Grant Ali on employee To ABC;

2) Revoke :- It is used to cancel previous granted or denied premissions

Syntax :- Revoke <previlege list> on
<relation name or view name>
from <username>;

eg :- Revoke update on employee from ABC;

4) Transaction control language (TCL)

A set of tasks into a single execution unit if any of the tasks is fail the transaction fail transaction has only 2 results.

→ Commit - permant save
→ Roll back - previous transaction
→ Save points - temporary.

eg:- Grant All on Employee To ABC;

Revoke :- It is used to cancel previous granted or denied premissions

Database Users

Database users are categorized based up on their interaction with the data base.

These are 5 types of database users in DBMS

1) Database Administrator (DBA):

Database Administrator (DBA) is a person/team who defines the schema & also controls the 3 levels of database.

The DBA will then create a new account id & pass word for the user if he/she need to access the db

DBA is also responsible for providing security to the database & he allows only the authorized users to access / modify the database.

→ DBA also monitors the recovery & backup & provides technical support.

→ The DBA has a DBA account in the DBMS which called a system or superuser account

→ DBA repairs damage caused due to hardware & S/w failures.

2) Naive parametric End users :- Or Native users

parametric End users are the unsophisticated who don't have any DBMS knowledge but they frequently use the data base applications in their daily life to get the desired results.

→ Users Interact with sys through application programs

For example :-

Railway's ticket booking users are naive users. Clerks in any bank is a naive user because they don't have any DBMS knowledge but they still use the database & perform their given task.

3) Application program :-

Application program are the back end programmers who writes the code for the application programs. They are the computer professionals. These programs could be written in programming languages such as Visual Basic, Developer, C, Fortran, COBOL etc.

→ Users who write & develop applications by using diff tools

4) Sophisticated users :-

It can be engineers, scientists, business analyst, who are familiar with the database. They can

develop their own data base applications according to their requirement. They don't write the program code but they interact the database by writing SQL queries directly through the query processor.

5) Specialized user :

Specialized users :- These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.

Stand-alone users :- These users will have stand-alone database for their personal use.

Database Architecture.

A Database Architecture is a representation of DBMS design. It helps to design, develop, implement and maintain the db management system. A DBMS architecture allows dividing the db system into individual components that can be independently modified, changed, replaced & altered. It also helps to understand the components of a database.

A Database stores critical information & helps access data quickly & securely.

Therefore, selecting the correct Architecture of DBMS helps in easy & efficient data management.

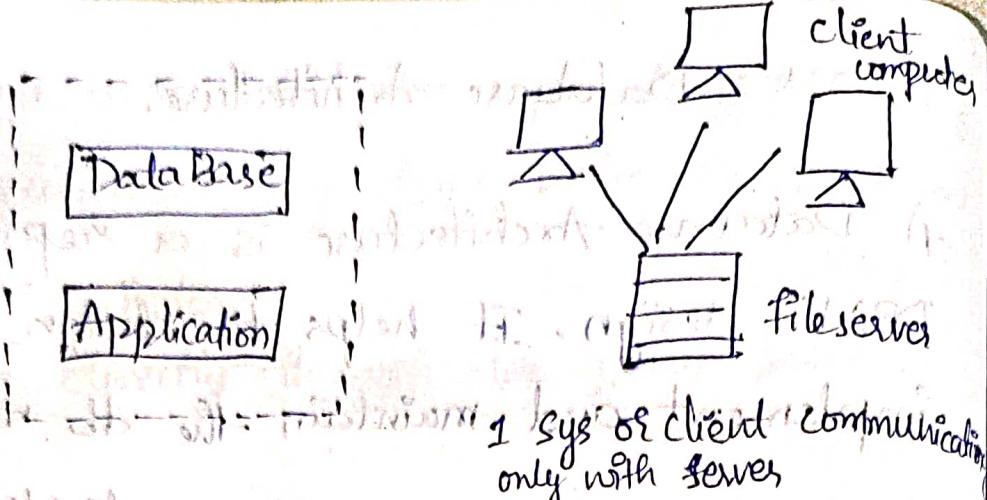
Types of Database Architecture:-

1) One Tier Architecture (Single)
2) Two Tier Architecture
3) Three Tier Architecture

(1) One Tier Architecture:-

Database & applications are resides in Single system that system known client system.

