Unit-I

Concepts & Overviews

* **Data**: Collection of related data.
* **Database**: It refers to collection of logically related data items
* **DBMS**

→ DBMS is a software which is used to manage database is known as DBMS.
→ The DBMS is a software system that facilitates the process of defining, constructing, manipulating & sharing databases among various users & applications.
→ The database defination is also stored by the DBMS in the form of database catalog or dictionary, it is called meta-data

Meta-Data

→ A meta-data is the data about the data. The meta.data describe objects in the database & makes easier for those objects to be accessed or manipulated.

types:

Descriptive meta data: it describes a resource for purpose such as discovery & identification.

Structural meta data: It describes how compound objects are put together. Ex: how pages are ordered to form Chapters.

Administrative meta data: it provides info. to help manage a resource such as when & how it was created.

* Various views of Data

External view ⇒ physical schema

↑

Conceptual schema

↑

Logical schema

↑

Physical schema

↑

Database ⇒ logical schema

↳ it Knows how data physically stored in database.

↳ it means storage details maintain by physical Scema.

↳ ex: | Name | DOB | Add | yp. schema.

Ex: Student table.

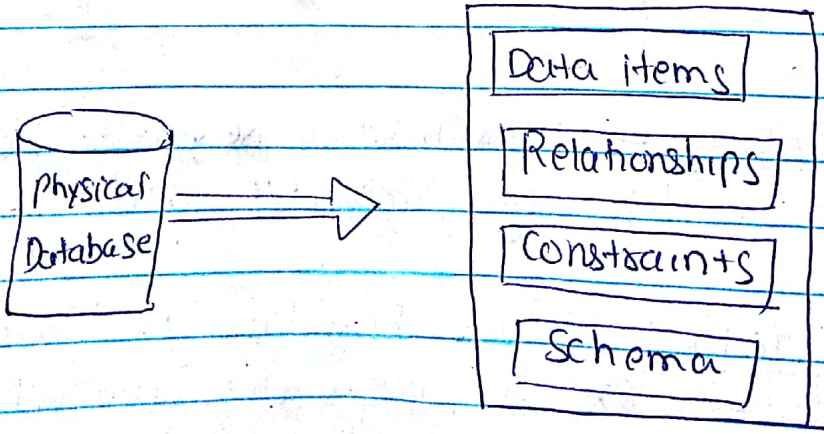
↳ it only tells where data store

↳ info btayega Kya exist Karta hai' data mein.

⇒ Conceptual schema : it hide yours physical schema

* Components of a Database

A database consists of four components as :



⇒ Data items : it is defined as distinct piece of information.

⇒ **Relationships**: It represents a correspondence b/w various data elements

⇒ **Constraints**: These are the predicates the define correct database states.

⇒ **Schema**: it describes the organization of data & relationships of database.

* **Database Models**:

* Database languages

The database provides two types of languages :

1) **DDL [Data Definition language]**

↳ Database schema is specified by a set of definations expressed by special language.

2) **DML [Data Manipulation language]**

↳ it facilitates statements to enter, update, delete & perform complex queries on tables.

two types of DML :

(A) **Procedural DML**

↳ it require a user to specify what data are needed & how to get those data.
↳ Ex : C language

(B) **Non-Procedural DML**

↳ it require a user to specify what data are needed without specifying how to get those data.
↳ Ex : C++ language.

* Responsibilities of Database Administrator (DBA)

1) **Database implementation & design :**

↳ the critical duty of DBA is designing database for maximum performance, scalability, flexibility & reliability.

↳ the DBA is responsible for installing new DBMs & upgrading existing DBMS.

2) Performance monitoring & timing

↳ A DBA must ensure database are fast & responsive.
↳ It is essential to monitor the state of database & tune accordingly.

3) Availability backup & recovery

↳ the most important job of the DBA is that of availability of backup & recovery of data
↳ It is bec. of the value placed on electronic data, the database must be protected from all forms of failures i.e. hardware, software & human.

4) User & Data security

↳ A DBA is responsible for assigning users to database & determining the proper security level for each other

5) working with developers & network administrator

↳ DBA need to work with closely with other system staff to ensure hike database performance.
↳ DBA also work with network administrator's who run's the server on which their database recite.

6) Deciding & loading database contents :

↳ the DBA in consultation with senior management is normally responsible for defining the conceptual schema of database.

7) **Deciding data structures**

↳ Once the database contents have been decided, the DBA could be normally make decisions regarding how data is to be stored & what data is need to be maintained.
↘ Indexes

8) **Monitor actual usage**

↳ The DBA monitors the actual usage to ensure that the ~~proce~~ policies laid down regarding use of the database are being followed.

* **Three schema architecture of DBMS:**

↳ The goal of this is to separate the user applications from the physical database divided into 3 categories:

- External schema } individual user view
- Conceptual schema } community user view
- physical / Internal schema } storage view or physical view

database ko combined kaisa hai {
location se deal karega {

① **Internal level / Physical schema**

→ It describes the actual physical storage of data & the way in which the data is actually stored in memory.
→ This level is not relational bec. the data is stored according to various coding schemes instead of tabular form

→ This is the low representation of data.
→ It gives complete details of data storage path, various record types, physical sequence of records etc.

② Conceptual level

→ The conceptual level is also known as logical level which describes the overall logical structure of whole data.
→ This level is relational bec. data visible at this level will be relational tables & operators will be relational operators.
→ This level represents entire contents of the database in an abstract form in comparison with physical level.
→ Here, conceptual schema is defined which hides the actual physical storage & concentrate on relational model of database.

