

# Important Topics Notes of DBMS :-

## Unit → ① (Introduction)

Q1 what is Data Independence, explain in detail & its types?

⇒ Data Independence: It is a characteristic that allows you to change the schema at one level of the database system without changing the other schemas of another level.

⇒ These are 2 types of Data Independence:

- 1) Logical Data Independence
- 2) Physical Data Independence

1) Logical Data Independence:

→ It refers to the characteristic that modify the Conceptual Schema without disturbing the external schemas.

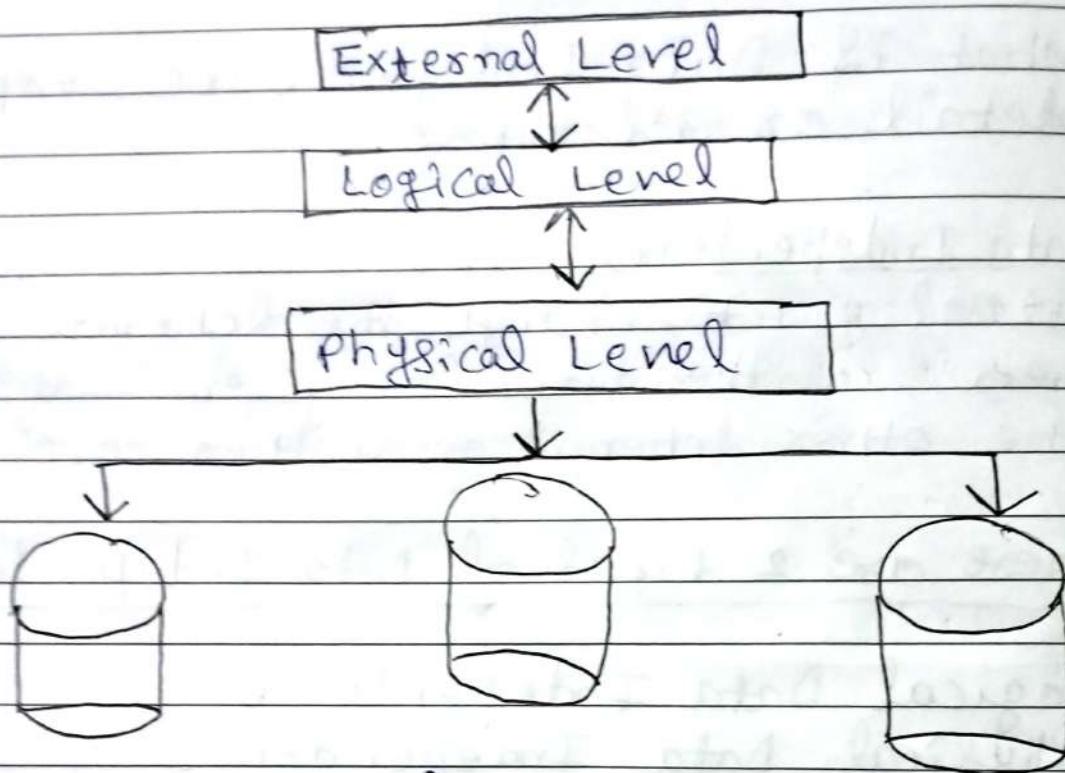
→ It is used to separate the Conceptual Schema from external Schema.

→ It occurs at the user-Interface level.

2) Physical: It refers to the characteristic that modify the internal Schema without affecting the conceptual schema.

→ It separated the conceptual schema from internal schema.

→ It occurs at the logical interface level.



Storage - Data Bus

→ why Data Independence Important?

It can enhance the database security by allowing security measures controls to be implemented at the logical level.

→ Advantages of Data Independence:

- 1) Improved the data quality.
- 2) It makes database system more efficient.
- 3) It makes database system more flexible or affordable.
- 4) It saves time or cost when changing the database.

## → Disadvantages of Data Independence:

- 1) It might be confusing for development as there are complexities at multiple level of complexities.
- 2) When the extra layer is added to the code, the navigation becomes challenging.

Q2 What are Primary / Candidate or Super key with Example?

⇒ ① Primary Key: A primary key is a column in a relational database table that uniquely identifies each record.

for Eg: In a school database, the roll number of each student uniquely identifies each student's identity.

② Candidate Key: A candidate key is a minimal set of attributes that can uniquely identify a tuple in a table called candidate key.

for Eg: If  $(A, B, C)$  is a candidate key, then  $(A, B, C)$  or any combination of them can't be a super key.

③ Super Key: Super key is an attribute or a set of attributes that can uniquely identify a tuple.

for Eg: In an employee table for (Employee ID, Employee - Name), the name of

of 2 employees can be same, but their employee-ID can't be the same.

Q3 what is the difference Primary or Candidate Key?

