

## Data:-

Raw facts and figures

↓  
Unprocess

Normally data store in texts, no.s, audio, videos etc.

Information:- Process form of data

Q# Differ blw data information?

## Database:-

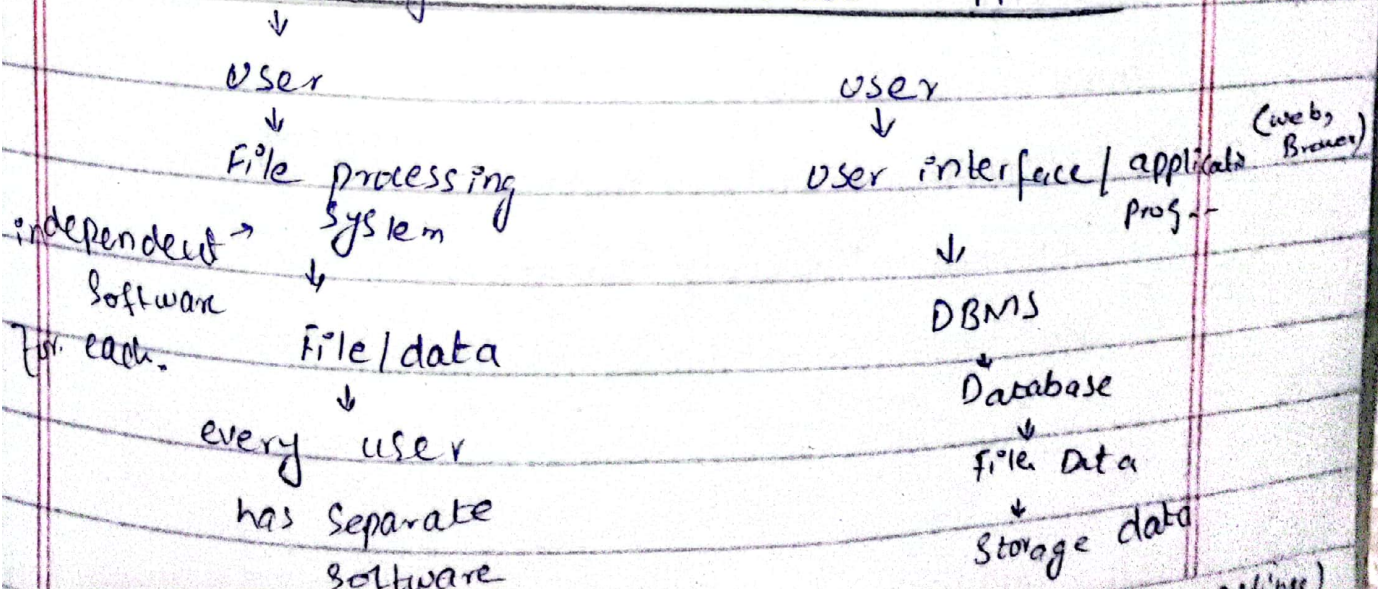
Is a collection of interrelated or objects.

## DBMS:-

Are basically tools which Provides environment where we perform activities. DBMS are Softwares.

cms  
campus  
management  
solution.

## File Processing vs Database Approach:-





user:-

- sophisticated user → (Database administrator)
- naive user → Bresner  
New User

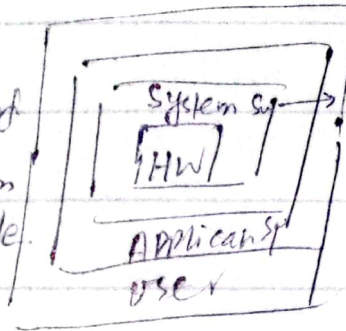
create  
read  
update  
delete

Conceptual Schema

DNS → convert address

firewall → is used as check point in IP address  
encrypt  
cryptolayer  
to check data

RDBMS  
DBMS → Data stored in the form of table.  
OS operating system



OS → (bavel krunna data ko)

Application layer  
↓  
to perform specific task

3 level Schema (Imp Question)

Mapping → link

(connection of two adjacent layer)

- External to conceptual mapping
- Conceptual to internal mapping

Data Independence:-

- logical data independence
- Physical data independence



## Data Models:-

Data models are representation of data in real environment.

- Hierarchical model → Top down hierarchy
- Network model → Bottom up hierarchy
- Object-oriented model → link objects
- relational model (RDBMS) → Data stored in tables.  
DBMS → tools environment

Table → Relation

Row → Tuple (Record)

(attributes) Column → field

Degree → no. of columns

cardinality → no. of rows

Key attribute (numeric must be)

## DBMS Functions:-

Transaction:-

No. of steps taken to perform an activity.

\* Components of Data Base System:-

• Data

• Database

• DBMS

• Query processor

(Question about data)

DDC

DAC

Catalog  
List of categories



Date: / / 20

• Data Dictionary (rules stored in (data dictionary) DBMS)