③ External level

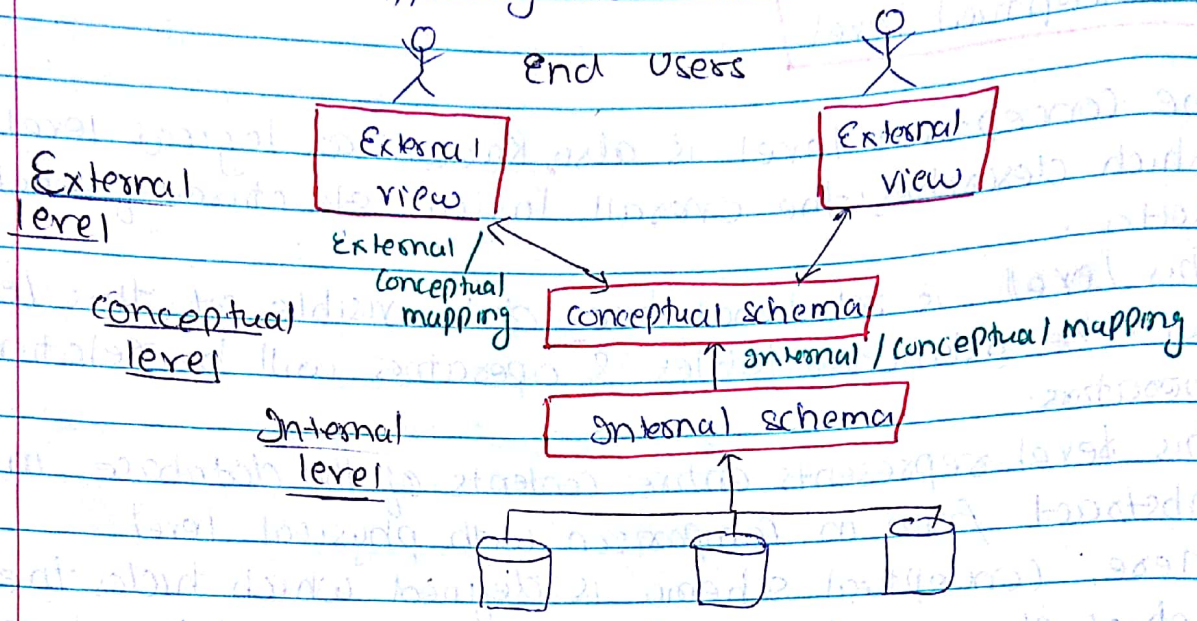
→ The external level is concerned with individual users.
→ This level describes the actual view of data seen by individual users.
→ The external schema is defined by the DBA for every user.
→ The remaining part of database is hidden from that user.
→ The user can access only that part of database for which he is authorized by DBA.
→ This level is also relational or very close to it.

↳ Advt

1. The changes to physical storage organization does not affect the internal structure of the database.

→ to use the database, the user is no need to concern about the physical data storage details.

→ the database storage details can be changed by the DBA without affecting the user's view.



[the three-schema architecture]

↳ External / conceptual mapping : It defines the correspondence b/w the conceptual view & the physical view

↳ Conceptual / internal mapping : It defines the correspondence b/w the conceptual view & the physical view operations

Client Server Architecture of a Database System

↳ In client-server technology, there is a software which acts as a whole DBMS or personal computers which are connected with server through a network interface.

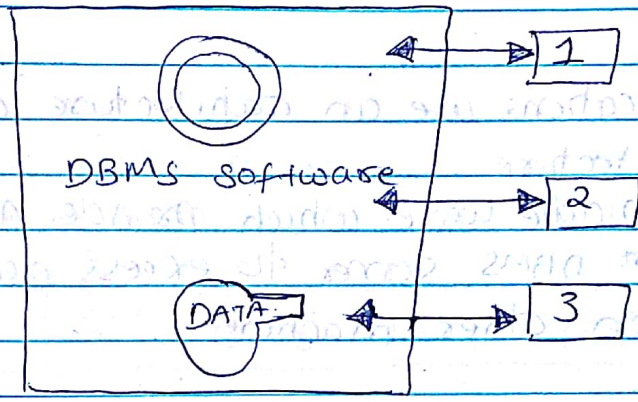
↳ Components of Client-Server architecture :

@ Server : Server is DBMS itself.

(b) Clients : Client machine is a personal computer or workstation which provide services to both server & users.

(c) Network interface : Clients are connected to server by network interface.

1) **Single Tier** : In single tier system, all the software are present on a single machine which are excessed by clients / terminates. → server system



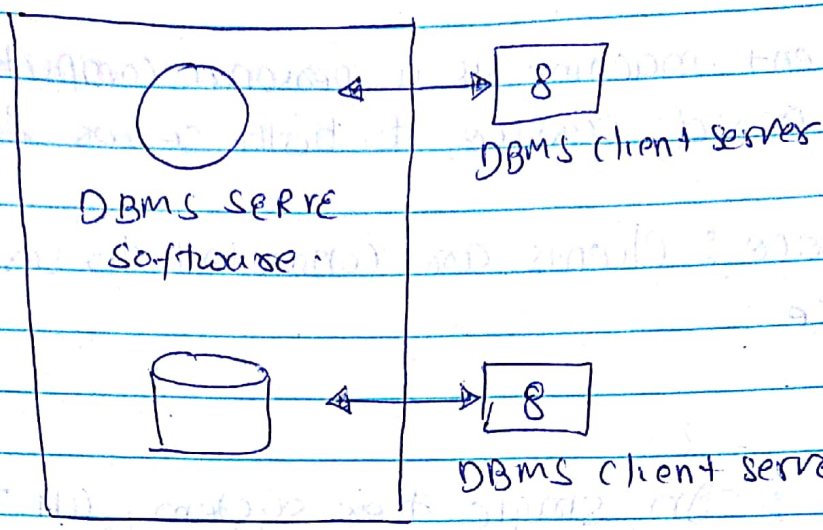
Server.

2) **Two-Tier Client** :

→ the application is divided into the component that besides at the client machine which invokes database system functionally at the server machine through query language statement.

→ the architecture described here are called two-tier architectures bec. the softwares components are distributed over two systems : Client & Server

→ the adv. of this architecture are its simplicity & seamless compatibility with existing systems.