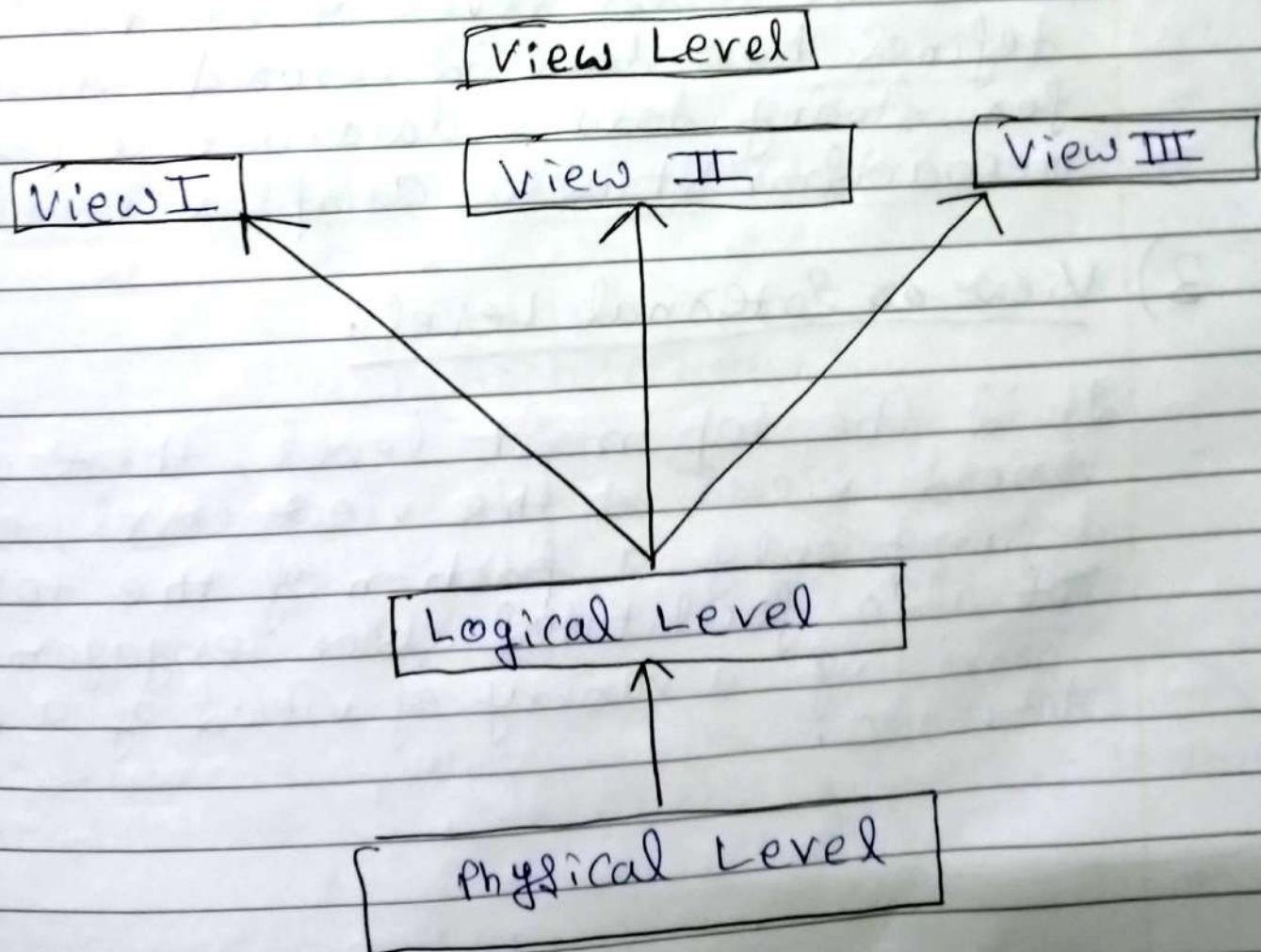
⇒ Primary Key	Candidate Key
1) It is a minimal super key so, there is one and only 1 Primary Key in relation.	In relation, there can be more than 1 primary candidate key.
2) The attribute of Primary key can't contain Null value.	It's attribute can be contain Null value.
3) It can be optional to specify any relation.	Without candidate key, there can't be specified any relation.
4) It specifies the important attributes for the relation.	It specifies the keys which qualified for Primary Key.
5) Primary Key is a Candidate Key.	Candidate key may be a Primary Key.

Q4 what is Abstraction, explain in detail in term of DBMS?

⇒ Abstraction: It means hiding unnecessary background details from the end user to make the accessing of data easy & secure.

→ These are 3 levels of Data Abstraction:

- 1) Physical / Internal Level.
- 2) Logical / Conceptual Level.
- 3) View or External Level.



1) Physical

1) Logical Level:

It is the next highest level of intermediate level. It explains what data is stored in the database & how those data are related. It is less complex than physical level.

2) Physical Level:

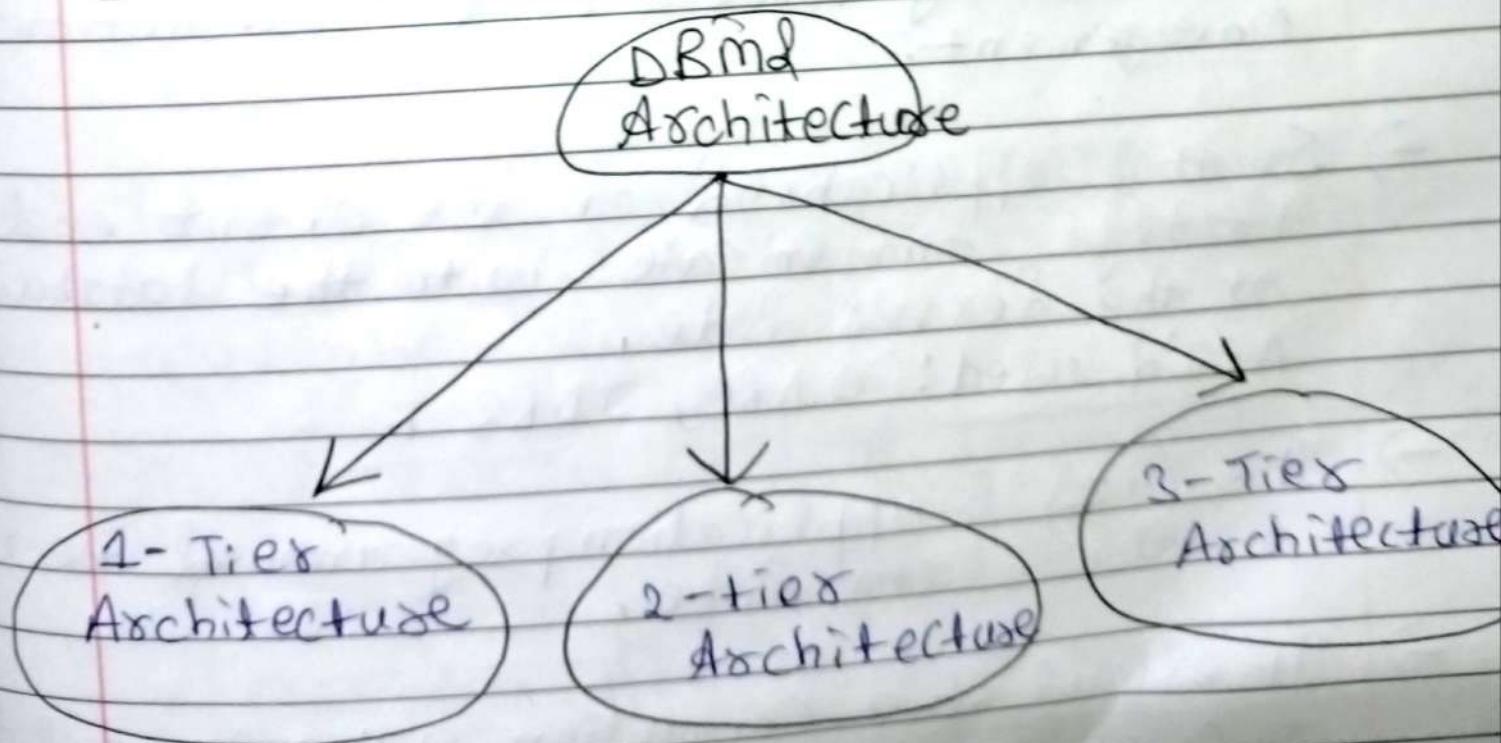
It is the lowest level of abstraction, defines how data is stored, database for storing data & database access mechanism. It is a complex level.