→ User Interface
→ presentation service
→ Application service.

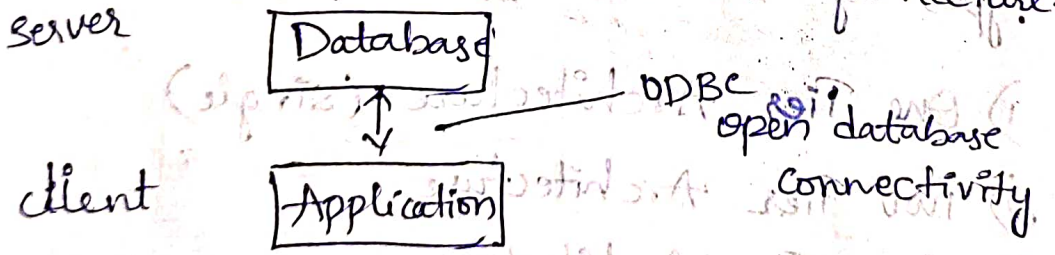


1 sys or client communication only with server

It is simplest architecture of the database in which the client, server & Database all reside on the same machine. A simple 1 tier architecture example :-

Would be anytime you install a database in your system & access it to practice SQL queries. But such architecture is rarely used in production.

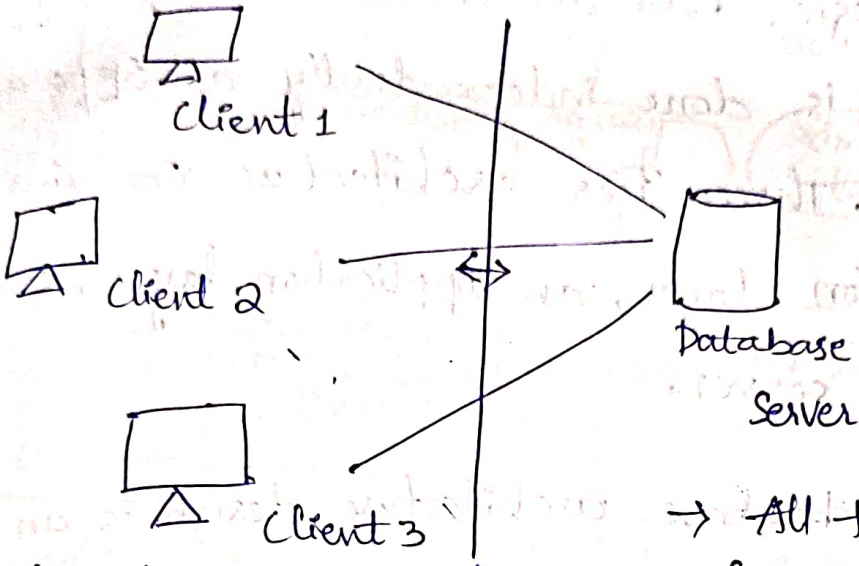
(2) Two Tier Architecture :- It is a client-server architecture.



Client Server Architecture.

A 2 Tier Architecture in DBMS is a Database architecture where the presentation layer runs on a client (pc, mobile, Tablet etc) & data is stored on a server called the second tier.

Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly It also provides direct & faster communication.

