

DBMSDatabase Management SystemsDatabase + Management System

Collection of data

is a set of programs to store & retrieve those data.

⇒ DBMS is a collection of data & set of programs to access & store those data in an easy and efficient manner.

⇒ DBMS is a software which is used to manage database.

Eg:- MySQL, Oracle etc are popular commercial DBMS used in different application.

Purpose of db System ⇒

Previously data is stored in files.

⇒ Drawbacks of using file Sys to store data

① Data redundancy & inconsistency →

Multiple file formats, duplication of info in diff 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.

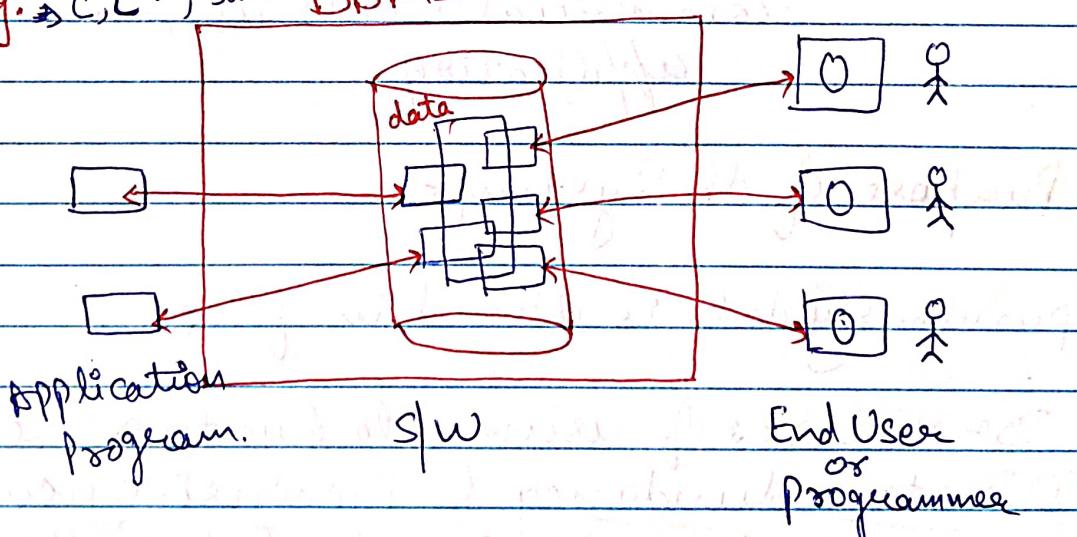
④ Data Security :-

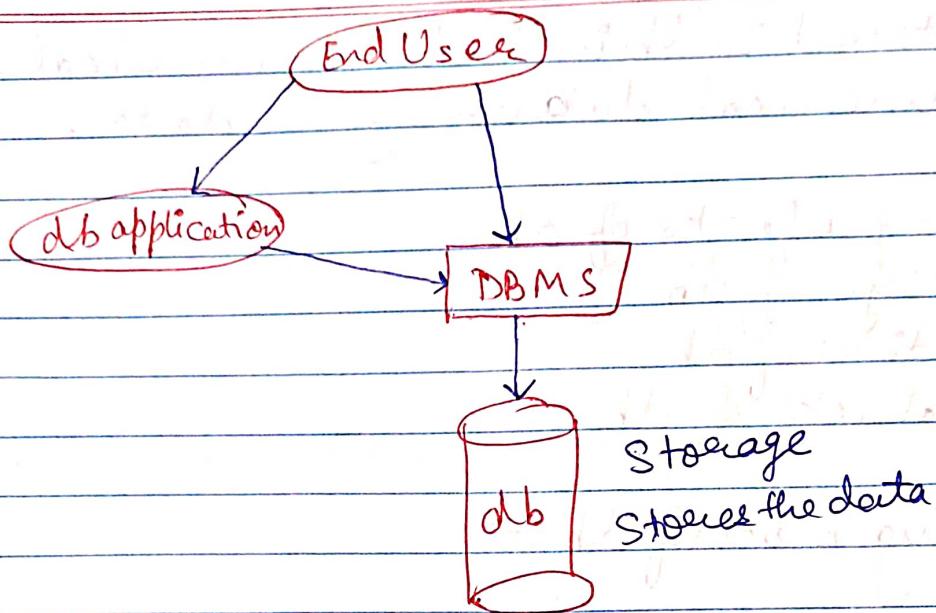
⑤ Transaction Problems :- Transferring of data will be problematic.