3) View or External Level:

It is the top most level, there are various views at the view level, each view defining only a portion of the total data. It also facilitates user engagement by providing a variety of views of a single database.

## Q5 Architecture of DBMS?

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PC's, web servers, that are connected with network.
- The client/server architecture consists of many PC's and a workstation connected to a network.
- It depends upon the user connected to the database to get their request done.
- Types of DBMS Architecture:



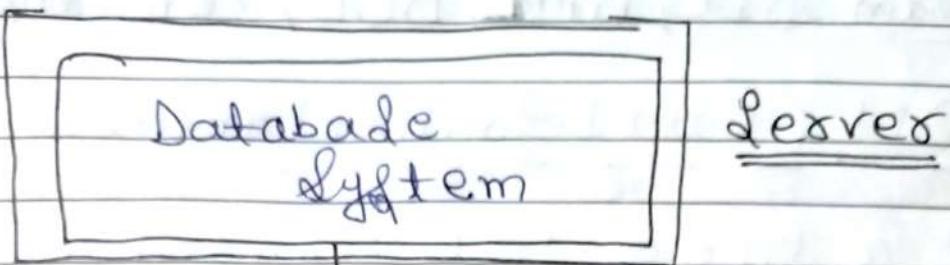
## 1) 1-Tier Architecture:

- In this, the database is directly available to the user, it means user directly access into it.
- Any changes done here will be done on database also.
- It is used for development of the local application where programmers can directly communicate with the database.

## 2) 2-Tier Architecture:

The server side

- ~~It is~~ responsible for the functionalities like: query processing & Transaction Management.
- In this application on the client end can directly communicate with the database at the server side.  
API's used: ODBC, JDBC.
- The (UI) & application programmed are run on the client side.
- It establish a connection with a server side to communicate with (DBMS) by client side.

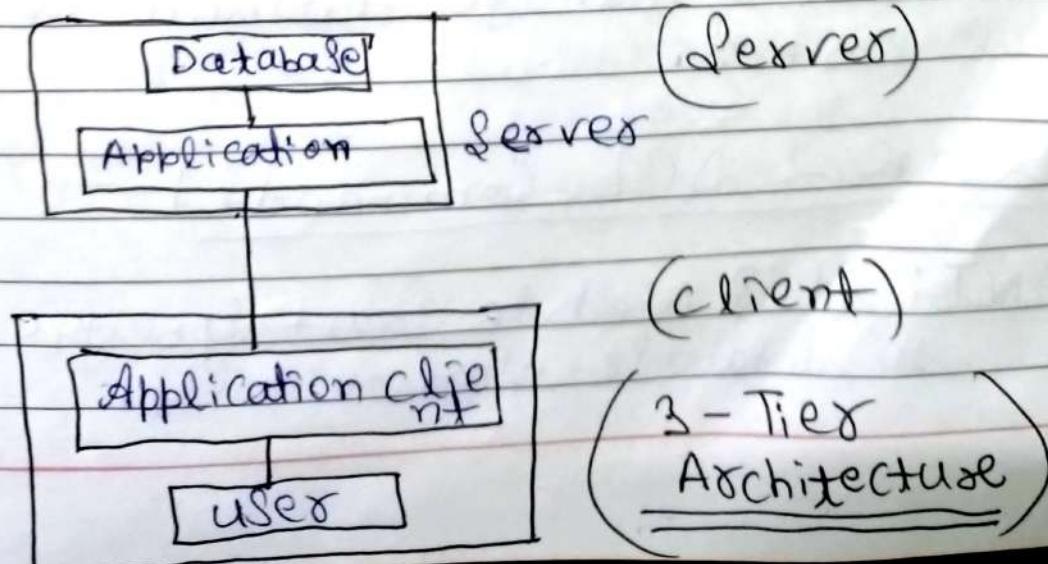
Server

2-Tier

Architecture

### 3-Tier Architecture:

- The 3-Tier Architecture contained another layer b/w the client & server.
- In this end user had no idea about the existence of the database beyond the application server.
- It is used in case of large web Applications.