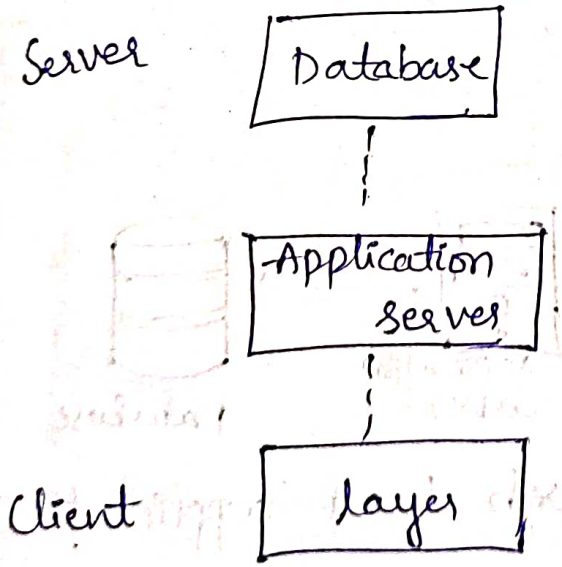


→ Direct Communication
→ Run faster.

→ All the client will Communicate with the db Server which is present in organization.

(3) Three tier Architecture

It is an extension of 2-tier archi tecture.

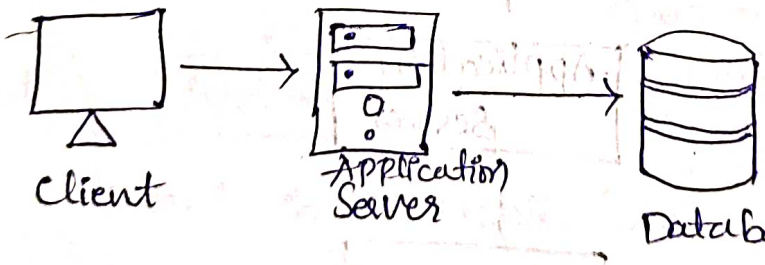


This Architecture is used to developed web application

A 3 Tier Architecture in DBMS is the most popular client-server architecture in DBMS in which the development & maintenance of functional process, logic, data access, data storage, & user interface is done independently as separate modules. Three-Tier architecture contains a presentation layer, an application layer, & a database server.

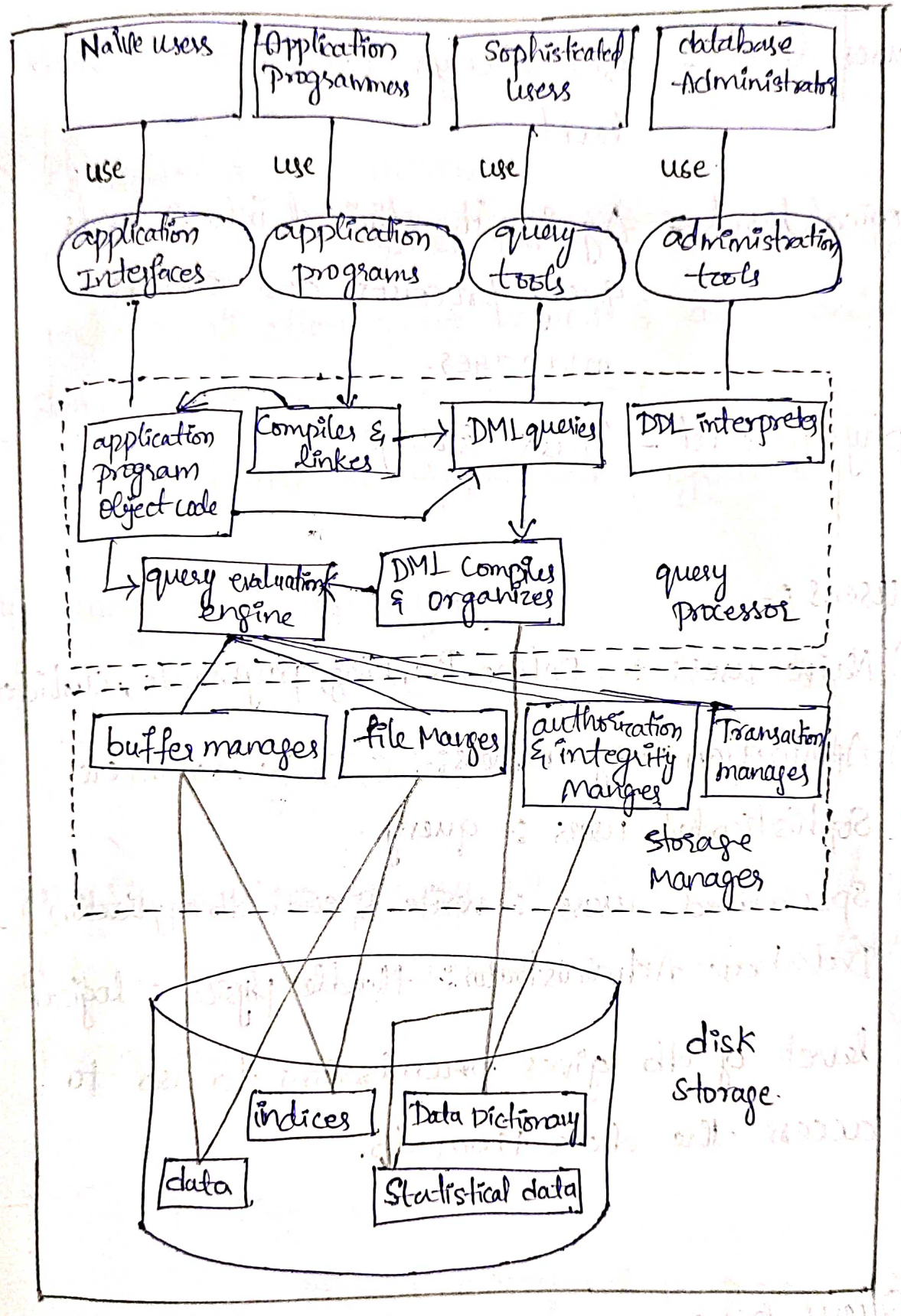
3-Tier database architecture design is an extension of the 2-tier client-server architecture. A 3-tier architecture has the following layers:

Presentation layer (your pc, Tablet, Mobile, etc)
Application layer (server)
Database server



→ It depends on web based application.

DBMS structure.



Overall DBMS Architecture

Divided into 3 parts

1) user level - user always present at the view level.

2) logical level - Again. it divided into 2 parts query processor and storage manager.

3) physical level - Disk storage

→ Users :-

1) Naive users :- Online Banking, paiments, custide

2) Application programmess :- C, Java, Cobol etc

3) Sophisticated users :- query

4) Specialized users : write special db application

5) Database Administrator :- Handle physical & logical level of db gives premissions to use to access the data from db.

→ Query processor :-

It Contains the following components

1) DML Compilers :- It processors the DML statements into low level language (Machine language)

2) DDL Interpreter :-

It processes the DDL statements into a set of table containing data.

3) Compiler and Linkers :-

It compiles the all DML statements and application programmers & also handle some procedure calls linker links some supporting files & library to application programmes.

4) Query Evaluation Engine :-

It executes the all instructions generated by DML compiles.

In single line query processor received all data from end users via different application programmers.

→ Storage Manager :-

Storage manager is a program that provides an interface between the data stored in database & the queries received from query Engine.

* It is also known as db control system.

1) Authorization Manager :- It ensures role based access control, it checks whether the particular user is to performed the request operation or not.

2) Integrity Manager :- It checks the integrity Constraints when db is modified.

   Eg age >30, phone no within 10 digits.

3) Transcation Manager : It controls concurrent access by performing the operations in a scheduled way.

4) File Manager : Manages the file space & data structure used to represent information in the DB.

5) Buffer Manager : It is responsible for cache memory & the transfer of data b/w the secondary storage & main execute memory.

→ Disk storage :-

1) Data files :- It stores the data.

2) Data Dictionary : It contains the information about the structure of any db object and also maintain the data

3) Indices → PK & FK : It provides faster retrival of data item.

Data base Design ← ER Modeling (Top down)
Normalization (Bottom up)

The database design process

The entity-relationship (ER) data model allows us to describe the data involved in a real-world enterprise in terms of objects & their relationships & is widely used to develop an initial database design

→ It is divided into six steps. The EE model is most relevant to the first three steps:

1) Requirement Analysis :-
The very first step in designing a database application is to understand what data is to

be stored in the db, what applications must built on top of it, & what operations are most frequent and subject to performance requirements. In other words, we must find out what the users want from the db.