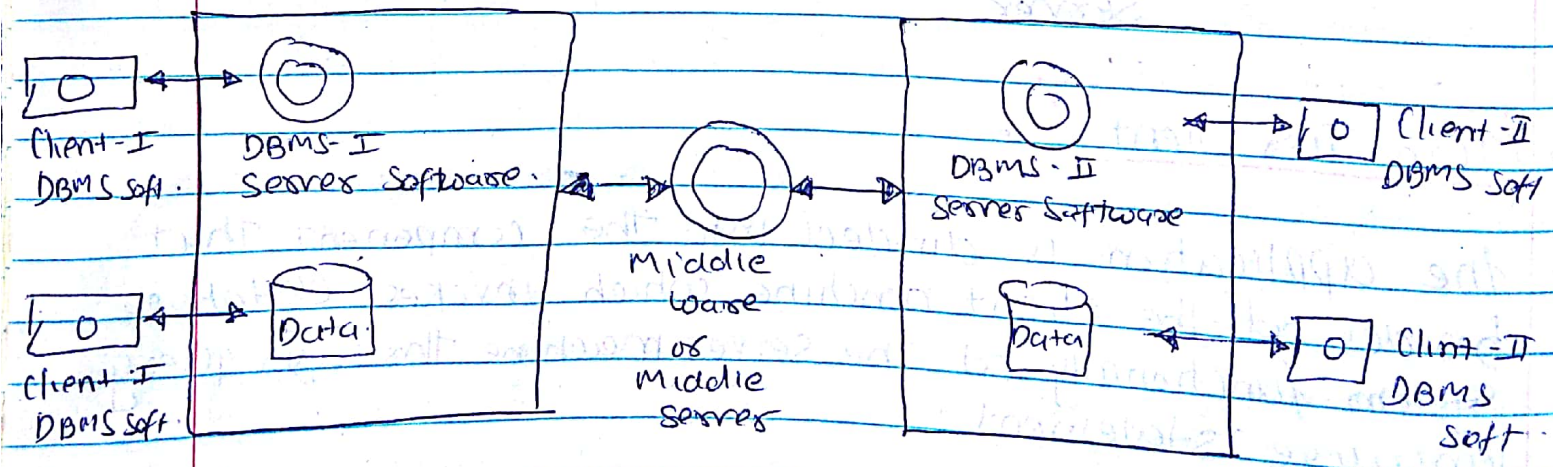


3) Three - tier Client

for web applications.

→ Many web applications use an architecture called the three - tier architecture.

→ This system at middle ware which provides away for the client of 1st - DBMS server to excess data from another DBMS on other machine.



→ This intermediate of middle ware is called the application server or the web server depending upon the application.

→ This server plays an intermediary role by running application programs & storing bussiness rules (procedures & constraints)

→ It can also improve database security by checking a client's credentials before forwarding a request to the database server.
→ It is used in very large applications like banking, airlines, hospitals etc.

>> **Adv. of Client-Server Architecture**

1. It inc. the overall performance of DBMS.
2. Load can be distributed among clients.
3. It reduces cost

>> **Disadv.**

1. Network is error prone
2. Recovery is typical

* **Adv. of DBMS over file processing System** are:

1) **Controlled redundancy**:

→ In a file Proc. system, each application program has its own data which causes duplication of common data items in more than one file. & that lead to wastes of a lot of storage space.
→ But in DBMS, this duplication can be carefully controlled & it assumes the responsibility of for propagating updates

2) **Data Consistency**:

→ In file pro. system, the problem of updating multiple files leads to inaccurate data as diff. files may contain diff. info of the same data item at a given point of time.

→ In DBMS, this problem of inconsistent data is automatically solved by controlling the redundancy.

3) Program Data Independence

→ In file proc. system, they are generally data dependent, which implies that the data organization & access strategies are dictated by the needs of the specific applications & application programs developed accordingly.

→ But in DBMS, it provide an independence b/w the file system & application program, that allows for changes at one level of data without affecting others.

~~4) Improved data & integrity~~

~~Data integrity means~~

4) Sharing of data

→ In file proc. system, there is limited data sharing possibilities as each application has its own private files & users have little choice to share data outside their own applications.

→ In DBMS, the data is centrally controlled & can be shared by all authorized users.

5) Security

→ In file proc. system, it is very difficult to enforce security checks & access rights, since applications pgm are added in an adhoc manner.

→ But DBMS, improved security as it protect the data

contained in the database from unauthorized users.

6) Data access is efficient.

⇒ The database system utilizes diff. sophisticated techniques to access the stored data very efficiently.

7) Improved backup & recovery facility

⇒ Through its backup & recovery subsystem, the database system provides the facilities for recovering from hardware or software failures as comp. to file proc. system.

8) Data quality is high.

→ In DBMS, the quality of data are very high as comp to file proc. system due to presence of tools & processes in DBMS.

9) Minimal program maintenance

→ In file proc. system, high maintenance effort are req'd.

→ But in DBMS, maintenance is reduced due to independance of data & application programs

10) Economical to sale.

⇒ In DBMS, the operational data of an organization is stored in a central Database

⇒ The application pgm's that work on this data can be built with very less cost as comp. to file proc. system

* Disadvantages of DBMS:

1) Complexity increases.

→ The data structure may become more complex bec. of the centralised database supporting many applications in an org.

→ This may lead to difficulties in its management.

2) Requirement of more disk space.

→ The wide functionality & more complexity inc. the size of DBMS.

3) Additional cost of hardware

→ The cost of DBMS installation is much more.

→ It depends on environment & functionality, size of & maintenance cost of hardware.

4) Cost of conversion:

The cost of conversion from old file system to new database system is very high.

5) Need for backup & recovery.

For a database system to be accurate & available all times, a procedure is req. to be developed & used for providing back up copies to all users when damage occurs

* Data Independence

↳ It is defined as the characteristics of a database system to change the schema at one level without having to change the schema at the next higher level.

↳ The application programs do not depend on any particular physical representation or access technique of the database.

↳ The DBMS achieved the data independence by the use of three-level architecture.

data independence are of two types:

1) **Physical data independence**: ◎ it indicates that the physical storage structures or devices used for storing the data could be changed without changing the conceptual view or any of the external views.

◎ In phy. data independence, the conceptual schema insulates the users from changes in the physical storage of data.

2) **Logical data independence**: → It implies that application pgms need not to be changed if fields are added to an existing records nor they have to be changed if fields not used by application pgms & deleted.

◎ It indicates that the conceptual schema can be changed without changing the existing external schemas.