Q6 Explain DDL, DML, DCL, ^ Commands;

### ⇒ ① DDL (Data Definition Language)

- It is a type of SQL Command used to define data structures and modify data.
- It creates/alter/ & Deletes database objects such as Tables, views, indexes & users.  
for eg: CREATE, ALTER, DROP, TRUNCATE.

### ② DML (Data Manipulation Language):

- It represents a collection of programming languages explicitly used to make changes in database.
- Such as: CRUD operations to create, read, update, & delete data. Using the Insert, Select, update & Delete commands.

### ③ DCL (Data Control Language):

- It is used to manage database security & access control.
- There are 2 DCL Commands:

a) GRANT: It is used to grant specific privileges to database user or roles.

B) Revoke: It is used to revoke previous granted privileges.

#### ④ TCL (Transaction Control Language):

- It is used to control the Transaction process or Transactional Security.
- It includes:
  - a) Commit: It commits a transaction, saving changes permanently.
  - b) Roll Back: It undoes changes made during a transaction.
  - c) Save point: Sets a point within a transaction to which you can later roll back.

#### Q7 Database Schema?

⇒ Schema: The schema is the skeleton of the database created by the attributes called schema.

Customer
Customer-id
Customer name
Purchased Item

(Detailed of a Customer)

⇒ Database Schema:

- It is a logical representation of data that shows how the data in the database should be stored logically.
- It contains; Tables/ fields/ views & relations b/w keys.
- It provides the organization of the data & the relationship b/w the stored data.
- ⇒ These are 3 main types of Database Schema:

- 1) Physical Database Schema,
- 2) Logical Database Schema.
- 3) View Database Schema.

1) Physical Database Schema:

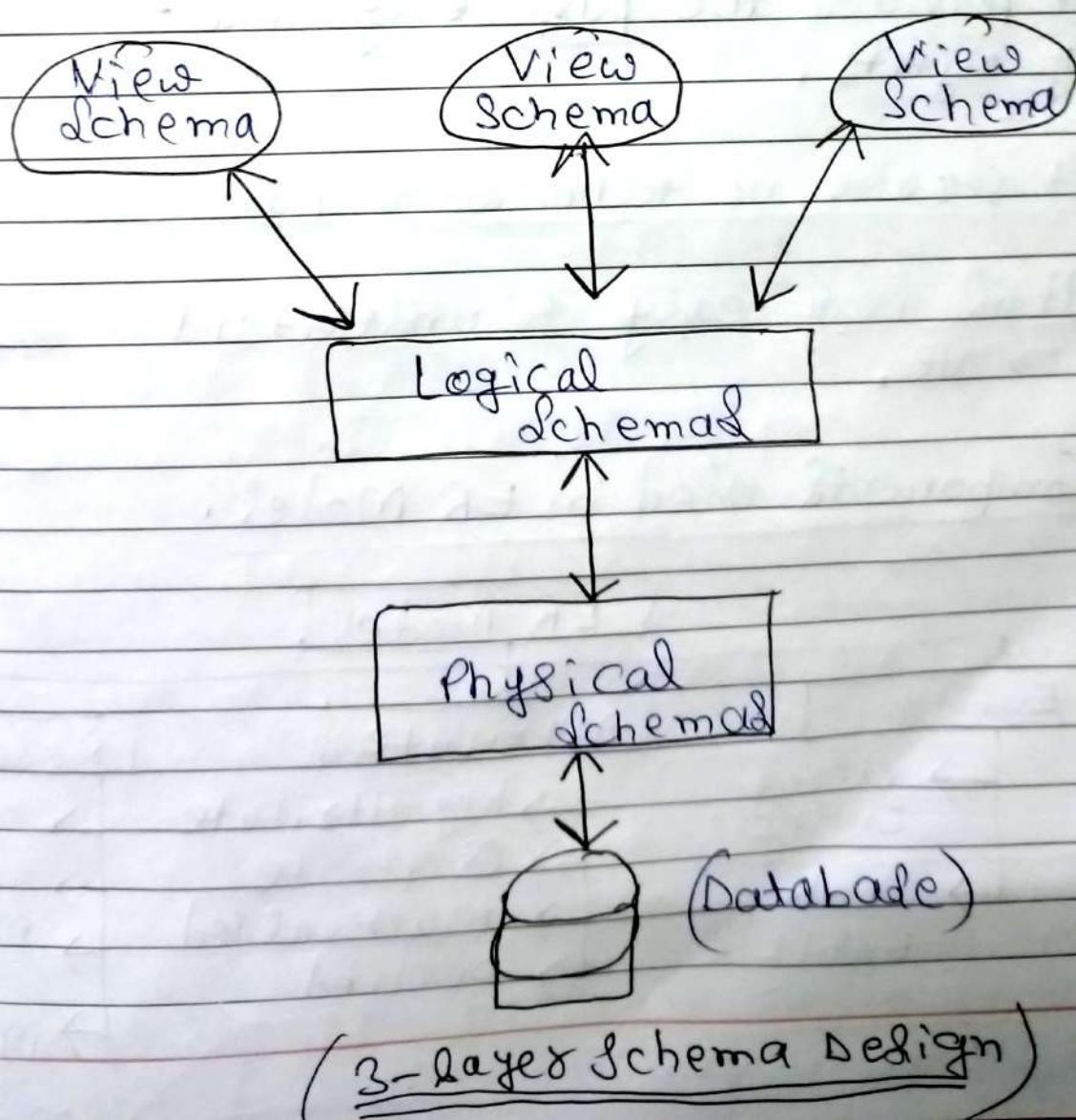
- It defined how the data or information is stored physically in the storage system in the form of files using syntax.
- The database administrator chooses that how & where to store the data in which block of storage.

- 2) Logical: → It defined all the logical constraints that need to be applied to the stored data, & also describes tables, views, entity, etc.