2) Conceptual db design :- The information gathered in the requirements analysis step is used to develop a high-level description of the data to be stored in the db, along with the constraints that are known to hold over this data. This step is often carried out using the ER model or a similar high-level data model.

3) Logical Database Design :-

To implement our db design & convert the conceptual db design into a db schema in the data model of the chosen DBMS we will only consider relational DBMSs, & therefore the task in the logical design step is to convert an ER schema into a relational db schema.

The result is a conceptual schema, sometimes called the logical schema, in the relational data model.

4) Schema Refinement :-

The db design is to analyze the collection of relations in our relational db schema to identify potential problems.

5) Physical DB Design :

In this design we must consider typical expected workloads that our db must support & further refine the db design to ensure that it meets desired performance.

6) Security Design :-

In this design we identify different user groups & different roles played by various users. (eg : Development team for a product)

E-R Diagrams

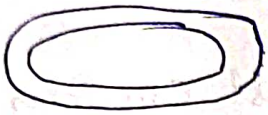
 - Entity

 - Relationship

 - Attribute.

 — weak entity

 - weak Entity relationship

 - Multivalued attribute.

1) Entity :-

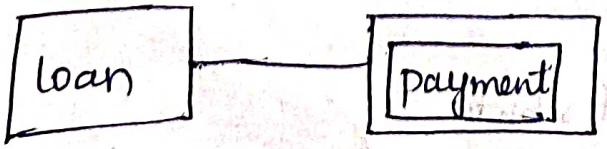
An Entity can be any Object, place, person or class. In E-R diagram, an entity is represented using rectangles.

Ex:-

weak Entity.

It depends on another entity, it doesn't have key attribute of their own. Double rectangle.

ex:-



2) Attributes :-

Entities are represented by means of their properties, called attributes. All attributes have the values. Ex :- Student entity may have name, class age as attributes.

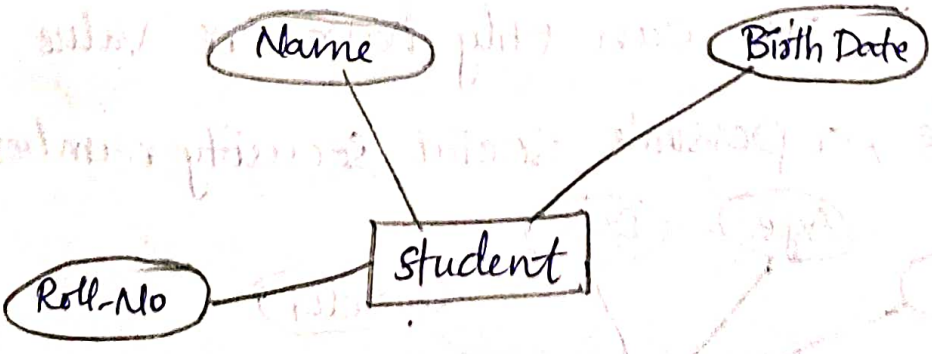
Types of attributes.

Attributes are properties of entites. It is represented by eclipses. Every eclipses represent one attribute & it is directly connected to its entity

i) Simple.

This are atomic values, which cannot be divided further..

ex :- Student's phone-number is an atomic value of 10 digits.