◎ It also insulates applications programs from operations like combining of two records into one, etc.

◎ It is more difficult to achieve than Physical D.I. Since application pgms are heavily dependent on the logical structure of the data they access.

→ Thus file may migrate from one type of physical media to another or the file structure may be change without any need for change in application pgms.

Unit-I

Entity-Relationship Model:

Person or real world object

↳ was introduced by "Chen in 1976"

↳ E-R model is a high level, conceptual data model.

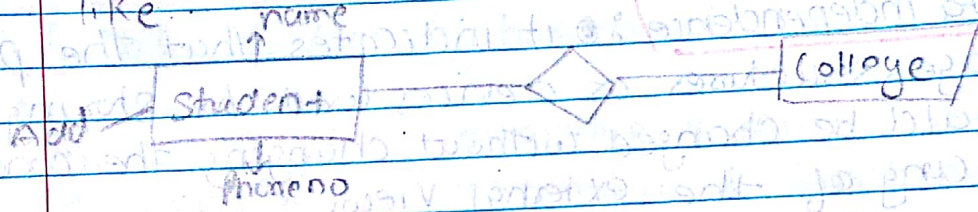
↳ The E-R model is mainly used for communication blw database designers & end users during the analysis phase of database development

Entity represented by

or

relation rep by

like



↳ It consists of various concepts:

1) Enterprise:

* It refers to any kind of organization.
* Ex - colleges, schools, banks etc.

2) Entity:- Entity is an object in the system that we want to model. store info about it.
ex: bb. emp, teach, student — agency

* Entity refers to an "object" or "thing" in real world with an independent existence.
* Object may be any person, place event etc. through which data can be stored in the database.
* Ex - Students of colleges & schools etc, employ, person, company etc.

↳ ER model is a representation of the structure & constraints of a database i.e. independent of DBMS.

2) **Entity types**: The collection of entities that have the same attribute.

3) **Attributes**:

* It gives the characteristics of the entity or attributes are the properties that describe an entity.
* Ex: a student can described by his name, age, address, height, class etc.

4) **data value**: data value is the info. or data which is stored in attributes of any entity.

5) **Entity sets**: All the entities having same attributes make an empty set.

6) **Domain**: Domain or value set is the set of all permitted values or info about any attribute.

* **Types of attribute** three major types:

1) **Single valued & Multi-valued attributes**

→ Single valued attribute: An attribute which have a single value for a particular entity is called single valued attribute.
Ex - age of a student.

→ Multi-valued Attribute: An attribute which have a set of values for a particular entity is called multi-valued attribute.
ex- colour of a car.

~~2) Single valued~~

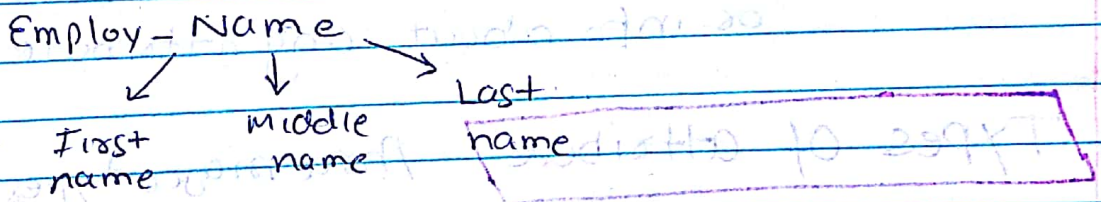
2) Simple & Composite attributes

→ **Simple attribute**: The attribute which can not be divided into smaller sub-parts are simple attribute.

Ex: age of student.

→ **Composite attribute**: The attribute which can be divided into smaller sub parts which represent more basic attributes with independent meanings called composite attribute.

Ex:



3) Stored & derived attributes

→ **Stored attributes**: An attribute which cannot be derived from an other known attribute is known as stored attribute.

Ex: D-o-B of employ.

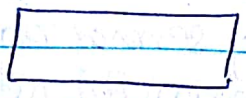
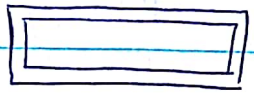
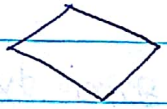
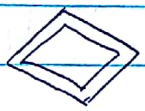
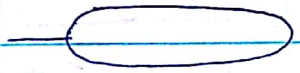
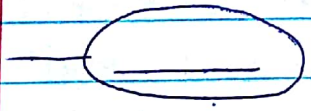
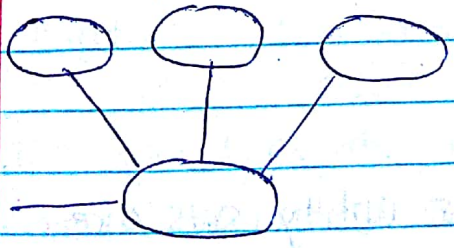
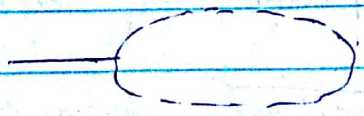
→ **Derived attributes**: An attribute which can be derived from an other known attributes is known as derived attribute.

Ex: age of an employ.

* Entity - Relationship Diagram / Notations.

↳ E-R dig. represents the logical structure of a database.

↳ Symbols used in E-R dig. are shown:

	Symbol	meaning
1.		Entity Set (Strong)
2.		Entity Set (weak)
3.		Relationship Set.
4.		Identifying relationship type.
5.		attribute.
6.		Key attribute.
7.		Multi-valued attributes
8.		Composite attributes.
9.		Desired attribute.

* Relationship sets

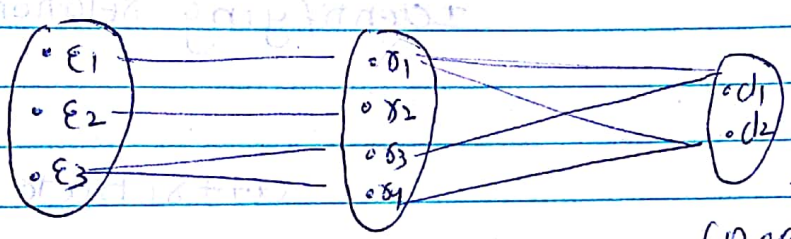
↳ ~~An relationship~~ set is an ~~association~~ set of ~~relationships of same type~~.

↳ Relationship

* A relationship is the association among several entities.
* It connects diff. entities through a meaningful relation.

↳ Relationship set

* A relationship set is set of relationships of same type.
* ~~Ex: Consider an example, emp~~



(Employee) (works for) (Department)

↳ Degree of Relationship sets:

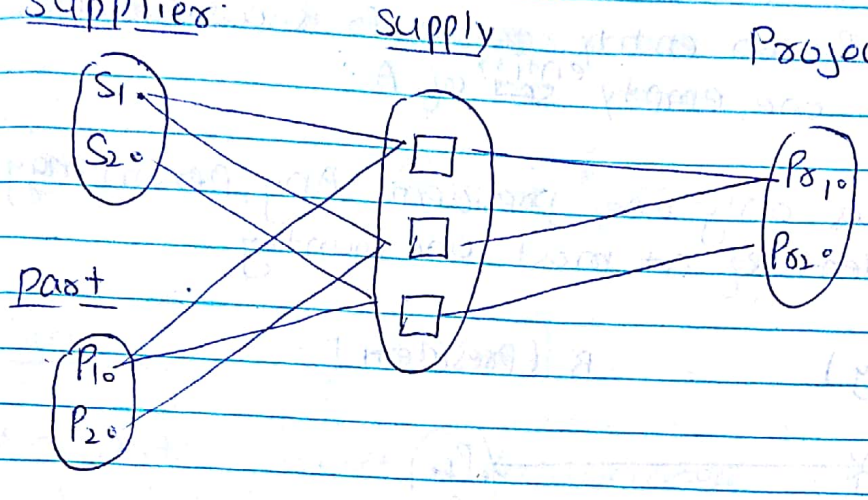
* Total no of entity sets participate in a relationship set is known as degree of that relationship set.

(i) Binary Relationship set

* A relationship set in which only two entity sets are involved is known as binary relationship set.

(ii) **Ternary Relationship Set :**

* A relationship set in which three entity sets are involved is known as ternary relationship set. or relationship set having degree three.



* **Mapping Constraints :**

→ There are certain constraints in E-R model. Data in the database must follow the constraints.

→ ~~These~~ Constraints acts as rules to which the contents of database must conform.

→ There are two types of mapping constraints :

(i) **Mapping Cardinalities (Cardinality Ratios)**

* It specifies the no. of entities of an entity set that are associated with entities of another entity set through a relationship set.

* Mapping Cardinalities are helpful in describing binary relationship sets

Two entity sets A & B having binary relationship set R must have one of the following cardinality:

(i) **One to One (1:1)**

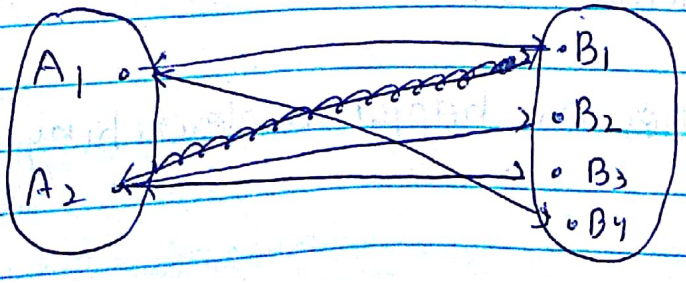
* An entity in A is associated with at most one entity in B & an entity in B is associated with at most one entity of A.
* Ex:
A country has only one president Any person may be the president of at most one country

A (country) B (President)

(ii) **One to Many (1:N)**

* An entity in A is associated with any no of entities in B & an entity in B is associated with at most entity in A.
* A (manager) Ex:
A manager has many employees under it but an employee works under only one manager

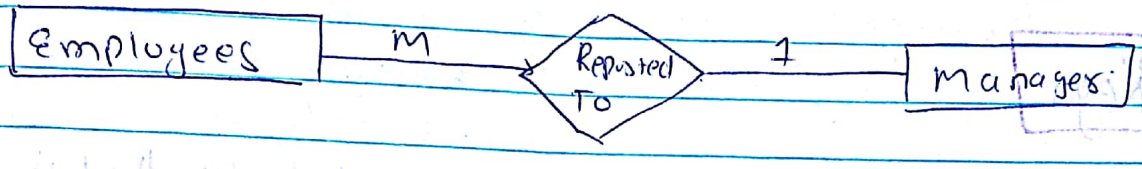
A (manager) B (Employee)



(iii) **Many to one (N:1):**

* An entity in A is associated with at most one entity in B & an entity in B is associated with any no of entities in A.

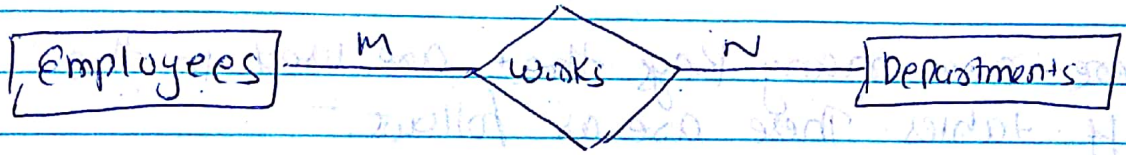
* Ex: Many employees work under one manager in department.



(iv) **Many to many (M:N):**

* An entity in A is associated with any no of entities in B & vice versa.

* Ex: many diff employees works under many diff departments.



* Adv. of E-R model.

1. Visual representation
2. Conceptual Simplicity
3. Effective Communication
4. Integration with relational database model.

* Disadvantages of E-R model :

1. Limited constraint relation
2. No relationship with data manipulation
3. Limited relationship representation.

* Keys

↳ A Key is an attribute or set of attributes that is used to identify data in entity sets.
↳ The attribute which is used as a key are known as Key attributes.
↳ Rest of all are known as Non-key attributes.

↳ **Types of Keys :**

There are many keys that are used in the diff. tables. These are as follows.

1) **Super Key**

* A super key is a set of collection of one or more than one attributes that can identify data uniquely.
* An entity ~~have~~ set have more than one super key.

Employee

Reg. no	Emp-id	E-name	E-salary	Dept-id
R_1	1	A	1500	D1
R_2	2	B	2000	D2
R_3	3	C	3000	D3

Prime att. = Cand. Key.
Non-Prime att = Att. Key.

Department

Dept-id	Dept-name.
D1	B-tech
D2	MBA
D3	BCA

Ex: In entity set employee, super keys are
(a) (Emp. ID, E-Name, E-Salary, Reg.no)
(b) (ID, E-Name, Reg.no)
(c) (Emp-id) etc.

2) Candidate Key:

* "An attribute or set of attribute that indicates data uniquely is known as candidate key"

* The minimal super key is known as candidate key.

* Ex: Emp-id & Reg.no are candidate key.

3) Primary Key:

* "An attributes which identifies data uniquely is known as primary key".

* An entity set can have more than one candidate key but only one primary key.

* Types: Prime

(a) * Ex: Either Emp-id is primary key or Reg.no " " "

4) **Alternate Keys**

* All Keys except primary Key is known as alternate Key.

* Ex: if a take emp-id as primary Key then Reg.no as alternate Key.

5) **Secondary Key:**

* An attribute or set of attributes which doesn't identify data uniquely but identifies an group of data is known as secondary Key.

* Ex: Emp-name, E-salary & Dept no. are all sec. Keys.

6) **Foreign Key:**

* An "foreign Key is an attribute in any entity set which is also a primary Key in any other entity set.

* Ex: Dept-id in employee table is a foreign Key & in Department table is primary Key.

* **Types of Entity Sets**

1) **Weak Entity Set**

↳ An entity set which may not sufficient attribute to form a primary Key is called weak entity set.

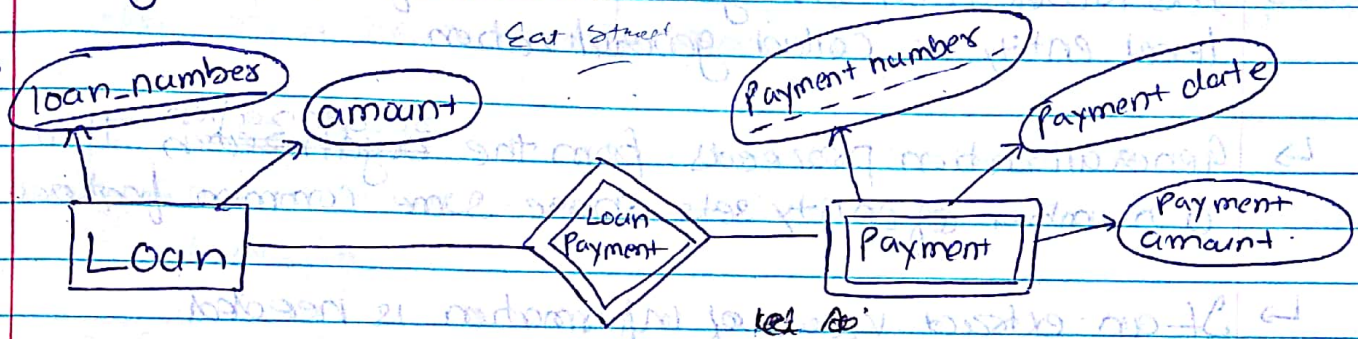
↳ weak entity set is represented by double rect. 

↳ An weak entity set is always associated with another entity set to make it ~~meani~~ meaningful.

↳ This another entity set is known as identifying entity set.

2) **Strong entity set.**

↳ An entity set which have a primary key is called Stong entity set.

↳ 

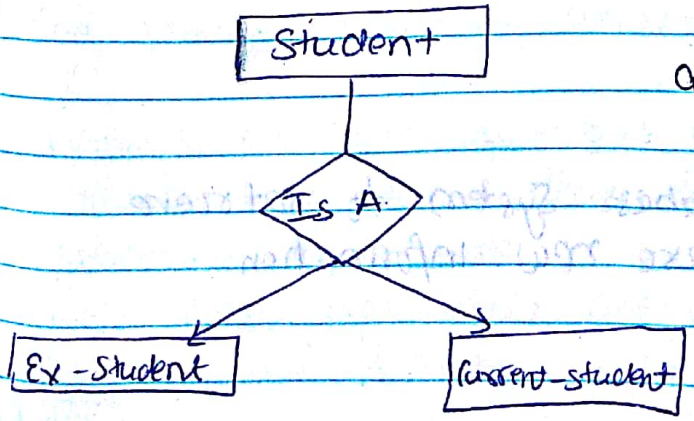
* **Extended E-R features**

1) **Specialization** : (top to bottom approach ↓)

↳ Specialization means creating new subclass from the existing class.

↳ Top-down design proces we designate subgroupings within an entity set that are distinct from other entities in the set.

↳ If deep information is needed, then go towards specialization.



a ~~person~~ student can be either a current student or ex-student. So, here student & Ex-student are known as disjoint super subclass.

2) Generalization: (bottom to top ↑ approach)

↳ The process of extracting common characteristics from two or more classes then combine to form a generalized superclass is called generalization.

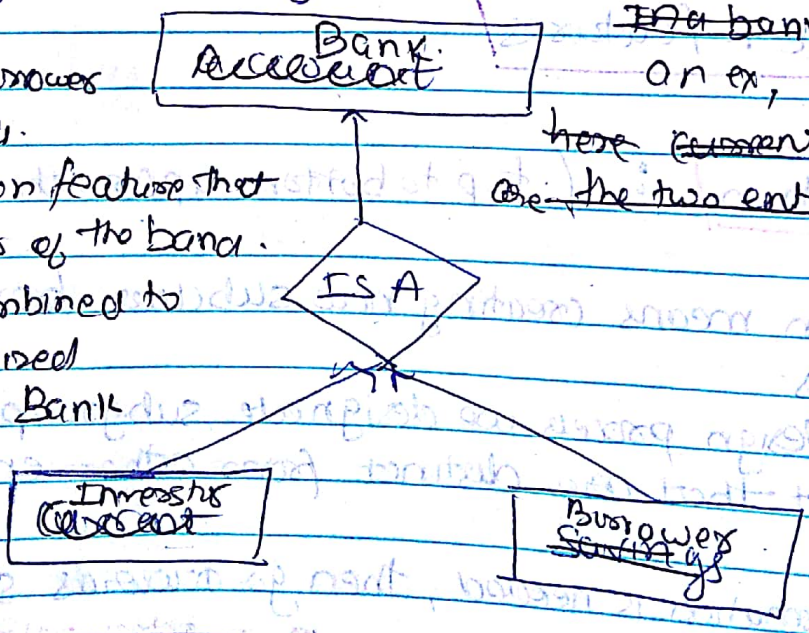
↳ This is bottom up design process.

↳ The lower level entity combined to form an higher level entity is called generalization.

↳ Generalization proceeds from the ~~organization~~ recognization that a number of entity sets share some common features.

↳ If an extract new of information is needed then go towards generalization.

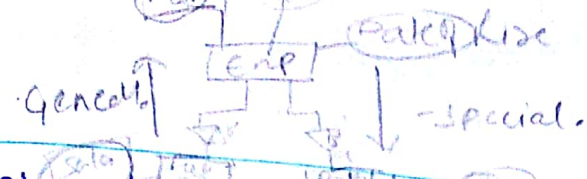
Here, Investor & Borrower are two entity sets. They have common feature that both are customs of the bank. & they both combined to form a generalized superclass called Bank.



~~In a bank~~, Consider an ex, here ~~current & savings~~ are ~~the two entity sets~~.

* Database Users.

↳ A primary goal of database system to retrieve information from & store new information in the database.



* People who work with a database can be categorized as database users

Database Users & Users Interfaces:

four types of database-system users.

① **Naive Users**: are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

Eg: A bank teller who needs to transfer $50 from account A to account B invokes a program called transfer

② **Application Programmers**: are computer professionals who write application programs.

* They can choose from many tools to develop new interface
Rapid Application Development (RAD) tools are tools that enable an application programmer to construct forms & reports with minimal programming effect.

③ **Sophisticated Users**: They interact with the system without writing programs.

* Instead they form requests in a database query language.
* They submit each query to a query processor, whose function is to break down DML statements into instructions that the storage manager understands.

④ **Specialized Users**: are sophisticated users who write specialized database applications that do not fit into the traditional data processing framework.

* Among these applications are computer-aided design systems, Knowledge base & expert systems.

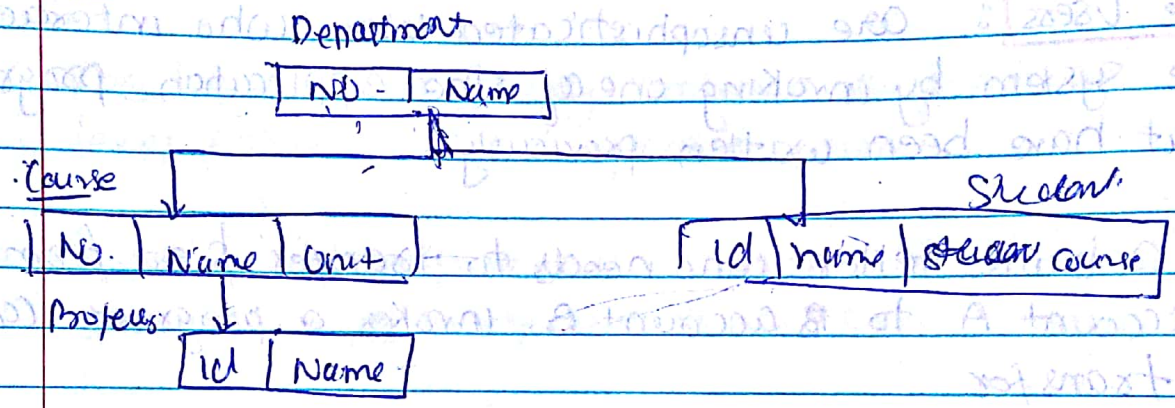
⑤ DBA

* **Data Models**

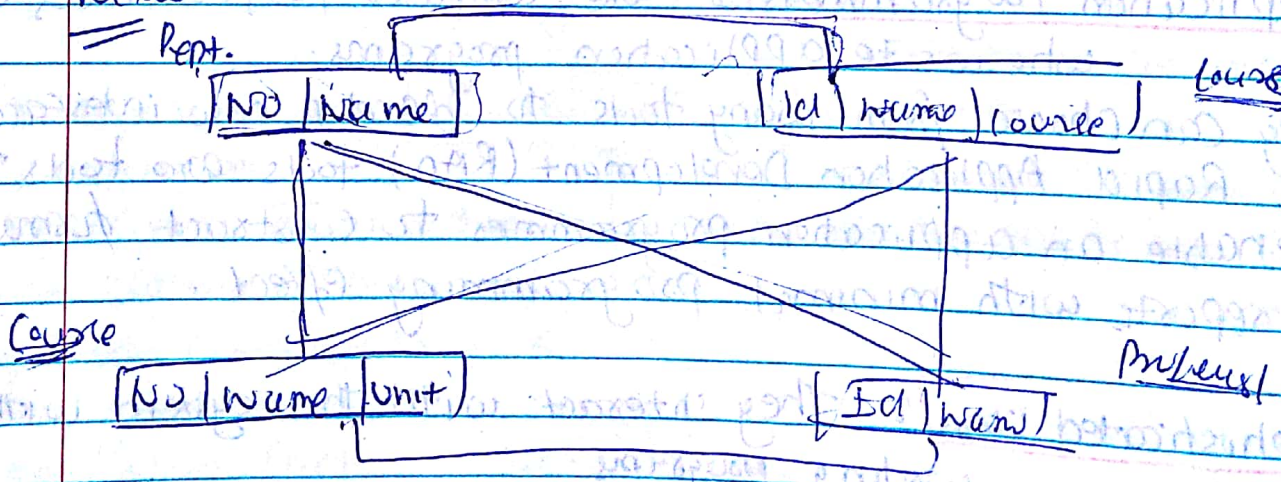
→ Data models defines how the logical structure of a database is modeled.

→ Data models are fundamental entities.

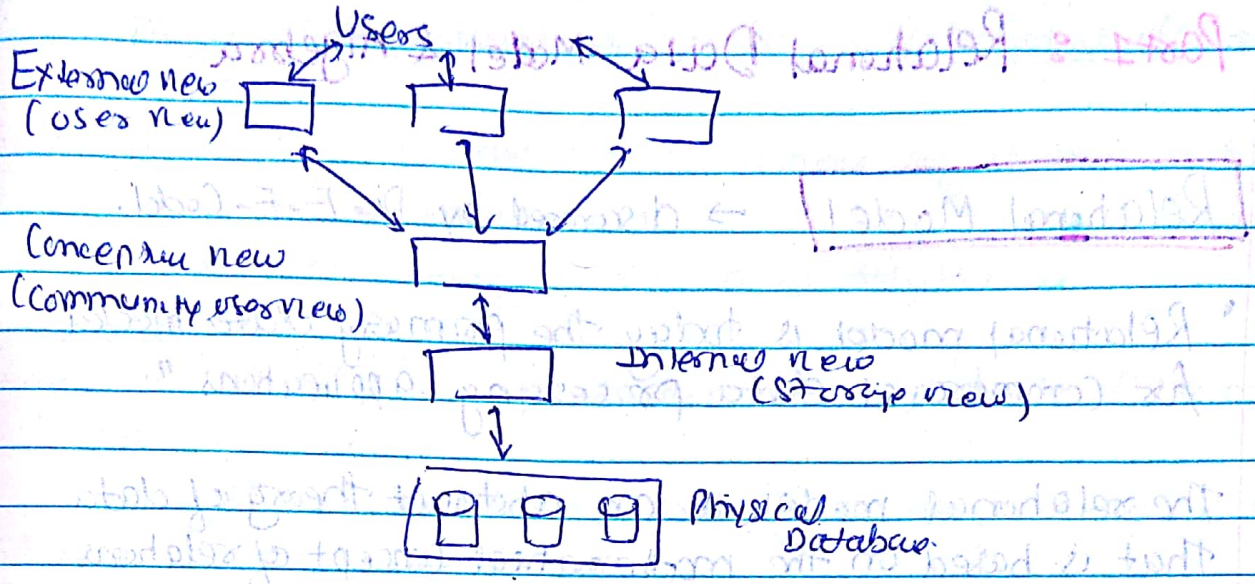
Hierarchy



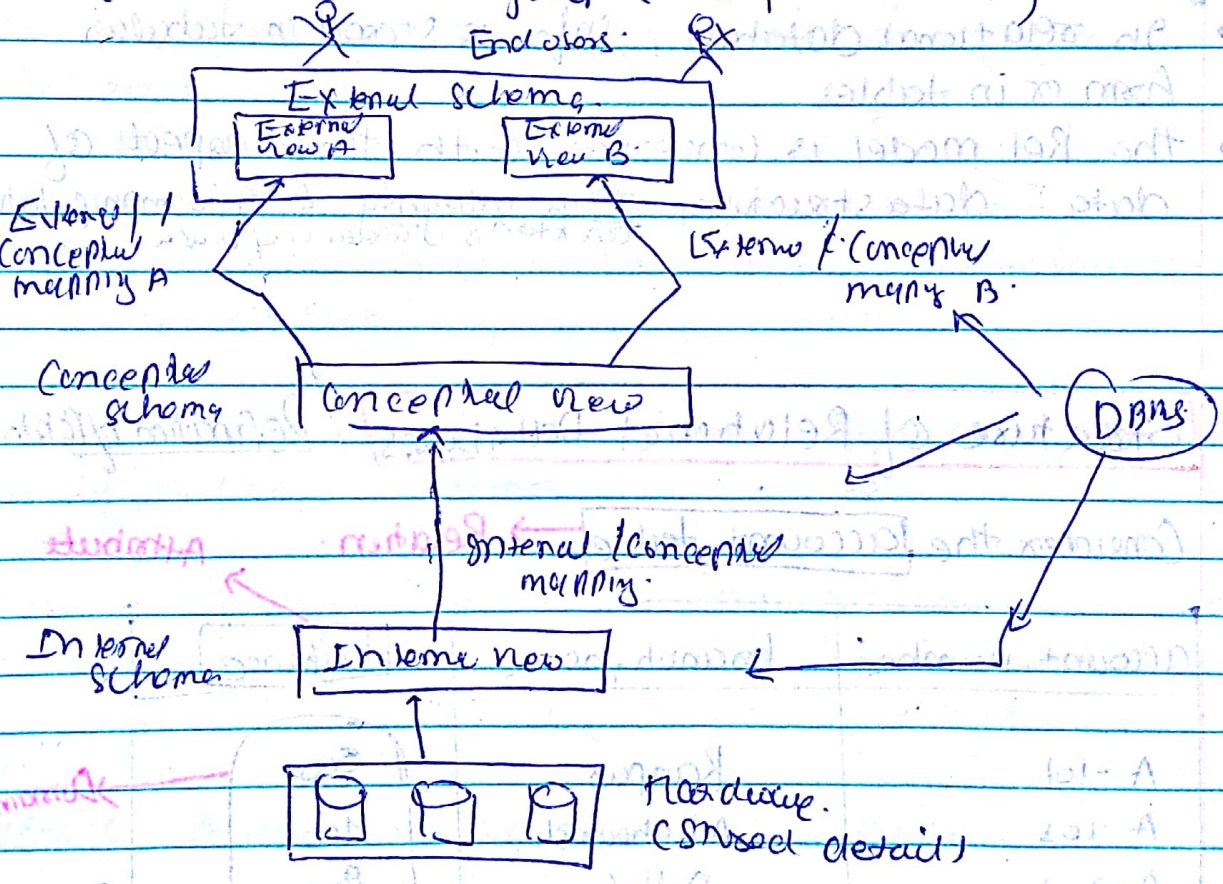
Network



Three level Architecture



(3-level arch. graphic representation)



(3-level Arch. @ detailed Report)