- It described how the data is to stored in the form of tables & how the attributes of a table are connected.
- The integrity constraints are defined in order to maintain the quality of data.

### 3) View Database Schema:

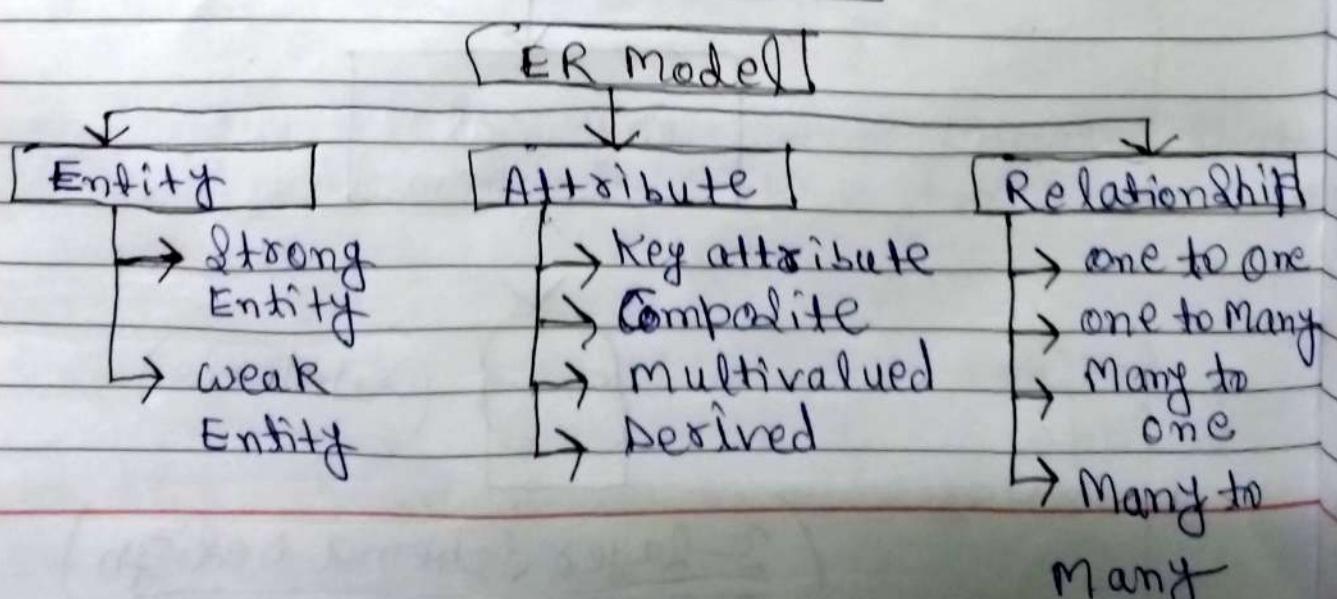
- It is a view level design which defines the interaction b/w end-user & database.
- User is able to interact with the database with the help of the interface.



Q8

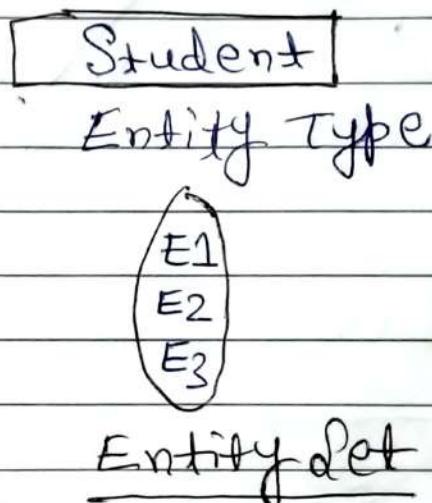
what is ER model & its components?

- ⇒ ER Model: The (ER) Entity Relationship model or high-level data model that define elements and relationships model for a system.
- ⇒ why we use ER model:
- ER diagram are used to represent the E-R model in a database, which makes them easy to be converted into relations.
- It provides the purpose of real-world modeling of objects.
- It requires no technical or hardware support.
- It is very easy to understand & easy to create.
- ⇒ Components used in ER Model:



⇒ What is Entity:

- The entity is an object with a physical existence for eg: Person, Car, Employee, etc.
- Entity Set: It is an object of Entity type & a set of all entities is called an Entity set.  
for eg: (E1) is an entity type & a set of all entities is called an Entity set.  
for example, E1 is



⇒ Types of Entity:

1) Strong Entity:

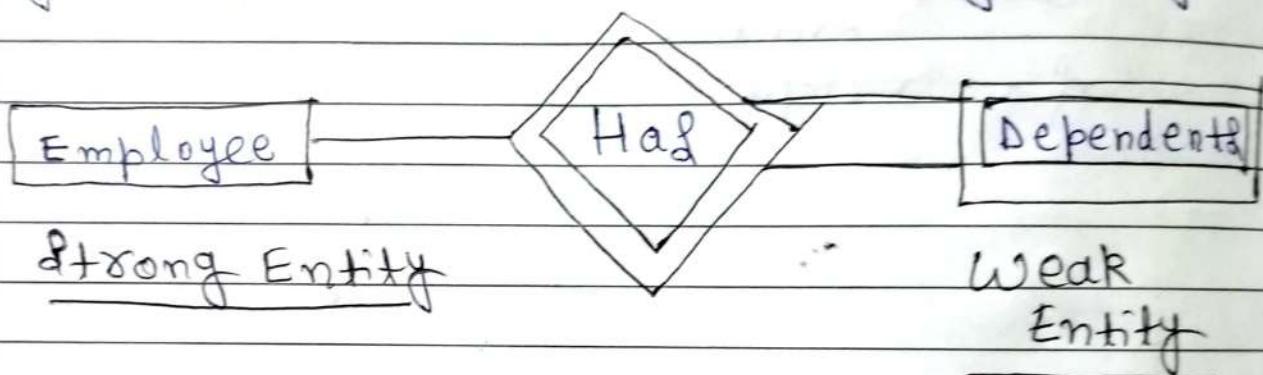
It is an entity that has key attribute. It does not depend on the other entity, it has a primary key or represented by rectangle.