Build in External

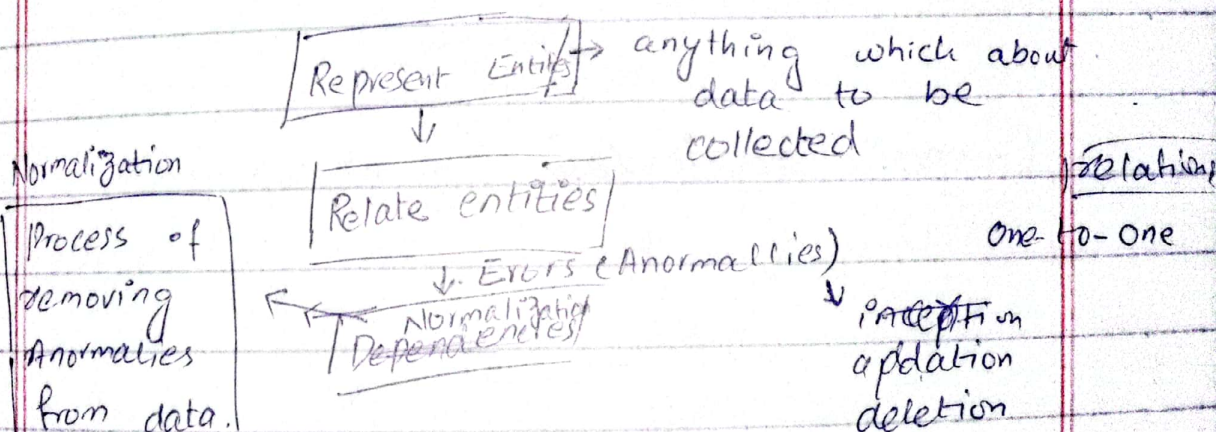
• Users { Sophisticated (Data base administrator, Engineer etc)  
Main

SDLC  $\Rightarrow$  System Development Life Cycle

[ER Diagram]

Stage Based approach

logical database development (Imp)



Chapter #03

RDBMS: (Relational model)

table  $\rightarrow$  relation

Data stored in table form so it is called RDBMS.

Entities  $\rightarrow$  Attributes  $\rightarrow$  Relation



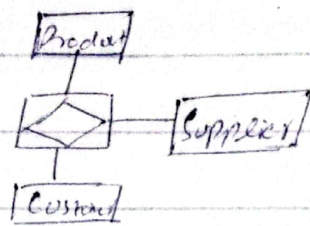
## Types of Entities:-

=> Strong entity (Independent)   

=> weak entity (derived from other entity)  
↓ (dependent)

   → representation

=> Associative entity:- It is used to join two or more entities together.

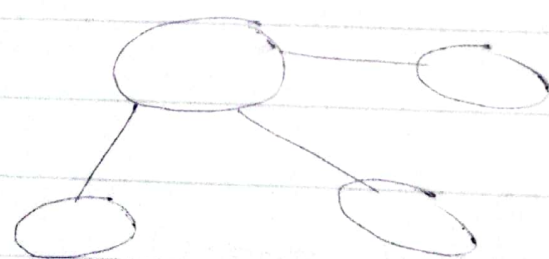


**Attributes:-** characteristics of entities like name product etc.

representation:-



**Composite attributes:-** if other attributes compose from one attribute.



**Derived attributes:-**

Attribute derived from other.

(Key attribute) **identifier attribute**

The attribute which make uniquely not duplicate representation: =



Key attribute / Non Key attributes (other than identifier)  
 entity instance:-

Single occurrence from the whole (product) entity.

Types of relation:-

- Unary relationship
- Binary relationship
- Ternary

1. If the instance of <sup>one entity</sup> ~~at~~ class associated with the instance of <sup>one class</sup> ~~one class~~ is called unary type of relationship (one entity.)

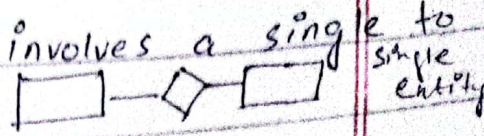
2. Binary (Two entity) (Preferred more)  
 If the instance of one entity class ~~as~~ more associated with one entity or more entity of other class is called binary.

3. If three entities include or involved then called ternary relationship.

Instance  
↓  
occurrence

Further types of Binary:-

one to one relationship (involves a single to single entity)  
 one to many relationship  
 many to many relationship



Cardinality constraints:



Date: 1/20

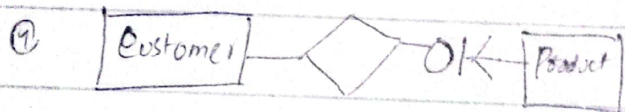
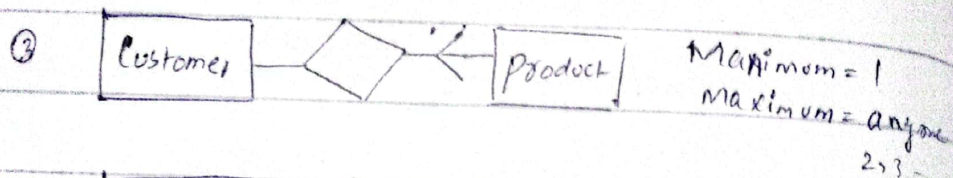
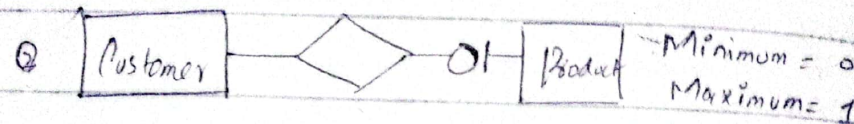
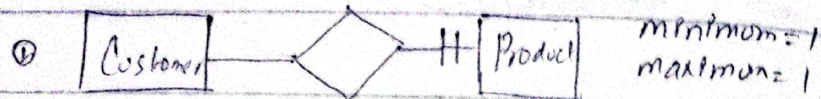
Day:

|| Mandatory one

o| optional one

(crowfoot) <= many

o<= optional many



① minimum cardinality: one product  
Maximum cardinality: one

Degree of relationship

EF Code (12 rule)

If follow all rules then perfect relation.