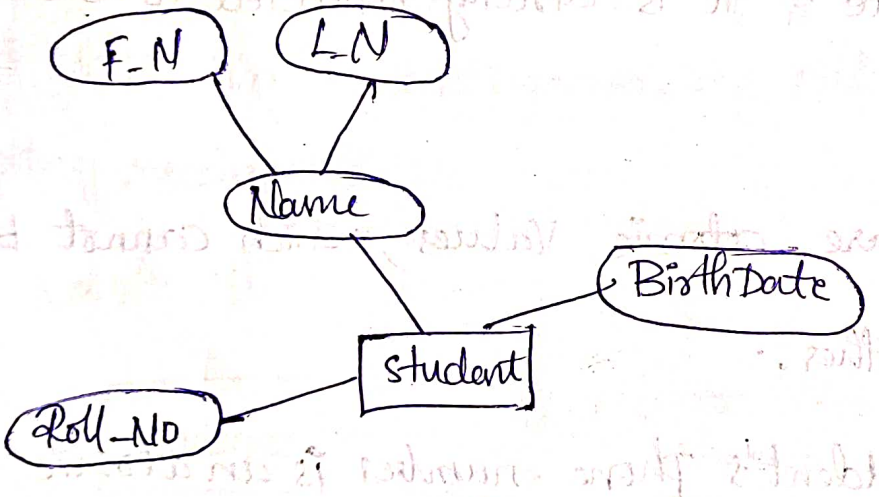


Simple Attribute.

ii) Composite attribute.

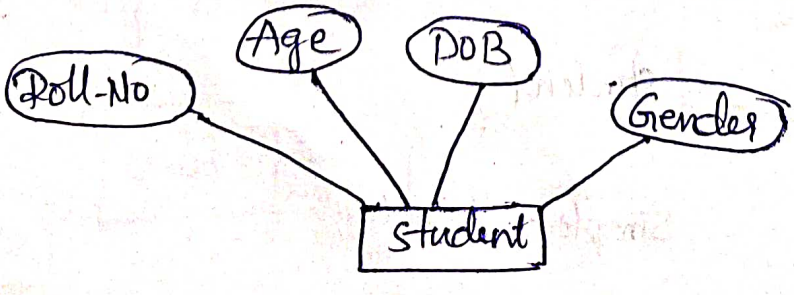
It is made of more than one simple attribute If the attributes are composite, they are further divided into tree like structure. Every node is connected to its attribute. That is composite are represented by eclipse that are connected with an eclipse.

ex:- Student's complete name may have first_name & last_name



iii) Single-Valued Attribute

It can only have one value For example, a person's social security number.

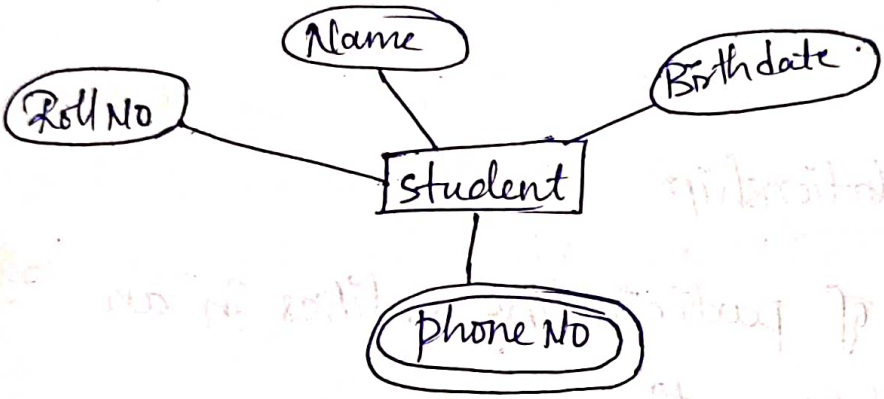


iv) Multi-Valued attributes

It Contains more than one values.

For ex :- a, person can have more than one phone num, email addresses etc.

It is represented by double eclipse

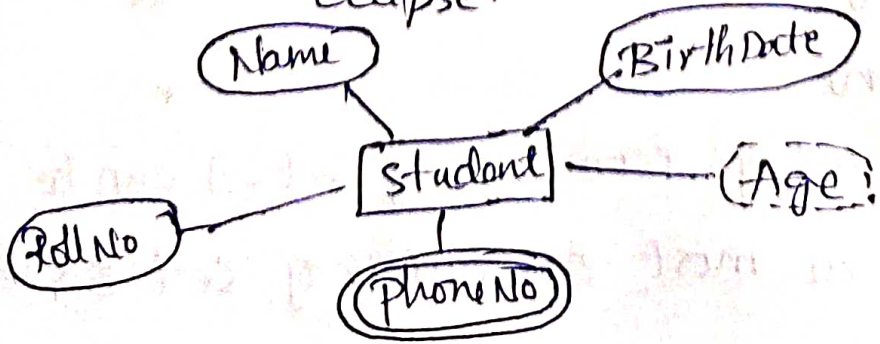


v) Derived attribute :-

It do not exist physical in the db, but there values are derived from other attributes presented in the db

ex :- age can be derived from dob.

Derived attributes are represented by dashed eclipse.



3) Relationships

The association among entities is called relationship.

ex:- Employee entity has relation called work.

Relationship set

It is Relationship of similar type is called relationship set.

Degree of relationship

The no of participating entities in an relationship defines the 'degree'

Binary - degree 2

Ternary - degree 3

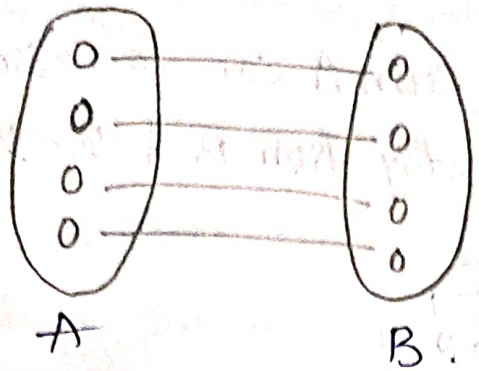
n-ary - degree.

Mapping Cardinalities :-

Cardinality defines the num of entities in one entity set which can be associated to the num of entities of other set via relationship set.

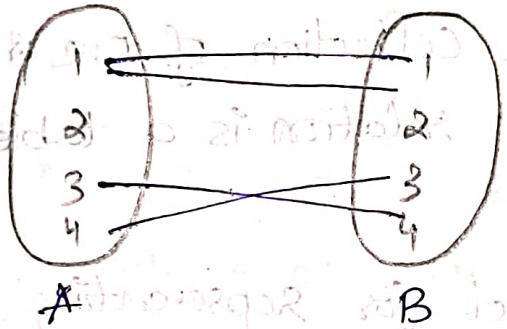
i) one - to - one.

One entity from entity set-A can be associated with at most one entity of set B & vice versa



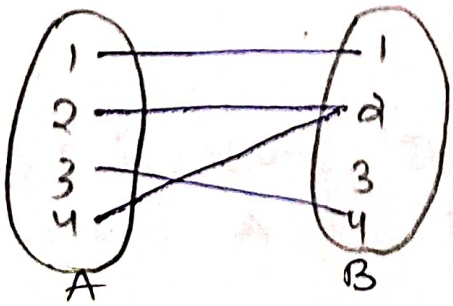
2) One to many.

One entity from set A can be mapped with more than one entities of entity set B. but from entity set B one entity can be associated with at most one entity.



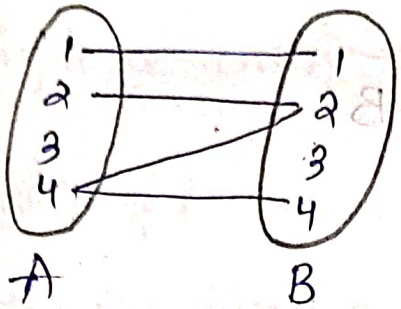
3) Many to one.

More than one entities from entity set A can associated with at most one entity of entity set B. but one entity from set B can be associated with more than one entity from entity set A.



4) Many to Many

One entity from A can be associated with more than one entity from B & vice versa.



Relational Model

The relational model is very simple.

→ A database is a collection of one or more relations, where each relation is a table with rows and columns.

→ The main construct for representing data in the relational model is a relation. A relation consists of a relation schema & relation instance

→ The relation instance is a table & the schema describes the column heads for the table.

structure of Relational db model

- collection of table
- Tables represent both data & relationships among the data present in the table
- Rows & column in table
- Tuple & attribute in relation.

→ Relation instance :- Specific set of rows

- Domain :- Set of permitted values. (condition)
  ex :- Bank. (Branch name)
- Null values (empty spaces or no values)

For ex :- student don't have phone num in that case we can't give '0'. In the table it should either unknown or Null value does not exsists

Ex :- (1)

Instructor

I_ID	Name	D_N	Sal
3458	John	Biology	65000
2121	Raj	physics	89000
2525	Abhi	Maths	87000
6786	Robin	Computer science	90000

(2)

Student → S_id, Name, Dept_name, Tot_cred

103 A Biology 56

(3)

Advisor

S_id	I_ID
103	24758
201	26589

→ Relation Schema & Instances.

1) Relational instance is represented by finite set of tuples; it do not have duplicate tuples

2) Relational schema contains the name of the relation & name of all columns or attributes

Properties of Relations

- Name of the relation is unique from all other relation (or table)
- Each relation cell contains exactly one atomic or single value.
- Each attributes contains different name.
- Each tuple is distinct but there are no duplicate tuples

a	1	2
a	1	2

X

→ Keys in RDBMS.

Why we need keys:-

For example :- Employee table (ID 101 to 109)

Here in the table we need to increment the salary of John but the table contains 3 similar name called John. , 101, 105, 108.

Here 105 John salary should be increment

So, here the real problem is to identify the unique tuples.

That the reason we need the keys.

1) Uniquely identify the tuple.

→ Super key.
→ Candidate key
→ Primary key
→ Alternate key
→ Unique key
→ Composite key
→ foreign key.

Employee table

ID	Name	SSN	Sal	Pho	Email

1) Super key.

- Like superkey. or superset.
- uniquely identify the tuple.
- Null values ex:- phone no

Ex:- ID is a super key, SSN, (ID, Name)

- Name is not a super key

A super is a collection of attributes or set of attributes so that we can uniquely identify the tuples.

2) Candidate key.

- super keys

{ID}, {SSN}, {ID, Name}

{ID, SSN}, {ID, phone no}

{Name, phone no}, {ID, Email}

{Name, SSN, phone no}

{Name, Email}

{ID, SSN, phone No} ----

- Minimal super keys are called candidate keys
- candidate keys:

{ID}, {SSN}, {Name, Phone no}, {Email}

- No repetation in the candidate key.

3) Primary key.

- To denote a candidate key
- Candidate keys

  {ID}, {SSN}, {Name, Phone}, {Email}.
- It should be unique & Not Null values.
- Primary key : {ID} 100 % unique & no duplication and not null.
- It is choosen by candidate key..
- Chosen with care by DBA.
- Never or very rarely changed.
- Candidate key with NULL values is not the primary key.

4) Alternate Key.

- The candidate key other than the primay key.
- All the keys which are not PK.
- CK → {ID}, {SSN}, {Name, phone}, {Email}
- PK → {ID}
- Alternate keys → {SSN}, {Name, phone}, {Email}

5) Unique Key

- Candidate Keys :- {ID}, {SSN}, {Name, phone}, {Email}
- Primary key - {ID}
- Alternate Key - {SSN}, {Name, phone}, {Email}
- Unique key - {Name, phone}
  - ↓ unique + not null
  - ↓ unique + Null values.

6) Composite Key.

- Composite Key :- {Name, phone}

There should be two attributes in this key. The two attributes forming the key. When the key has more than one attribute that key is called composite key.

{SSN, phone, Email}.

7) Foreign key.

All the key are deal with one table but in foreign key we deal with two tables

Ex :- Student table & Dept table.

Student

SID	Name	Dep-code	credits
101	John	101	12
102	Robin	102	14
103	Alya	103	20
104	Yusuf	104	10

Dept

Dep-Code	Dep-Name
101	CSE
102	EEE
103	ECE
104	Mech.

Referential integrity.
Constraints