2) Weak Entity:

An entity type has a key attribute that uniquely identified each entity in the entity set. But weak Entity key attribute can't

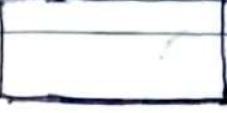
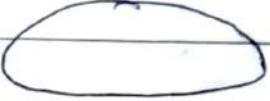
be defined, these are called weak Entity set.

for eg: A Company stores their customer info (parents/childrens / spouse) of an employee. But the dependents don't have existence without the employee. So, dependent will be a weak Entity Type & Employee is a identifying Entity type for dependent & also a strong entity.



=> Symbols used in Entity Relationship model:

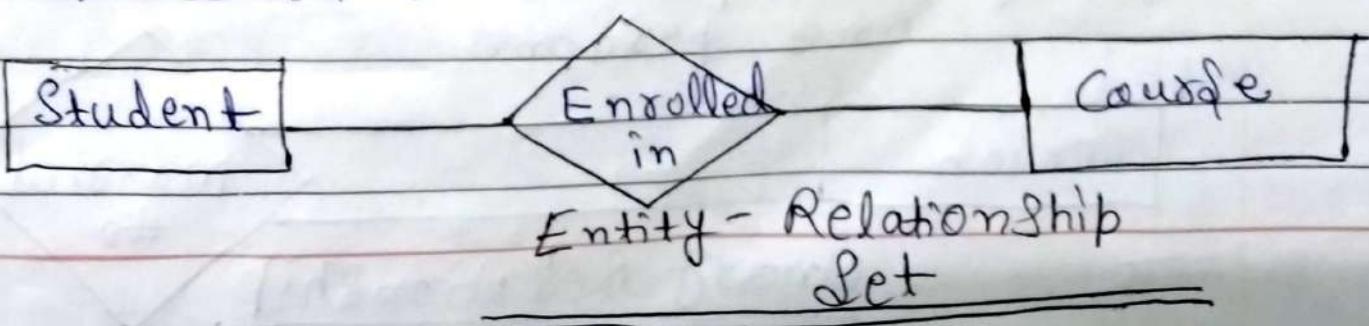
- 1) Rectangle: Represents Entities in the ER model.
- 2) Ellipses: Represents attributes in the ER model.
- 3) Diamond: Represents relationship among Entities.
- 4) Lined: Represents attributes to the entities & entity sets with other relationship types.
- 5) Double Ellipse: Represents multi-valued attributes.
- 6) Double Rectangle: Double Rectangle represents a weak Entity.

Figure	Symbol	Represented
Rectangle		Entities in ER model.
Ellipse		Attributed in ER model.
Diamond		Relationship among entities.
Lined		Attributed to entities and entity sets with other relationship types.
Double Ellipse		Multi-valued Attributed
Double Rectangle		weak Entity

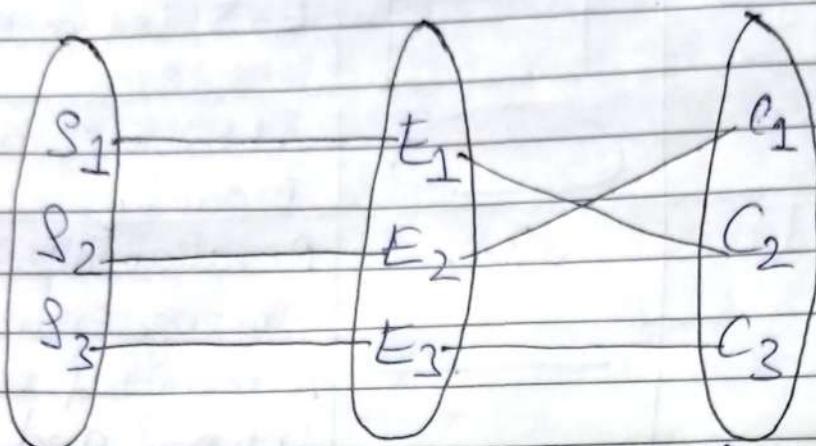
⇒ Relationship Type & Relationship Set:

It represents the association between entity types.

for eg: "Enrolled in" is a relationship type that exists b/w entity type student & course.



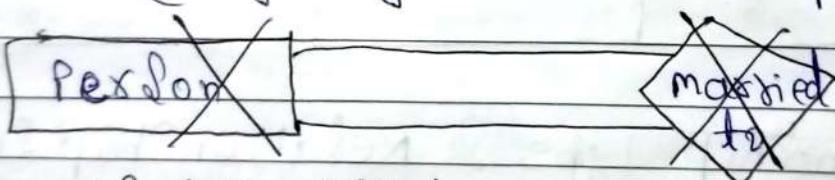
→ A set of relationship is called Relationship Set.



→ Relationship set depicts  $S_1$  as enrolled in  $C_2$ ,  $S_2$  as enrolled in  $C_1$ , &  $S_3$  as registered in  $C_3$ .