So, db System offer Solutions to all these problems.

DBMS → Oracle, MySQL, PLSQL

Application Prog. → C, C++, Java DBMS





Components of DBMS :-

↳ S/W

Users

Software

Data

Users:- Application programmers, End Users, db administrator.

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

Hardware:- H/w of a system can range from PC to a n/w of components.

⇒ It also includes various storage devices like hard disk & I/O devices like monitor, printer etc.

Data :- Data stored in db includes, numerical data, nonnumerical data or logical data.

- ⇒ Building blocks of db
- ⇒ Advantage of db
- ⇒ Applications of db
- ⇒ drawbacks of db

① Building blocks of db :-

- ⇒ Columns/ fields /attribute
- ⇒ rows/ tuples / record
- ⇒ Tables

② Advantages of db :-

- ⇒ Data Independence
- ⇒ Efficient data access
- ⇒ Data Integrity & Security
- ⇒ Data Administration
- ⇒ Concurrent access & crash recovery
- ⇒ Reduce application development time

③ Applications of database :-

- ⇒ Banking :- (All transactions)

- ⇒ Airlines :- Reservation & Scheduling flights

- * Universities :- Registration, grades
- * Sales :- Customers, products, purchases
- * Manufacturing :- Production, Inventory, Orders, Supply chain.
- * Human Resources :- Employee record, Salaries, tax deduction
- * Telecommunication :- Records of calls, monthly bills

(4) Drawbacks of dbms :-

- The Overhead cost of using dbms
- High initial investment in h/w, s/w & training
- Cost of defining & processing data.
- Overhead for security, concurrency control, recovery.

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

- Simple, well defined db app that are not expected to change.
- No multiple user access to data.

DBMS Architecture

3 levels

External Level / View Level

Conceptual Level / Logical Level
Internal Level / Storage / Physical Level

① External Level :- This level describes that part of database that is relevant to each user.

② Conceptual Level :- This level describes what data is stored in db & relationship among the data.

- (a) All Entities, attributes & their relationship
- (b) Constraints on the data
- (c) Security & Integrity information

③ Internal Level :- It is physical representation of database. This level describes how the data is stored in db. It covers the data structure & file organization.

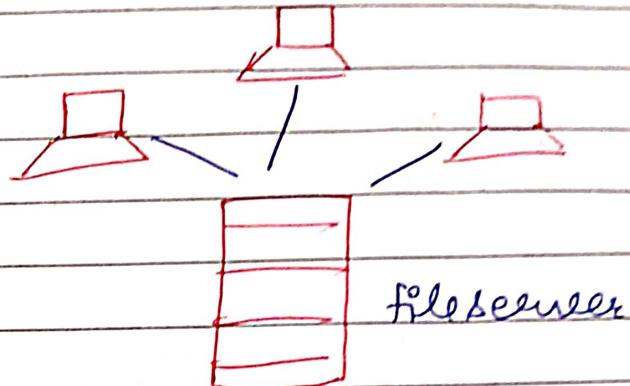
⇒ DBMS can be seen as either Single tier or multtier.

⇒ An n tier architecture divides the whole sys. into related ~~but~~ but independent n modules.

① 1-tier architecture

1-tier ~~HTMl~~
1 Server

- User Interface
- Presentation Service
- Application Service



Communication b/w Client & Server.

Preferences ⇒ When we want to save constant data.

When there are no multiple user to access data.

② 2-tier architecture

~~Layer~~

Client
tier

Database
tier

Client computers

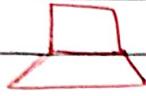
db
server

- Limited clients
 - It is a Client Server architecture.
 - Direct Communication
 - Run faster
 - Easy maintenance
- [Offline Ticket reservation]
 [Crediting or debiting money from bank through offline mode.]

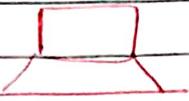
③ 3-tier architecture

Web based application

Client tier



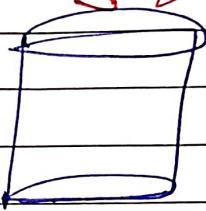
Business logic tier



Application Server



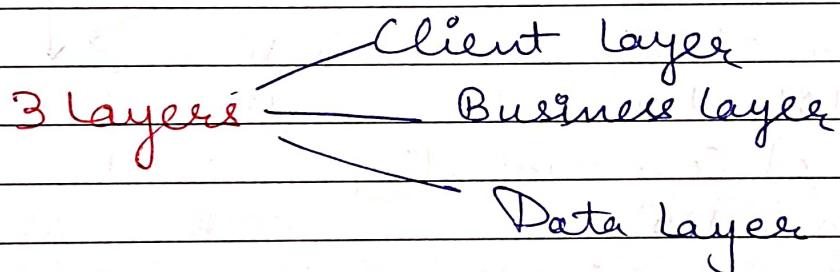
db tier



db Server

⇒ It separates tiers from each other based on complexity of the users & how they use the data present in db.

⇒ It is a web based application.



⇒ Database (Data) tier :-

db resides along with its query processing lang.

⇒ Application (Middle) tier :-

- for a user, this presents an abstract view of db.
- It acts as mediator b/w user & db.

⇒ User (presentation) tier :-
Client

- End user operate on this tier & they know nothing about any existence of db beyond this layer.

Data Independence



Capacity of changing the Schema at one level without affecting the other.

एक लेवल पर जो change हुआ वह दूसरे लेवल पर Effect
नहीं करता।

⇒ db System are designed in multilayer.

⇒ DB system stores data about data, known as meta data.



follows a layered architecture

Two types of Data Independence

- logical data independence (Conceptual Schema)
- physical data independence (Internal Schema)

⇒ It stores information about how data is managed inside.

⇒ Ability to change logical Schema without changing external or application program.

⇒ Ability to change the physical data without impacting the logical data / External.

Change in Internal Schema, diff. file organization, storage device, structure should be possible without affecting the conceptual / External Schema.

Application Prog \Rightarrow access by the user.

Logical Schema - relationships

Physical Schema - Storage

End Users

External
Schema 1

External
Schema 2

External
Schema 3

logical data
independence

Logical Schema

physical data
independence

Internal Schema

Database

Attributes \Rightarrow Are the properties which define Entity type.

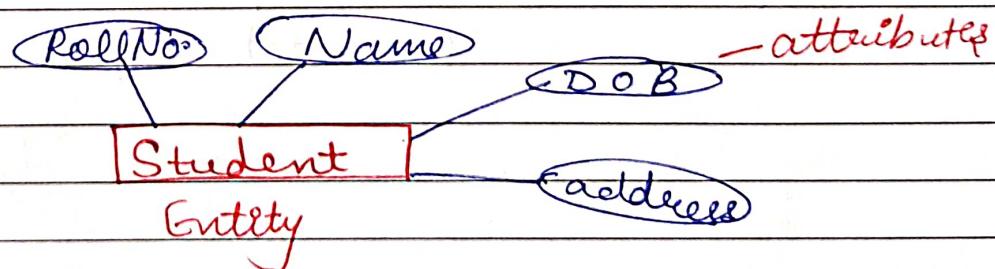
Entities are represented by means of their properties is called attributes

Entity \Rightarrow An object with physical existence.
or Eg:- Person, Car, house
for obj with Conceptual Existence.
Eg. Company, Job, University

Entity type :- An Entity is an object of Entity type.

Entity Sets :- Set of all Entities.

Eg :- Roll No, Name, DOB, Age, Address are the attributes which define Entity type Student. (C)

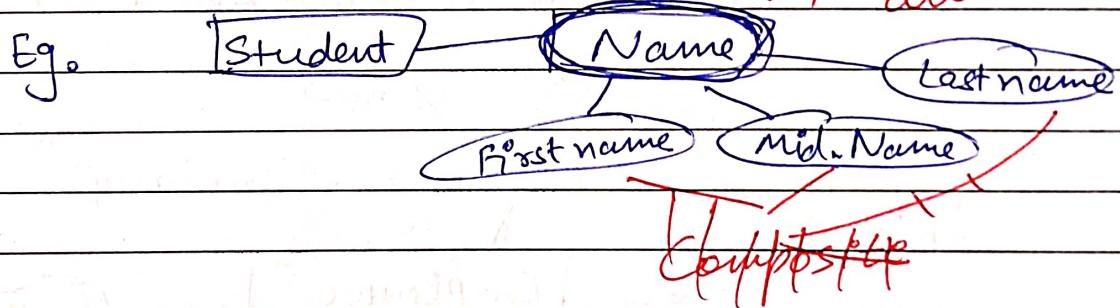


Types of attributes

- ① Simple attribute
 - ② Composite attribute
 - ③ Derived attribute
 - ④ Single Valued attribute
 - ⑤ Multivalued attribute
- ⑥ Simple attribute \Rightarrow These are atomic values, which cannot be divided further.
 Eg:- Student / phone no. \rightarrow atomic value

TStudent \rightarrow (Age) atomic value

- ⑦ Composite attribute \Rightarrow Which can be divided into smaller Subparts.



- ⑧ Derived attribute \Rightarrow These attribute do not exist in the physical database, but their values are derived from other attributes present in the db.

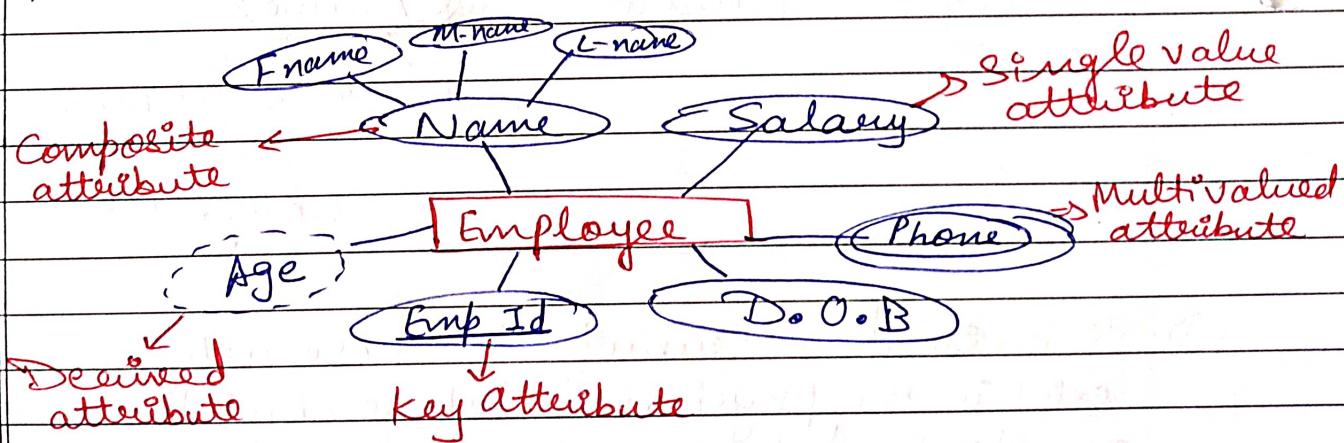
Eg. T-Student → (Age) → Derived attribute
 (DOB) → Stored attribute

④ Single Value attribute ⇒ contains only single value.

Eg. T-Student → Age
 Security-no.

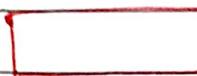
⑤ Multi Value attribute ⇒ May contain more than one value.

Eg. Person can have more than one phone no.
 person can have more than one email address.

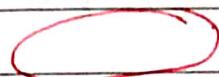


ER Diagram Components

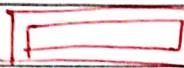
It is a pictorial representation of data that describes how data is communicated and related each other.



Entity set



Attribute



Weak entity set



Multivalued attribute



Relationship set



Derived attribute



Identifying relationship set for weak Entity set



Total participation of Entity set in relationship



Primary key



Discriminating attribute of weak Entity set

ER Model to Relational Model.

Some of the methods are automated & some of them are manual.

ER diagrams mainly comprise of

- ① Entity & its attributes
- ② Relationship which is association among entities.

Keys in DBMS

It is an attribute or set of attribute which help you to identify a row (tuple) in a relation (table).

→ It also allows you to find the relation b/w two tables.

Eg:-

Std	Person
id → key	Name
name	D.O.B
address	Pass No. License No.

are keys

Types of Keys

- 1 Primary key
- 2 Candidate key
- 3 Superkey
- 4 Foreign Key

① Superkey ⇒ A Superkey is a set of attributes that can identify each row uniquely.

⇒ Superkey consists of any no. of attributes.

Eg.: Std (roll, name, Sex, age, add, class, section)

- ⇒ (roll, name, Sex, age, add, class)
- ⇒ (class, section, roll)
- ⇒ (class, section, roll, sex)
- ⇒ (name, add)
 - ⇒ Superkey is a superset of candidate key -

② Candidate key ⇒

- ⇒ A Superkey without redundancy.
- ⇒ It is not Reducible further.
- ⇒ Minimum set of attributes used to uniquely differentiate record of the table.
- ⇒ A Candidate key can never be null or Empty. And its value should be unique.
- ⇒ There can be more than one candidate key for a table.
- ⇒ A Candidate key can be combination of more than one attribute.

③ Primary key ⇒ It is Candidate key that is most appropriate to become the main key for any table.

- ⇒ It is a key that can uniquely identify each record in a table.

Student Primary key

Std id	name	phone	age	Primary key \Rightarrow Std id
1	Akhil	98123456	16	
2	Bhanu	92103421	17	
3	Akhil	99100012	18	
4	Bhanu	93242654	17	

- \rightarrow Unique values must never be null.
- \rightarrow Uniquely identify each record in table.

Note: Primary key are mandatory for Every table
Each record must have a value for its primary key.

- \rightarrow When choosing a primary key from the pool of Candidate key always choose a single simple key over a composite key.

(i) Foreign key \rightarrow Is an attribute which establish relationship b/w 2 tables.

- \rightarrow Foreignkey is a field (collection of fields) in one table that refers to the Primary key in another table.

Eg Persons table

Primary key

PId	lastname	Fname	Age
1	Kavi	G	30
2	Raghu	P	32
3	Priya	A	34

Orders table

Foreign key

Or Id	Or No.	PersonId
1	3246	3
2	1234	3
3	3543	2

Parent table
(or)
Reference
table

Child
table

It can
have
duplicate or
null data.

- ⇒ Records cannot be inserted into a detail table if corresponding record in the master table does not exist.
- ⇒ Record of a master table cannot be deleted if corresponding record in the detail table actually exist.
- ⇒ Parent must be unique or Pk
- ⇒ Child may have duplicate or null value unless it is specified.
- ⇒ Constraint ~~specified~~ specified on child not in Parent.

CLASSTIME	Page No.
Date	/ /

- ⇒ Parent record can delete only if no child record exist.
- ⇒ Parent cannot modify if child record exist.

Composite Key

A key which has multiple attributes to uniquely identify rows in a table is called composite key.

- ⇒ But the attributes which together form the Composite key are not a key independently or individually.