⇒ Degree of a Relationship set:

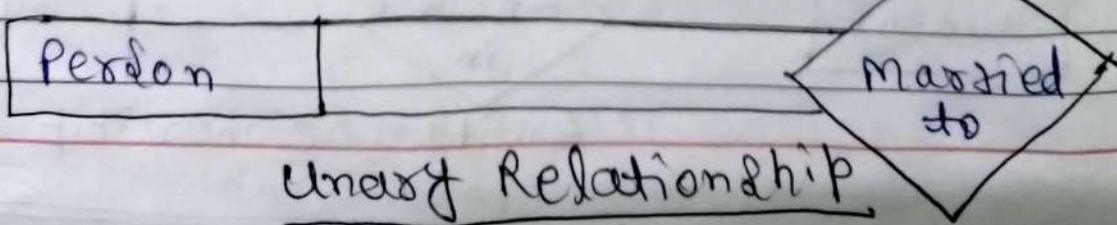
The no. of different entity sets in a relationship set called Relationship set called (Degree of a Relationship set),



1) Unary Relationship:

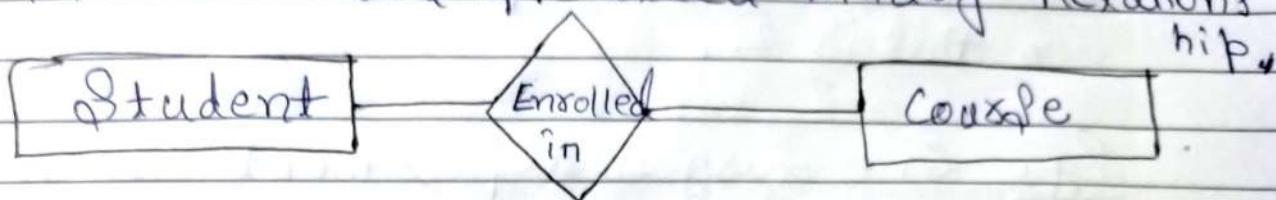
When there is only one entity set in a relation, the relationship is called a unary relationship.

for eg: one person is married to only one person.



## 2) Binary Relationship:

When there are 2 entities let participating in a relationship, called Binary Relations hip.



## Binary Relationship

## 3) n-ary Relationship:

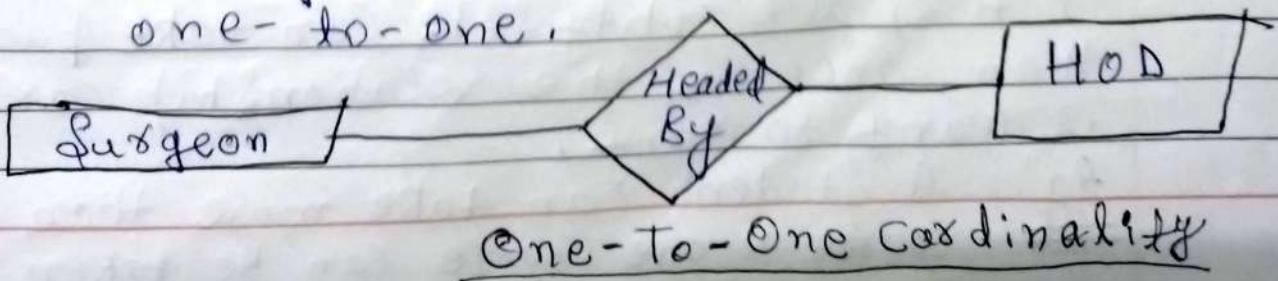
When there are (n) entities let part in a relationship called n-ary relationship.

⇒ Cardinality: It is the no. of times an entity in a set comes in a relationship set called Cardinality.

## Types of Cardinality:

### 1) one - To - One:

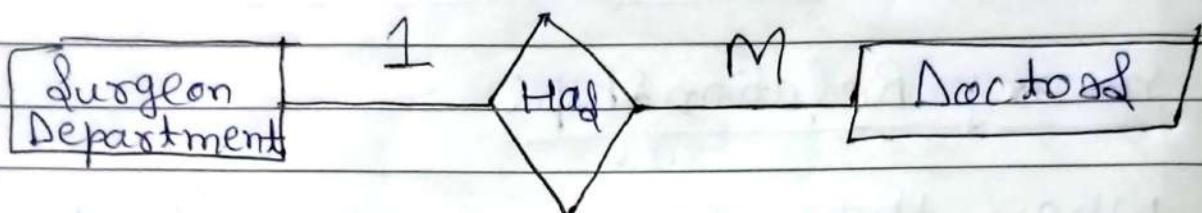
It means each entity in a set can take part only once in a relationship.  
for eg: Assume a male can marry a one female & a female can marry one male, relationship will be one-to-one.



## 2) One-to-Many:

In this, each entity can relate to more than one relationship & the total no. of tables in this is 2.

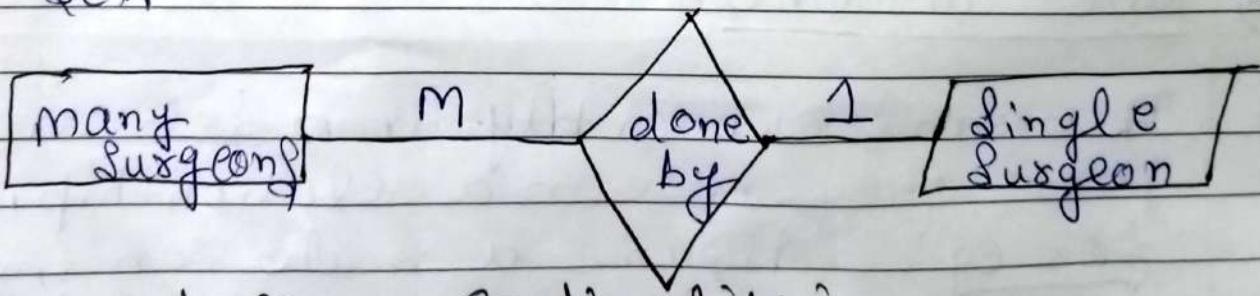
Eg: One surgeon department can accommodate many doctors. Cardinality will be 1 to M.



One-to-many Cardinality

## 3) Many-to-Many Cardinality:

It means the entity of one entity set can take part only once in the relationship set, & entities in other entity sets can take part more than once in the relationship set.

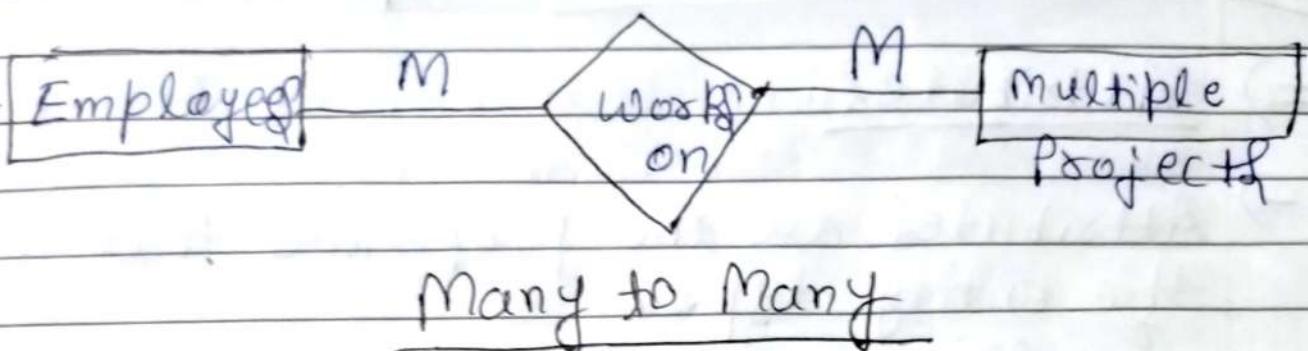


## 4) Many to Many Cardinality:

In this, all entity sets can take part more than once in the relationship cardinality is Many to Many.

Eg: A student can take more than one course & one course can be taken by many students.

student,



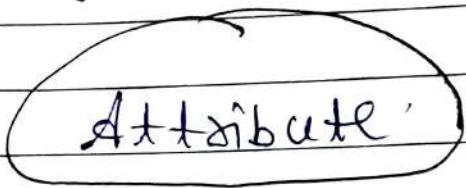
⇒ How To Draw (ER) Diagram:

- 1) The 1st step is to identifying all the entities and place them in a rectangle.
- 2) The next step is to identifying relationship b/w them & place them using diamond & make sure that, relationship are not connected to each other.
- 3) Attach attributed to the entity properly.
- 4) Remove redundant entities & relationship.
- 5) Add proper color to highlight the data present in the database.

Q9 What are Attributes, explain in detail?

⇒ Attributes:

- Attributes are the properties that define the entity type.  
for eg: Roll-No, Name, DOB, Age, etc.
- Represented by an oval shape.



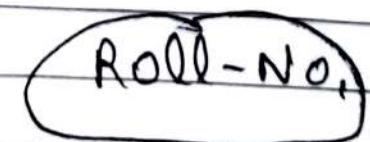
Attribute

⇒ Types of Attributes:

1) Key Attribute:

The attribute which uniquely identifies each entity in the entity set called the Key attribute.

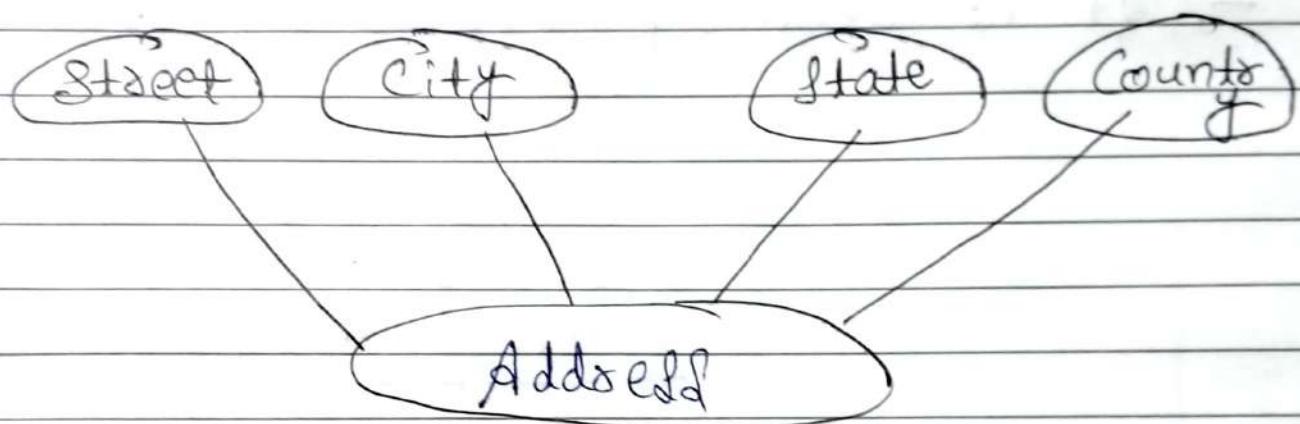
For eg: Roll-No, unique for each student.



Key attribute

## 2) Composite Attribute:

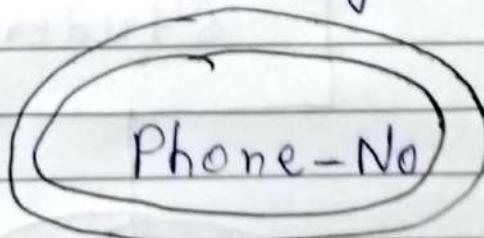
An attribute composed of many other attributes called composite attributes.  
for eg: The address attribute of the student entity type consists of Street, city, state, & country.



## Composite Attribute

## 3) Multivalued Attribute:

- An attribute consisting of more than one value for a given entity.  
for eg: Phone-No. (can be more than one for a given student),
- It is represented by double oval.



## Multivalued Attribute

#### 4) Derived Attribute:

→ An attribute that can be derived from other attributes of the entity type.

Eg: Age (can be derived from DOB)

→ It is represented by dashed oval.

( Age )  
---  
Desired Attribute

⇒ The Complete Entity Type of Student:

