UNIT- 1

INTRODUCTION TO DATA BASE SYSTEMS

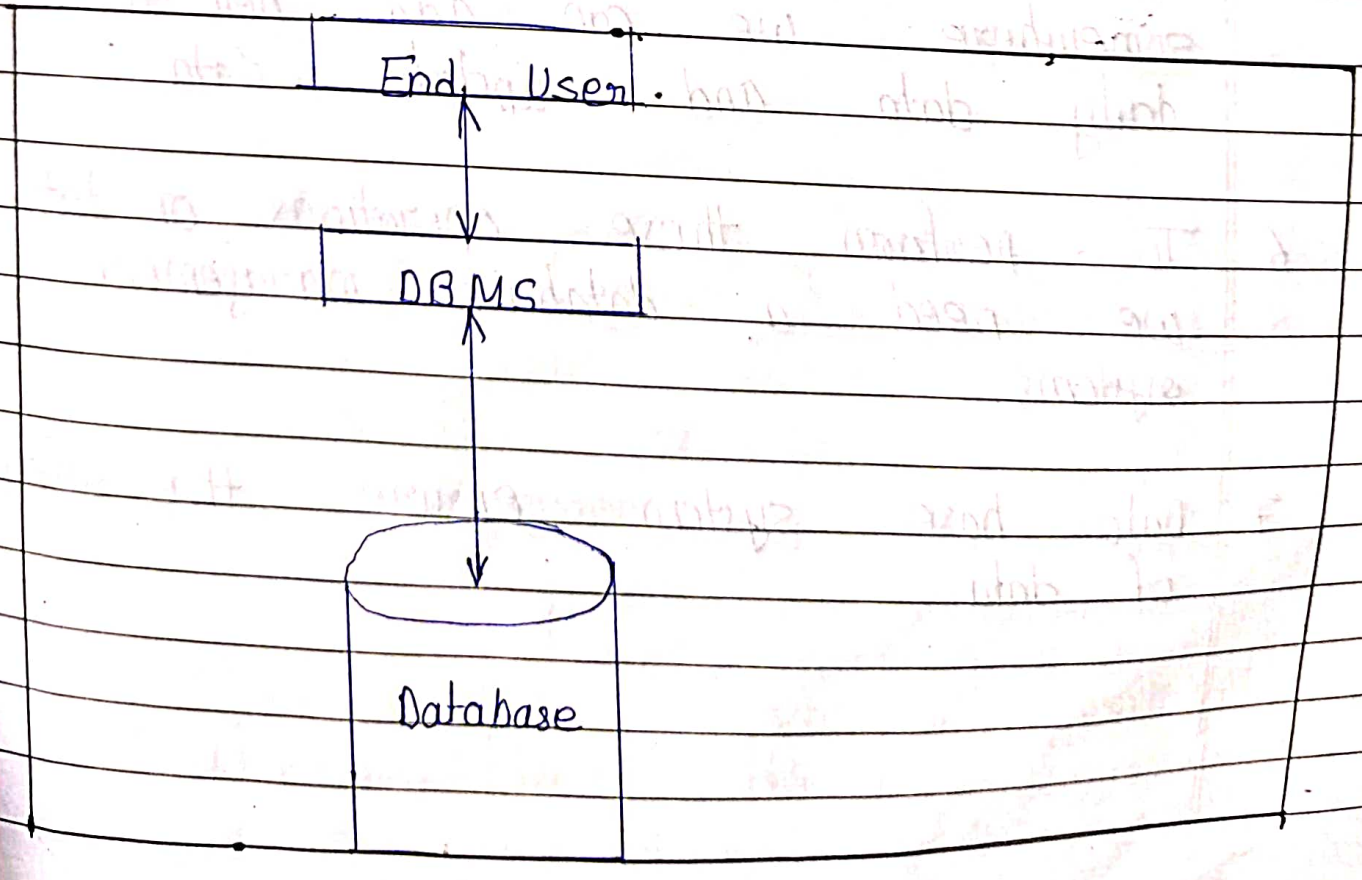
A database is a collection of information that is organized so that it can be easily accessed, managed and updated.

DBMS [database Management system] :

It is a software for creating and manging databases.

A DBMS makes it possible for end user to create, read, update and delete data in a database.

A DBMS acts as an interface b/w the database and end user or application program.



DBMS purpose :-

1. DBMS is a collection of programs that enables the user to create and maintain database.

2. Database system are design to manage Large amount of data.

3. The database system provide user facility of defining creating and manipulating the database.

4. The main purpose of database system to manage the data.

5. To manage large amount of data we need to store this data somewhere we can add new data, daily data and update data.

6. To perform these operations on data we need a database management systems.

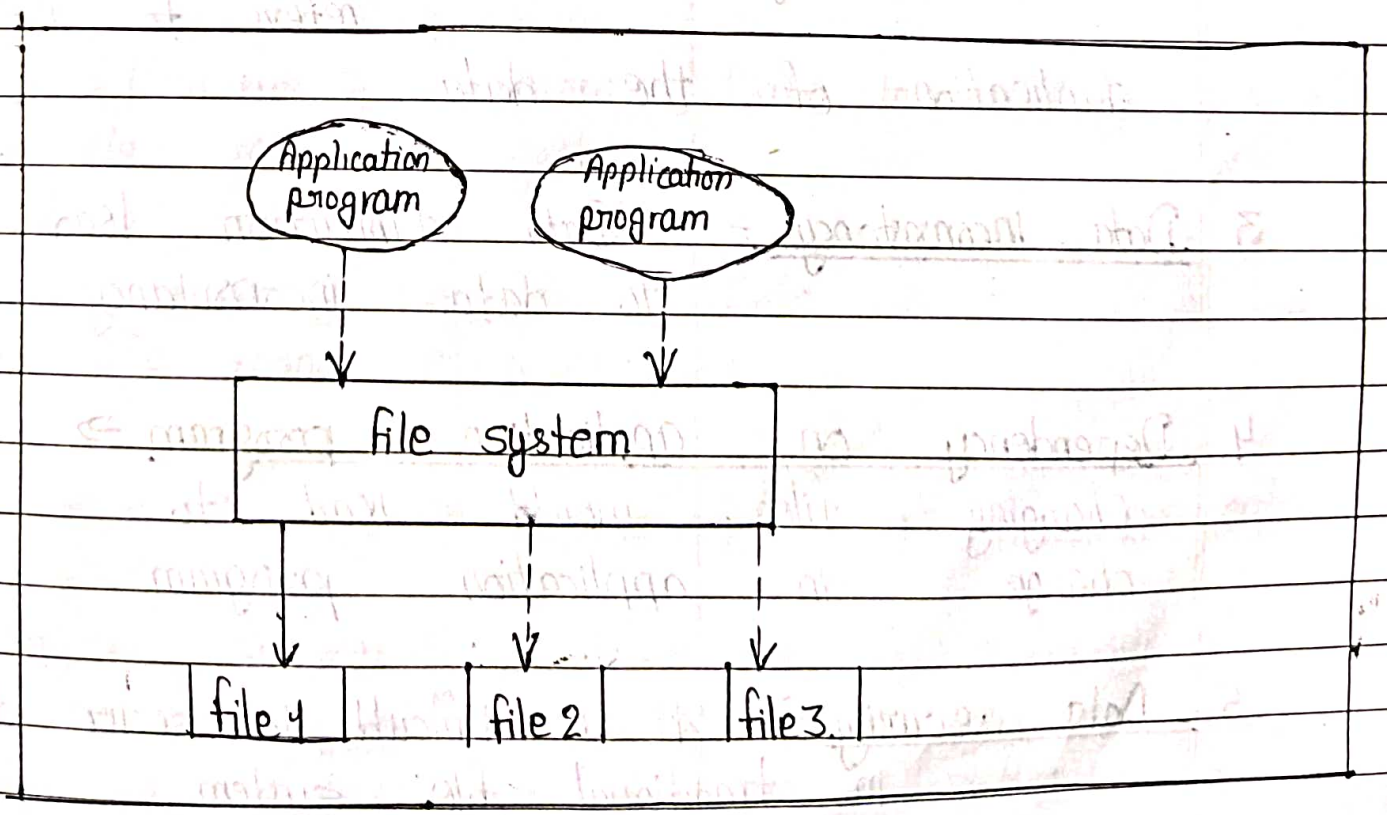
7. Data base system ensure the safety of data.

Traditional File Management System :-

Before the user of computer a manual file system was used to maintain the records and files.

In traditional approach information is stored in Flat file which are maintain by the file system. Application program go through file system to access these Flate files. In this traditional file system each file is independent of other files.

The data and application program that use the data are arranged in such a way that any change to data requires modification of all the programs that use the data.



Advantages of A traditional file systems :-

1. It is simple to use.
2. Less Complex.
3. It is cheaper.
4. It requires minimum investment.

Disadvantages or Drawback of traditional file :-

1. **Data isolation** :- Data is scattered in various file. Data is stored in diffrent file and each file is independent of each other.
2. **Data redundancy** :- [duplicity] :- Data redundancy refers to the duplication of the data.
3. **Data inconsistancy** :- Data duplication leads to data inconsistancy.
4. **Dependency on application program** :- Changing files would lead to change in application program.
5. **Data security** :- It is difficult to secure data in traditional file system.

6. Data Atomicity Issue :- Atomicity refers to all or nothing which means all the operations in a transaction execute or none.
Multiple user access is not supported in this approach.

7. Data program dependency.

File System Vs DBMS:

Difference b/w File system and DBMS :-

	File Management System	Database Management system
1.	Data redundancy is more	Data redundancy is less
2.	Data inconsistency is more in file Management system.	Data inconsistency is less
3.	Data security is low in file management system.	Data security is ~~less~~ high
4.	These are easy to implement.	Difficult to implement.
5.	It is simple	It is complex.
6	It is cheaper	It is costlier.

Advantage of DBMS over the file system :

1. No duplicacy or redundant data.
2. Data consistency and integrity.
3. Data Security.
4. Easy access to data.
5. Sharing of data.
6. Privacy of data.
7. Easy recovery of data.
8. Flexible ⇒ Database system are more flexible then file processing system.
9. Easy backup and recovery services.

Disadvantage of DBMS :

1. Complexity : Database systems are complex to understand.
2. DBMS software is costly.
3. Size of DBMS software is very large.
4. High hardware costly.
5. Higher programming cost.
6. Difficult to implement.

Diffrent type of database user :

1. Application Programmer : Application programm

write application programs that use the database. They interact with DBMS using DML [Data manipulation language] calls.

2. End User :- End user access the database from terminal end. They use the developed application and they don't have any knowledge about its designs and developing of database.

These users are of two types :-

(i) Casual User
(ii) Naive user.

(i) Casual User ⇒ These user have knowledge of query language. They access data by entering diffrent queries.

(ii) Naive User ⇒ Any user who does not have any knowledge about database. e.g ⇒ clerical staff in any bank. They don't have any dbms knowledge but they use database.

3. Database Administer (DBA)

DBA is a person responsible for design implementation maintain and repair of an organisation database.

4. System analyst :- It is responsible for design structure and properties

of database all the requirements of end user are managed by this user.

Role of DBA [Database Administrator] :-

1. Installation and configuration of database ⇒ DBA is responsible for installing and configure the database software.
2. Deciding the hardware device.
3. Defining storage structure and access method
4. Defining the schema ⇒ DBA define the schema which contains the structure of the data.
5. Database design ⇒ logical design of the database is designed by DBA.
6. Upgrading the DBMS server.
7. Design and implement the database.
8. Defing Backup and recovery procedures.
9. Tuning the database performance.

10. Migrate the database server.

11. Monitoring performance.

12. DBA decide the users of database.

13. Provides help and support to user.

14. Decide content of the database.

15. DBA is responsible for saving the data.

UNIT - 2

DATABASE SYSTEM CONCEPTS AND ARCHITECTURE

DBMS Schema : A design of a database is called the schema. It is the overall design of a database.
It is overall description of a database structure of a database.

Schema is of three types :

1. Physical Schema
2. logical Schema
3. View Schema

1. **Physical Schema** :- It is a design of a database at physical level is called physical schema. How data is a stored in block of storage is described by this schema.

2. **logical Schema** :- It is a design of database at logical level is called as logical schema.
What type of data is stored is defined using this schema.

3. **View Schema** :- It is a design of database at the view level is called as view schema. It describe and user intraction with database. View schema is

also known as sub schema database.

Instance :-

The data stored in a database at a particular moment of time.

- snapshot of the database at a particular moment.

Schema	Instance
1. Schema is overall description of the database.	It is collection of information stored in a database at a particular moment.
2. It do not change frequently.	It changes frequently.
3. Schema is same for whole database.	Data in instance can be change using addition, deletion, updation.

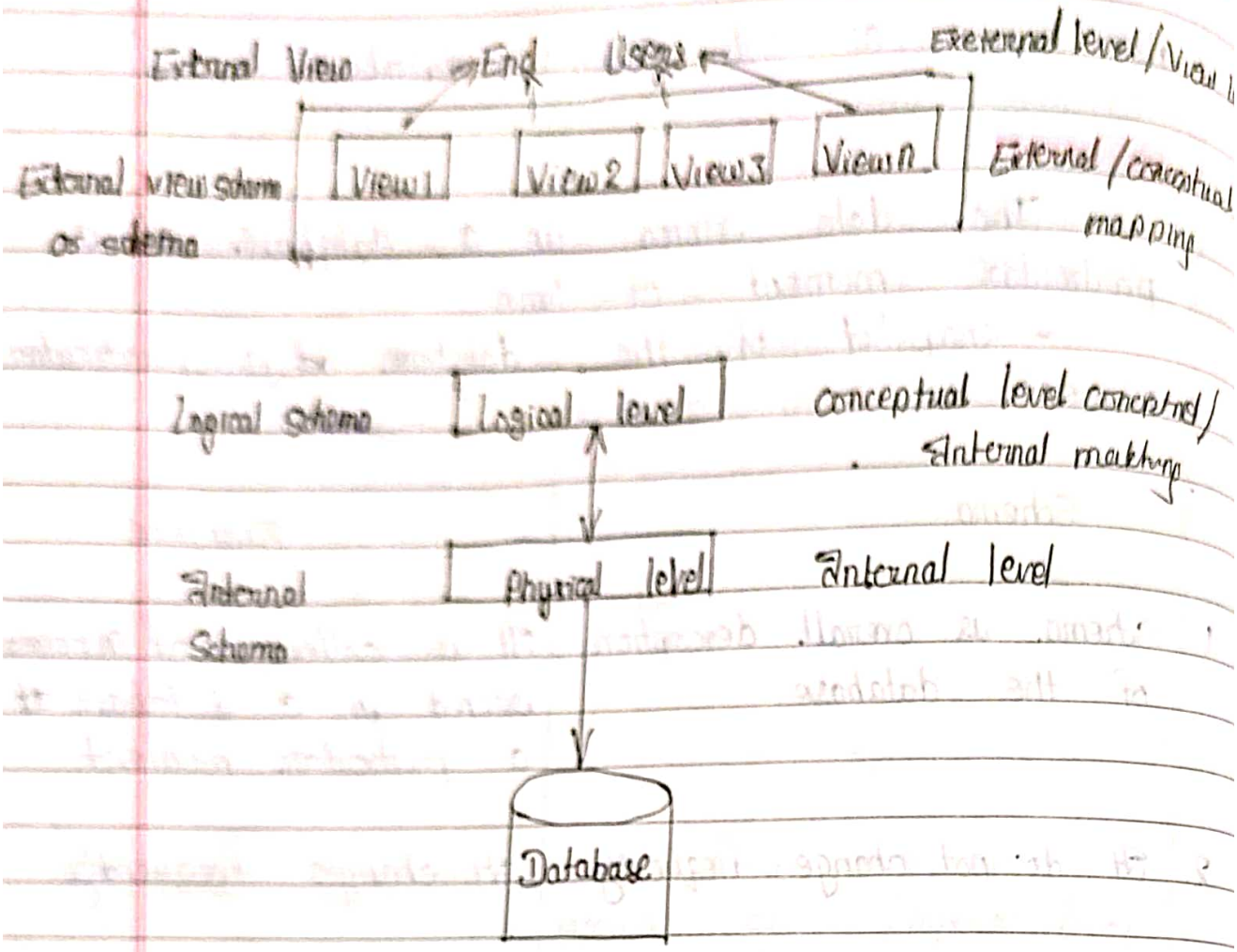
ANSI / SPARC - [Architecture of a database system]

ANSI - American National Standards Institution.

SPARC - Standards Planning and requirements Committee

3-level Architecture of a database

3- Schema Architecture of a database.



1. Physical / Internal Level :- Internal level is also known as physical level. Internal level has internal schema which describe the physical storage structure use of the data base internal schema is also known as physical schema. It uses physical data level.

It is use to define how the data this actualy store on this store device. This level is hidden from the programmer.

This is the lower level of the architecture. This level is responsible for alocating space

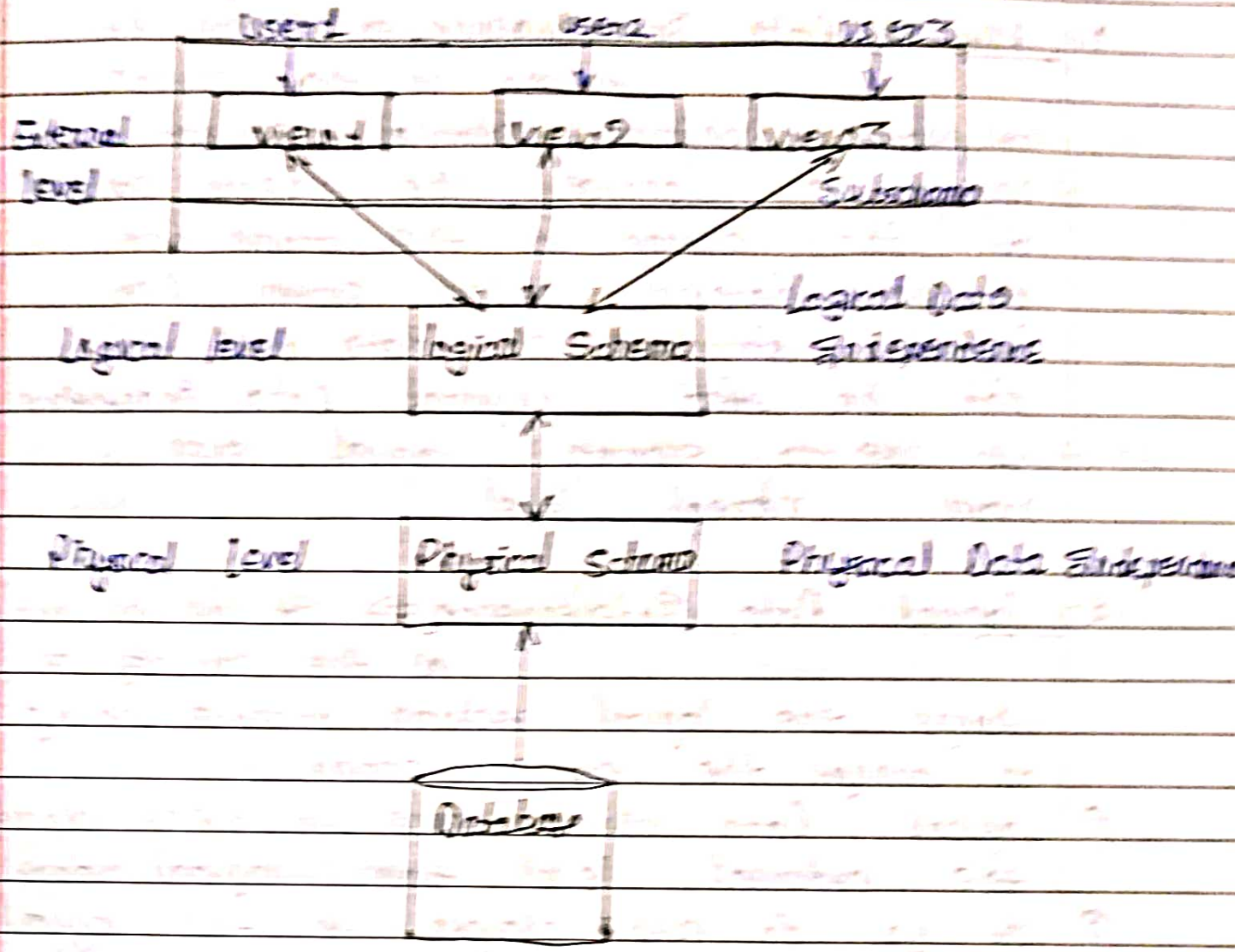
to the data. how the data is store.

2. Logical level / Conceptual level.
Logical level :- Conceptual level or logical level user logical schema is also known as conceptual schema.
Conceptual level describe what data are to be stored in the database. At also describe what relationship exist among these data.
Conceptual schema describe the structure of whole database. This level hide the detail Physical storage structure.
programer and DBA work at display.

3. External level :- At is also known as view level or user level at the external level. A data base content many schema that are called external schema or view schema.
(1) View schema is also known as subschema describe the diffrent view of the database. At describe the part of database that a user is interted and hide the re of database and from that user.
(2) External level is the top level of three of level DBMS Architecture.

3. Mappings :- The process to convert a request and the result from one level to another level is known as mapping.

Data Independence



Data Independence ⇒ Data independence is define as a property of DBMS that helps you to change the database schema at one level of a database system. Without require to change schema at next higher level. Data independence help you to keep data separated from all programs that make use of it.

Type of Data Independence :-

1> Physical Data Independence

2> Logical Data Independence

1> Physical Data Andependence ⇒ It can be defined as the capacity to change physical schema without having to change the logical schema If the change in the storage size of the database system server then logical structure of data base will not be effect. Physical Data Andependence is use to separate logical level from internal level.

2> Logical Data Andependence ⇒ It can be define as the capacity to change the logical schema without having to change the external schema.

① Logical Data Andependence is use to seprate the external level from logical data.

② If you do any changes in the logical view of the then user view of the data good not be effect.

25/03

	Logical Data Andependence	Physical Data Independ
1.	It is concerned with the structure or changing the data defination.	It is concerned with the storage of data.
2.	It is difficult as retreving the of data is dependent on logical structure of the data.	It is easy to retreive.

3	It is difficult to acheive logical data independence	It is easy to acheive
4	You need to make change in the application program if new fields ~~or data~~ are added or deleted from the database.	A change in physical level does not require to change application program.
5	It is related to logical schema. e.g.:	It is related to internal schema. e.g.
6.	add, modify, or delete ~~are~~ and ~~the~~ an attribute.	Change in compression technique or storage devices, etc.

DATA MODELS :-

Data model is a model that defines in which format the data are represented and accessed. Data model are used to show how the data is connected and stored inside a system. Data model represent the relationship b/w data. Data model define the logical design and structure of a database. and defines how data will be stored accessed an updated in a database management system.

Types of data Model :-

(i) Record Based data Model
(ii) Object Based data Model.

(i) Record Based data model :-

When the database is organized in some format of records then the model is called record based data model.
The three most popular record based data model are :-

(a) Relation Data Model
(b) Network Data Model
(c) Hierarchical Data Model

(a) Relational Data Model :- It represents data as relations or tables.

Relational data Model uses table to represent the data and the relationship among these data. Tables are known as Relations in relational model. Each table has multiple column and each column has a unique name. All the information related to a particular type is stored in rows on the tables.

Advantages of Relational data Model :

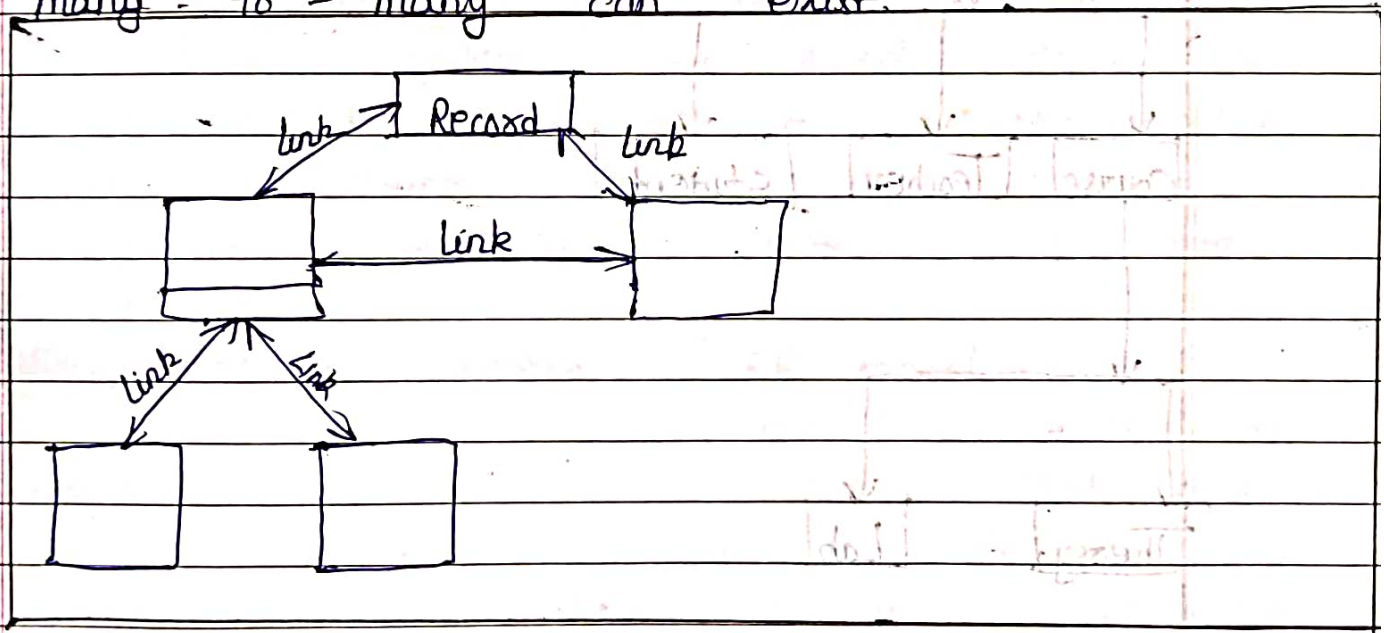
1. It is the simplest model.

2. It is easy to design, implement maintain & use the database using this model.

(b) Network data Model :- It represent the data as record type and one to many relationship. In this model data are represented by collection of records and relationship among data is represented by links. In this data model graph data structure is used. It allow a record to have more then one parent.

Advantage of Network data Model :-

1. This model is simple & easy to design. In this model relationship like one to one and many - to - many can exist.

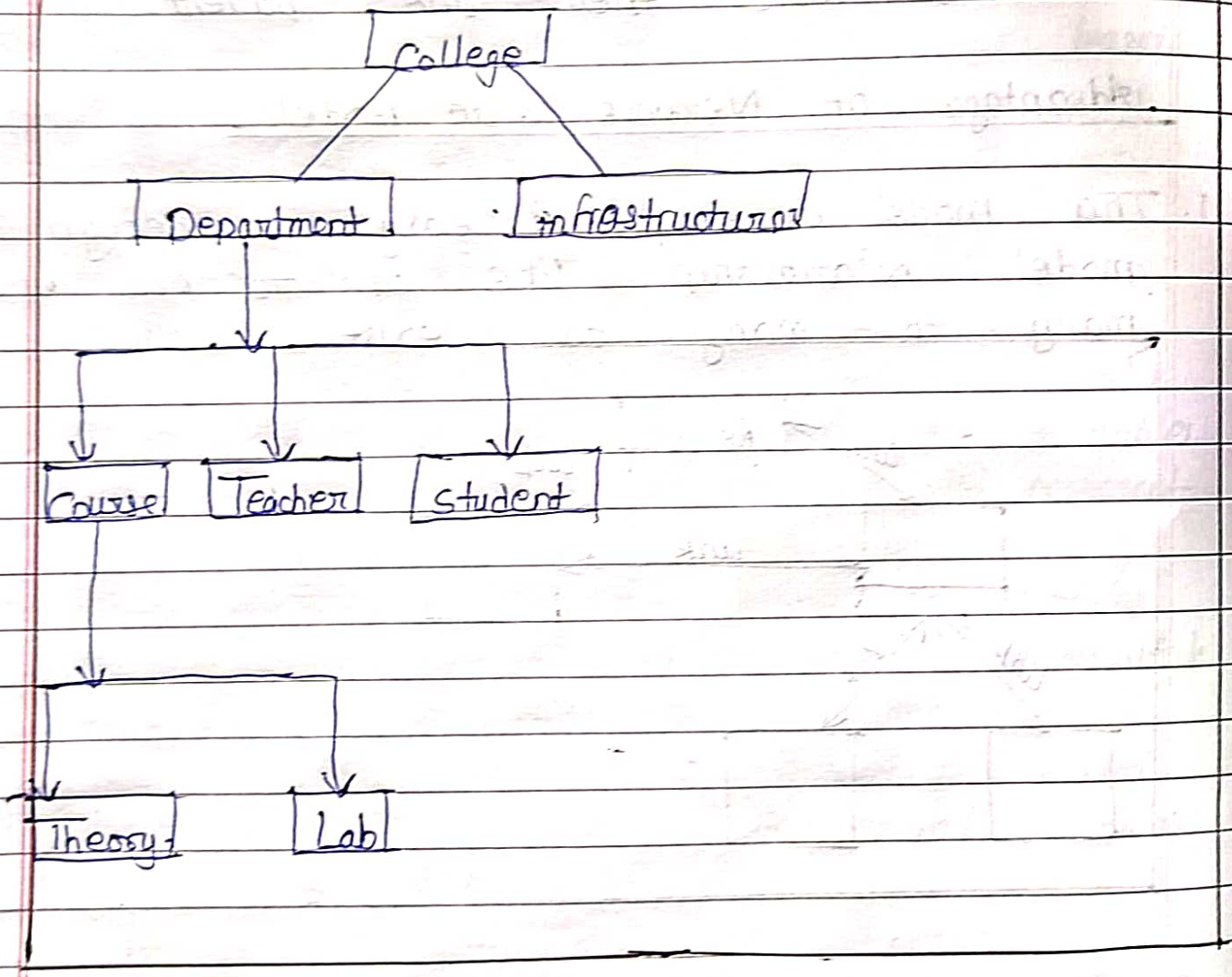


2. It replace the hierarchical tree with a graph.

3. This model is easy to understand

(c) Hierarchical Data Model :- In this, data is represented by collection

of records and relationship among data is represented using link. Here pre-data structure is used. This model is based on parent child relationship. This model is the first DBMS model. It uses a pointer to navigate b/w the data.



This model organize data into a tree like structure. In this model a child nodes will only have a single parent node.

26/03

(11) Object based data Model :-

An object based data model the database is organized in real world objects. A no. of fields or attributes are define in each object. Most popular object based data model are :-

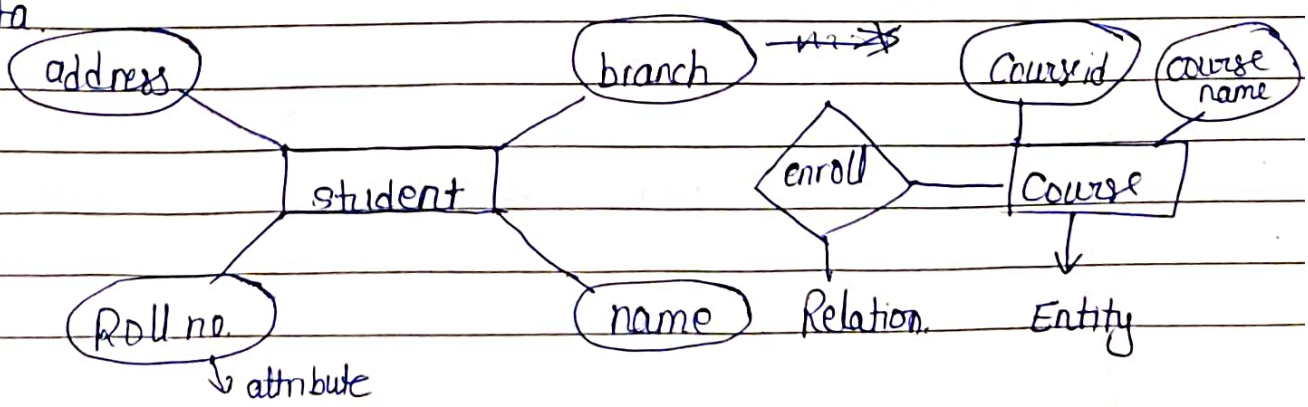
① object oriented model

② Entity relational model [ER model]

① **object oriented Model** :- This model is based on collection of objects. This model stores the data in the forms of object, classes, and inheritance. This model is used in file mangement system. This model represent real world objects and attribute. Object oriented database system handles more complex application such as scientific experiments.

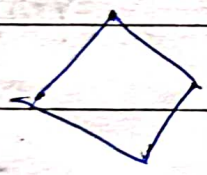
② **Entity Relational Model** [ER model] :-

This model is based on perception of the real world that consists of a collection of basic objects and relationship b/w them. It is a high-level data model. This model is very simple and easy to design logical view of data.



 – attribute

 – entity

 – Relationship

UNIT – 3

Relational Model

Relational Model in DBMS :-

Relational Model represents how data is stored in relational Databases.
A relational database stores data in the form of relations (tables)
A relational model can be represented as a table of rows and columns.
Relational Model (RM) represents the database as a collection of relations.
A relation is a table of value.
Every row in the table represents a collection of related data values.
The data are represented as a set of relations.
In the relational model, data are stored as tables.
Each table is a group of column and rows where column represents attribute of an entity and rows represents records.
In relational model, the data and relationships are represented by collection of inter-related tables.

A relationship database has following major components :

1. Table
2. Record or Tuple
3. Field or column name or attribute
4. Domain
5. Instance
6. Schema
7. keys

Relational Model Concept:

1> **Relation:** A relation in a relation database is describe as a table.
A table in relational model is know as relation.

2> **Table:** Table is group of column and rows and each table as a name in database.

3> **Record or Tuple:** A rows in a table represent the record of a relation and is known as a tuple of a relation.

4> **Attribute:** A column header of a table is known as attribute of a relation. Attribute are the properties that define a relation
eg: roll no, Name, branch.

30/03 5: **Domain:** A set of possible values for a given attribute is known as domain.
A domain is atomic it means values are

indivisible

A domain is a set of predefined or permitted values for an attribute in a table.

6 Degree of a relation : A no. of the attributes in a relation is known as degree

7. Cardinality : A ~~set~~ no. of tuples of a relation or the no. of tuples in a relation is called as cardinality.

8 Relation instance : The set of tuples of a relation at a particular instance of time. is called as relation instance.

RELATIONAL schema : A named relation defined by a set of attributes and domain name pair is called as relational schema

Keys :

Key is a single attribute or combination of two or more attributes of a relation which is used to identify tuples of the relation.

Super key :

Super key is an attribute of set of attributes that uniquely identifies a tuple in a

relation.

4) **Candidate key** :- Candidat key is a set of attributes that uniquely identify tuples in a table.
Candidate key is a super key with no repeated attribute. Every table must have atleast a single candidate key. A table can have multiple candidate key but only a single primary key.

Candidate key

Rollno.	Name	Phono.	Emailid	Branch
1	Abc	7893	a@gmail.com	IT
2	XYZ	12345	b@gmail.com	CSE
3	PQR	98765	c@gmail.com	ME

Candidate key is minimal Super key.

Super key Example :-

{Rollno}
{Rollno, Name}
{Rollno, Phoneno}
{Rollno, Email id}
{Rollno, Branch}
{Rollno, Name, Phone No.}
{Rollno, Email ID, Branch}

{PhoneNO}
{Emailid}
{Phone_No, name}
{Phoneno, Emailid}
{Phoeno, Branch}
{Emaidid, Name}
{Emailid, Branch}
{RollNo, Name, PhoneNo, Emailid, Branch}
{RollNO, Name, Branch}

Primary key : Primary key is an attribute or group of attributes that uniquely identify a tuple in the table. Primary key is selected from one of the candidate key. Primary key cannot be duplicate and it can't be null.

Alternate key : A table can have multiple candidate keys. Among these candidate keys only one key is selected as primary key and the remaining keys are known as alternate keys.

Foreign key : Foreign key are the columns of a table that points to the primary key of another table. It is a column or attribute that creates a relationship between two tables. Foreign key is useful in linking two tables together.

primary key

Foreign key

primary key

RollNo	Name	Phoneno	Emailid	Course id	CourseName	RollNo
1	Abc	123	a@gmail.com	C01	diploma	1
2	def	345	b@gmail.com	C02	degree	2
3	ghi	678	c@gmail.com	C03	IT1	3

Constraints in Database :-

Constraints are used to specify rules for data in a table.
Constraints are used to limit the type of data that can be inserted in a table.
Constraints can be column label or table label.
Column lebel constraints apply to a column and table label constraints apply to the table
Constraint ensure the correctness of the data in the database
Relational constraints are the restrictions imposed on the database content or the object

Types of Constraints in Database :-

- Domain Constraint
- Tuple Uniqueness constraint / Unique key Constraint
- Key Constraint
- Not Null Constraint
- Integrity constraint
  - Entity Integrity Constraints
  - Referential Integrity Constraints

1. Domain Constraint :- It defines the domain or set of values for an attribute. It specify that value taken by the attribute must be atomic value from its domain

Example :-

Name	Age
ABC	19
DEF	20
GHI	A

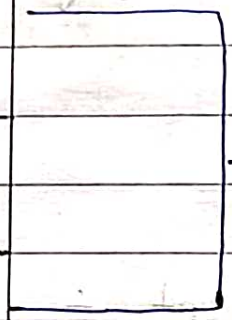
A → Not allowed since only integer value is allowed here.

2. Tuple Uniqueness Constraint :- This Constraint specifies that all the tuples must be unique in any relation. Duplicate

record of tuple are not allowed in a relation.
Unique key constraint duplicate values are not allowed in a relation.

Example:-

Name	Age
Abc	19
XYZ	20
Abc	19



3 Key Constraint :- It specifies that in any relation all the values of primary key must be unique and primary key cannot have null values.

Example:-

Roll no	Name	Branch
1	XYZ	IT
2	ABC	CSE
3	PQR	ME

4. Not Null Constraint :- This constraint specify

that an attribute cannot have null values.

04/08/25 Integrity Constraints :-

(i) Entity Integrity Constraints :- The Entity Integrity constraints states that primary key cannot be null. This is because primary key value is used to identify tuples or a records in a relation. If primary key is null then we cannot identify those records.

Example :-

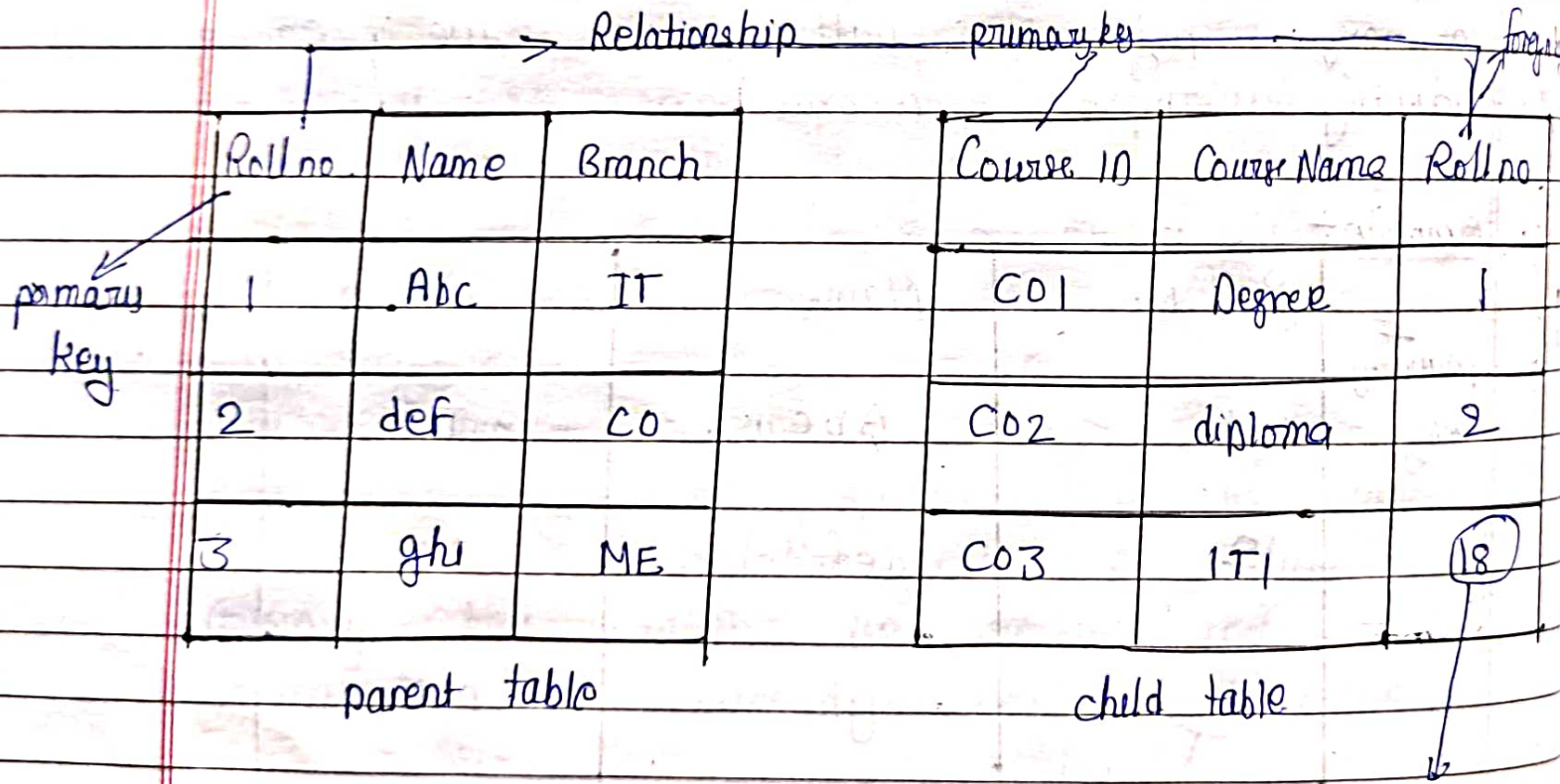
Roll No (primary key)	Name	Branch
1	Abc	IT
2	def	CO
	ghi	ME

Not allowed
prim. key cannot be null

(ii) Referential Integrity Constraint :- This constraint is specified b/w two tables. It is also known as foreign key constraint. This constraint states that if a foreign key exist in a relation either the foreign key value must match the primary key value of some tuples in its

parent table or foreign key value must be null.

Foreign key value must be equal to the primary key value of parent table or it should be null. This constraint ensure that their must exist a valid relationship b/w two tables.



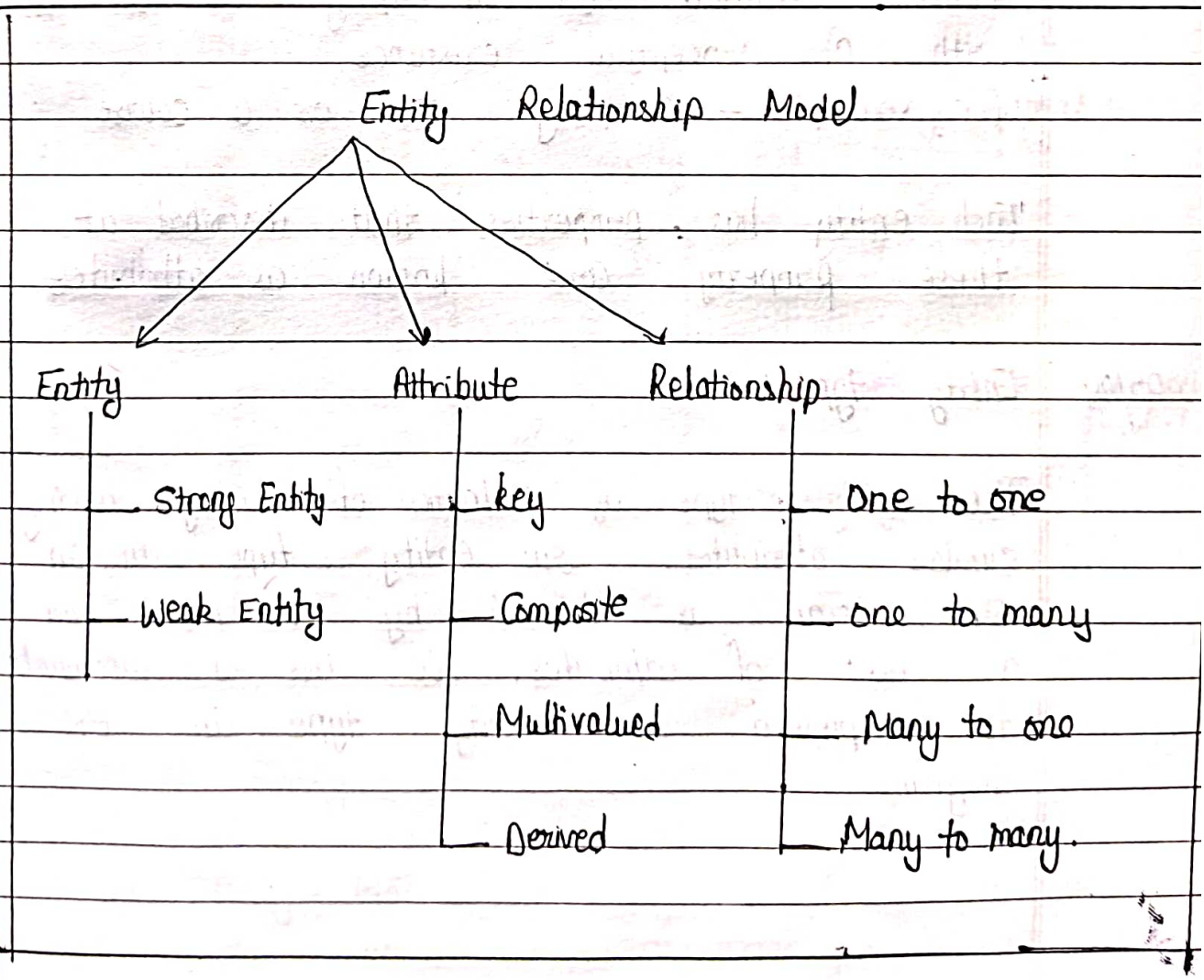
UNIT-4

ENTITY RELATIONSHIP MODEL [E.R Model]

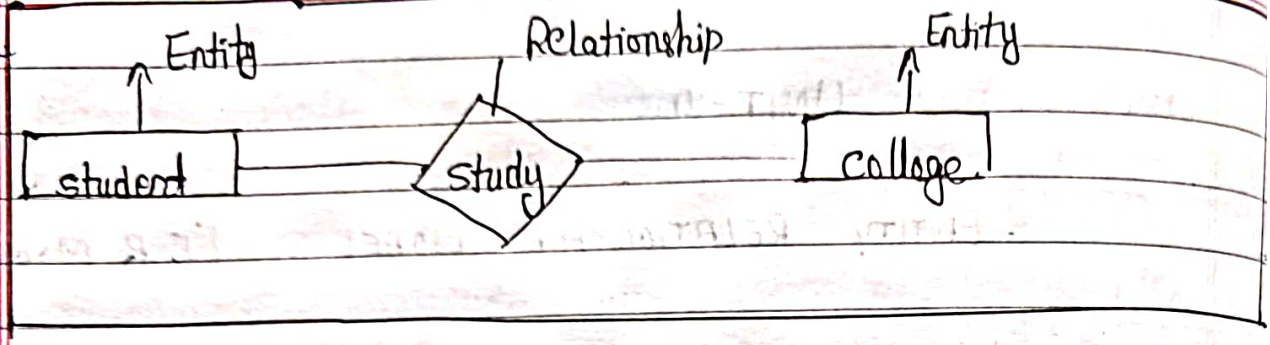
ER model describe the structure of database with the help of a diagram which is known as entity relationship diagram [ER diagram].

* Main Component of ER model are entity set and relationship set.

ER model is a design of a database. This model is used to define the data elements and relationship for a system.



Example:



✳ **Entity** :- An entity is an object, place or person, or event that the system store information about
It is a thing or object which can be distinctly identified
An entity may be an object with a physical existance such as a person, car, house, student or it may be an object with a conceptual existance.
For example - company, Job or a course.

Each entity has properties that describes it these property are known as attributes.

05/04 **Entity type** :-

The entity type is collection of entity having similar attributes. An entity type in an ER diagram is defined by a name and a set of attributes. We use a rectangle to represent an entity type in ER diagram.

Example :-

student → Entity type

Rollno.	Name	Branch	
1	A	IT	→ entity
2	B	CO	→ entity
3	C	ME	→ entity

Student
Entity type

Entity Set :-

Entity set is a collection of entities of the same entity type.

* The collection of all entities of a particular entity type in database at any point in time is called as entity set.

student → Entity type

Rollno.	Name	Branch	
1	A	IT	→ Entity [E1]
2	B	CO	→ Entity [E2]
3	C	ME	→ Entity [E3]

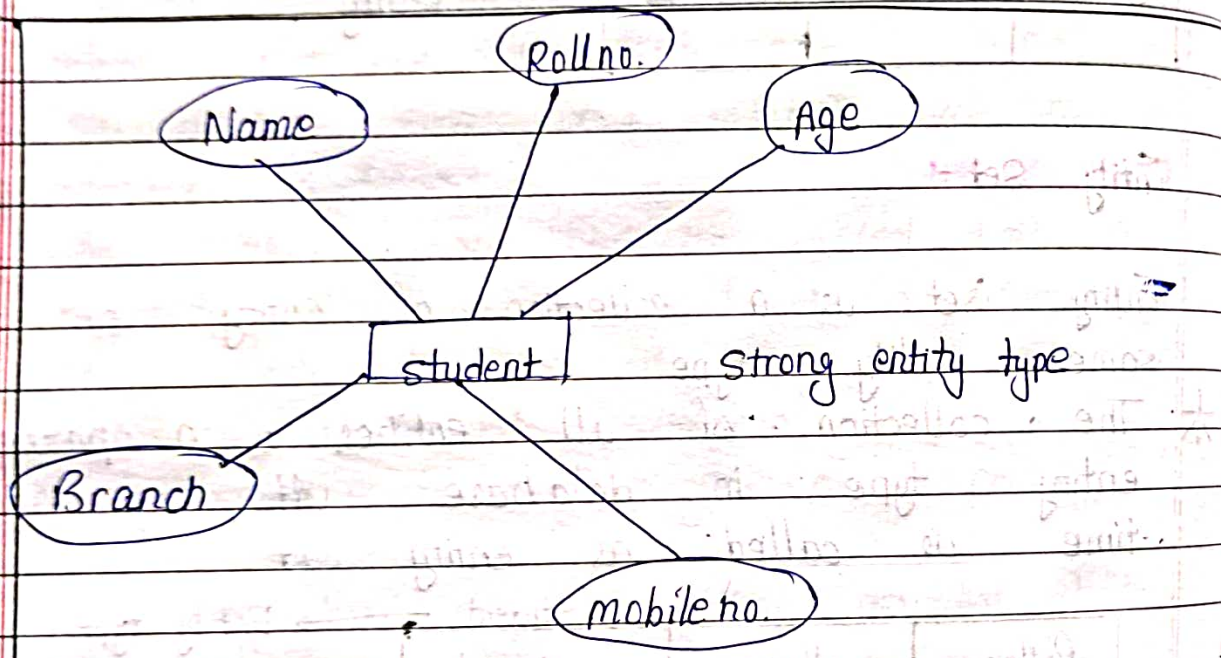
Entity set

E1
E2
E3

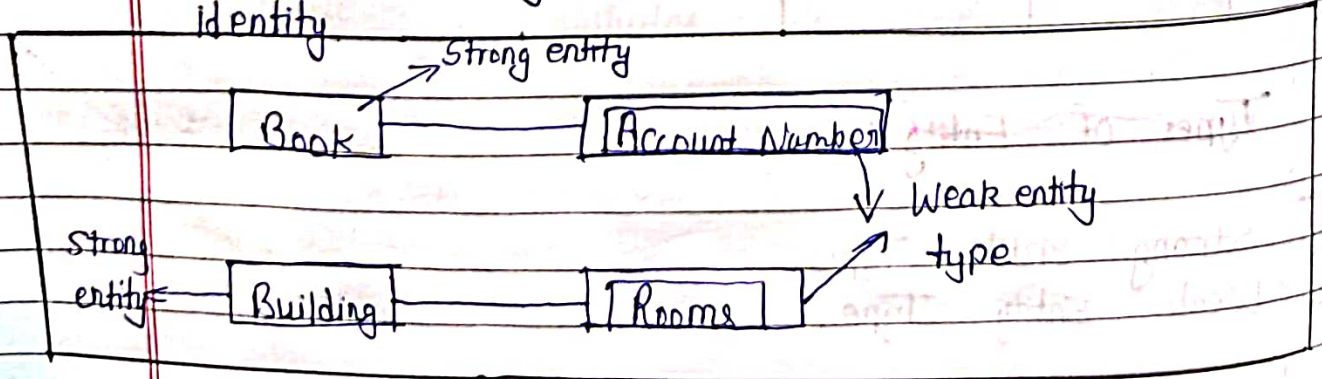
Types of Entity :-

1. Strong Entity Type
2. Weak Entity Type.

(i) Strong Entity type :- Strong entity types are those entity types which has a key attribute. This key attribute is primary key and is used to identify each entity uniquely. Strong entity type is represented by a rectangle.

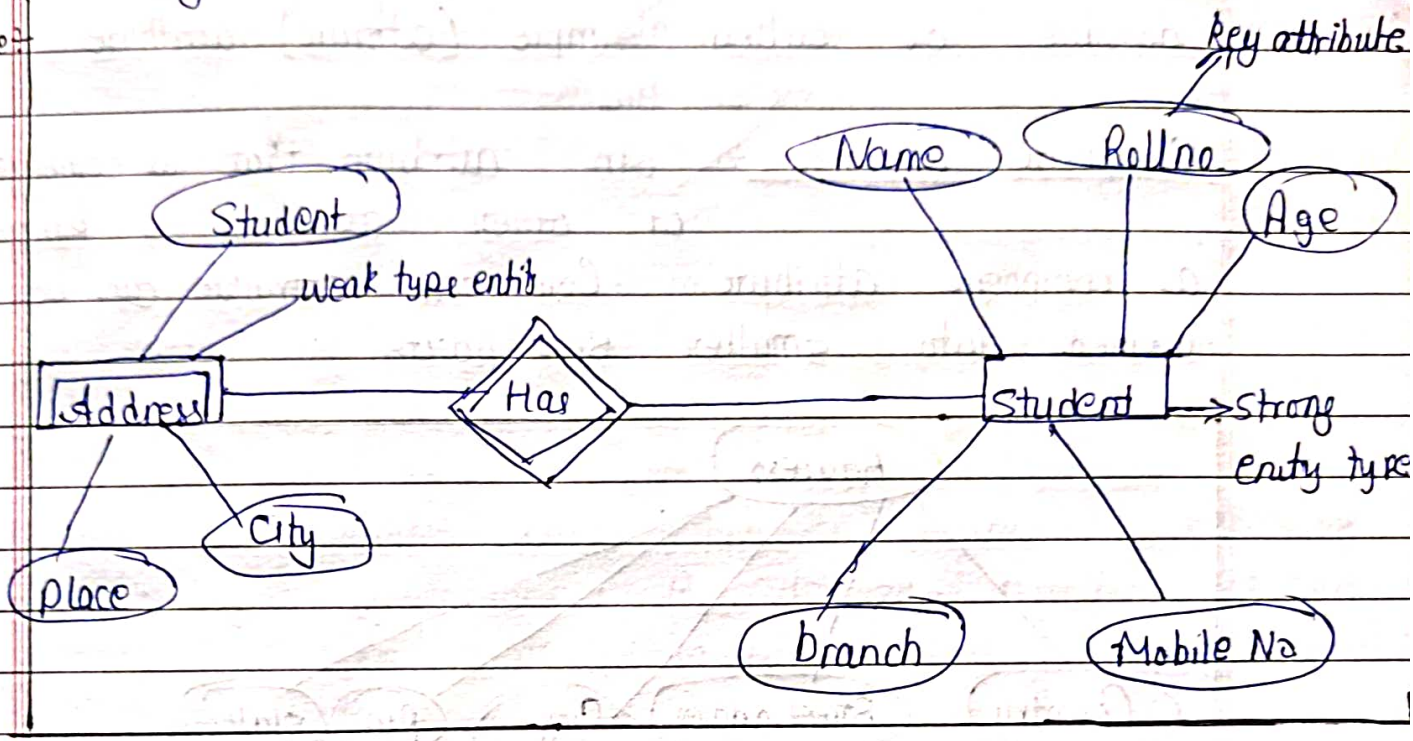


(ii) Weak entity Type :- Weak entity type does not have a key attribute of their own. Weak entity type depends upon some other strong entity for its distinct identity.



A weak entity is represented by a double outlined rectangle.

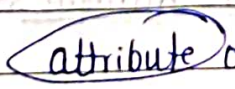
Example :-



Attributes :-

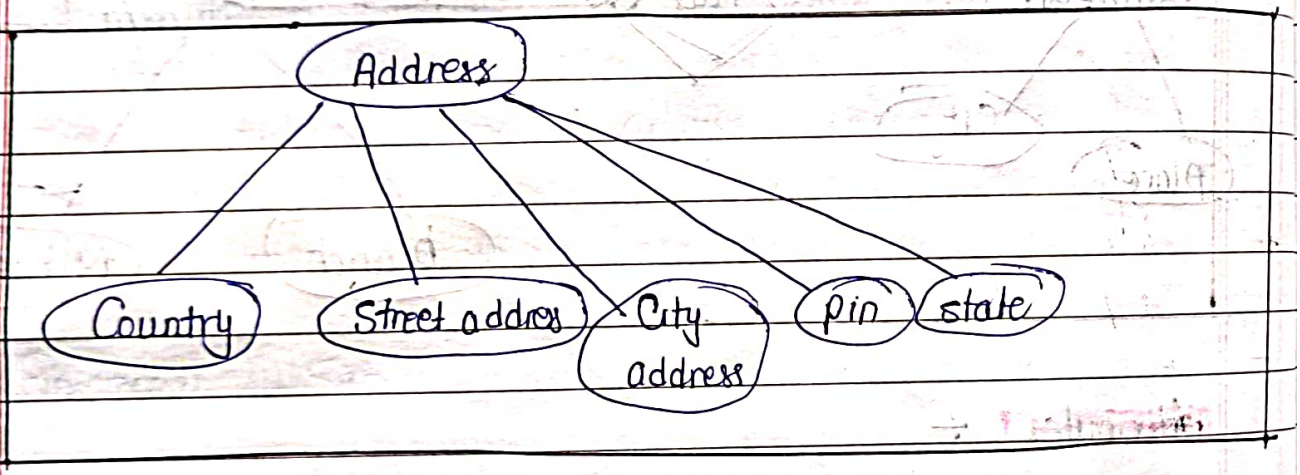
Attributes are the properties which define the entity type. An attribute is represented as oval shape in an ER diagram There are several types of attributes in ER model :

(i) **Key attribute :-** A key attribute can uniquely identify an entity from an entity set. Key attribute is represented ~~like~~ by oval, text of key attribut is underlined

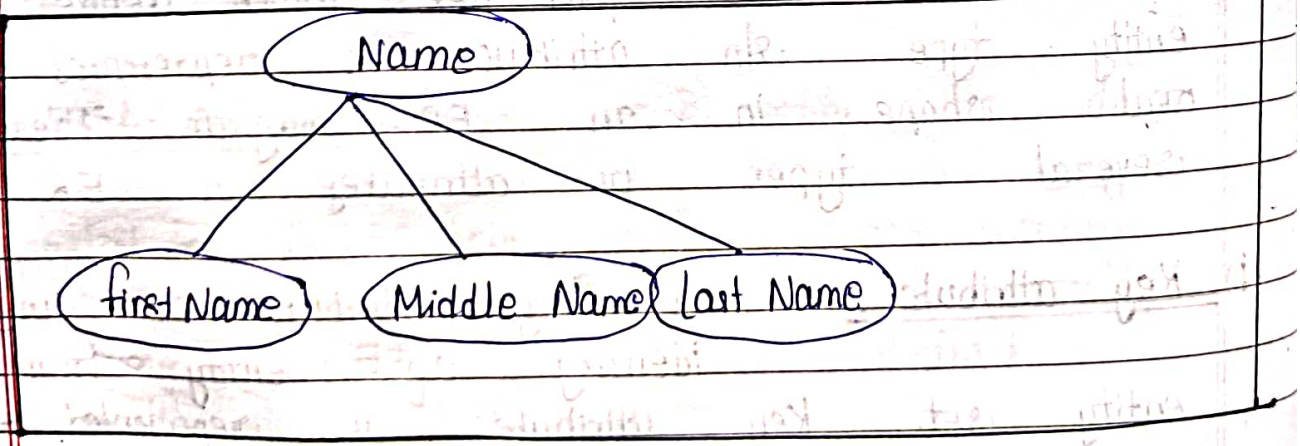


(ii) Simple (atomic) Vs Composite attribute :- Attributes that are not divisible or called simple (atomic) attribute.

Composite attribute :- An attribute that is combination of other attributes is known as composite attribute. Composite attribute can be divided into smaller sub parts.



or



06/04/(iii) Simple Valued Vs Multivalued Attribute :-

Simple Valued attribute :-

Attributes having single value for a particular entity is known as single valued attribute.

for example - age.

Multivalued attribute - An attribute that can have multiple values is known as multivalued attribute. It is represented by double oval in an ER diagram.
for example - phone number.

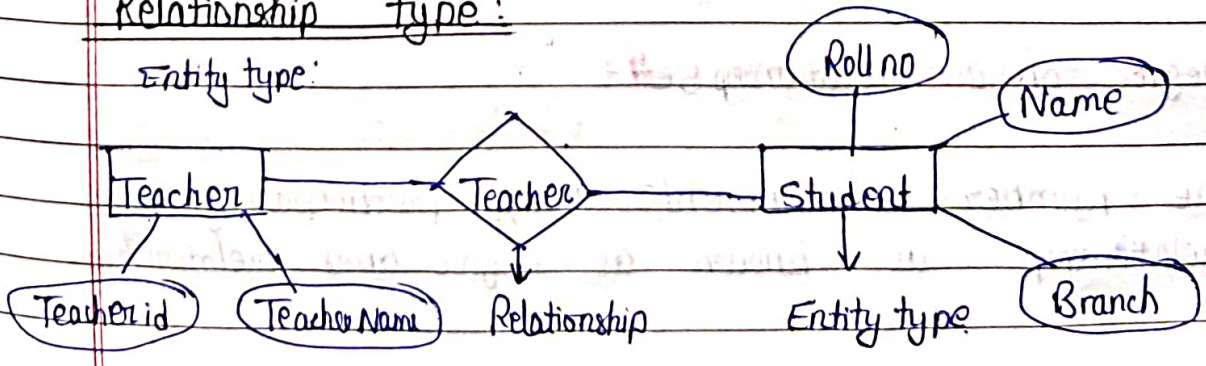
(iv) Base [Stored] Vs Derived Attribute :-

In certain cases two or more attributes values are related. A derived attribute is one whose value is dynamic and derived from another attribute.
For example - A person's age is a derived attribute as it changes over time and can be derived from another attribute. e.g :- d.o.b.
Here d.o.B attribute is called as Base or stored attribute.
Derived attribute is represented by dashed oval.

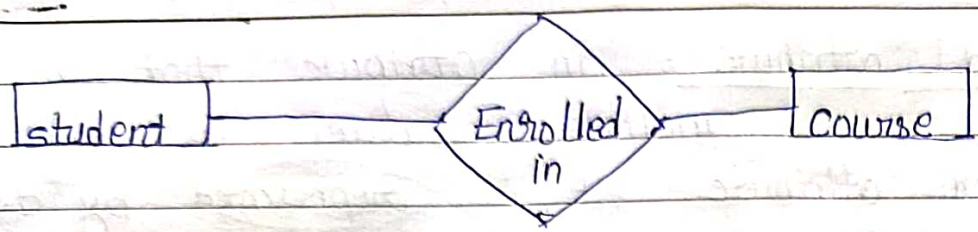
Relationship

A relationship is association among many entities.

Relationship type :



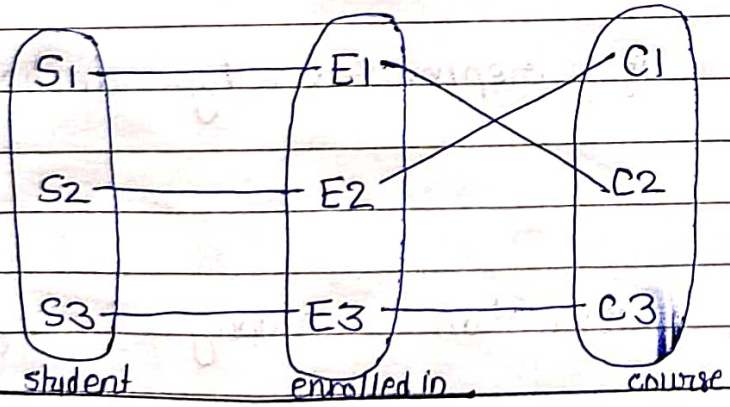
Relationship Type :



A relationship type represent the association b/w entity types e.g. enrolled in is a relationship type that exist b/w entity type student and course. Relationship type is represented by a diamond and connecting the entities with lines.

Relationship Set :

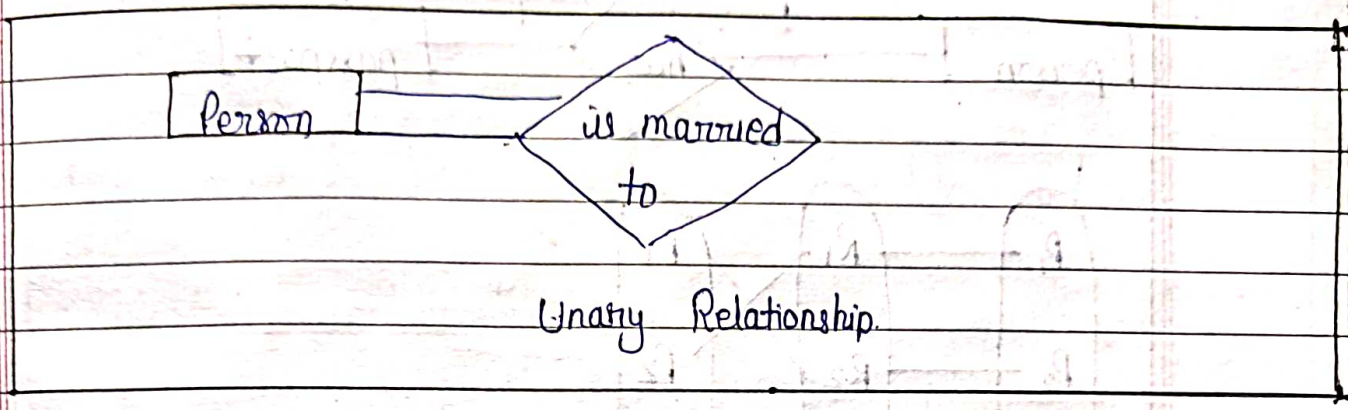
A set of relationship of same type is known as relationship set



Degree of a relationship set :

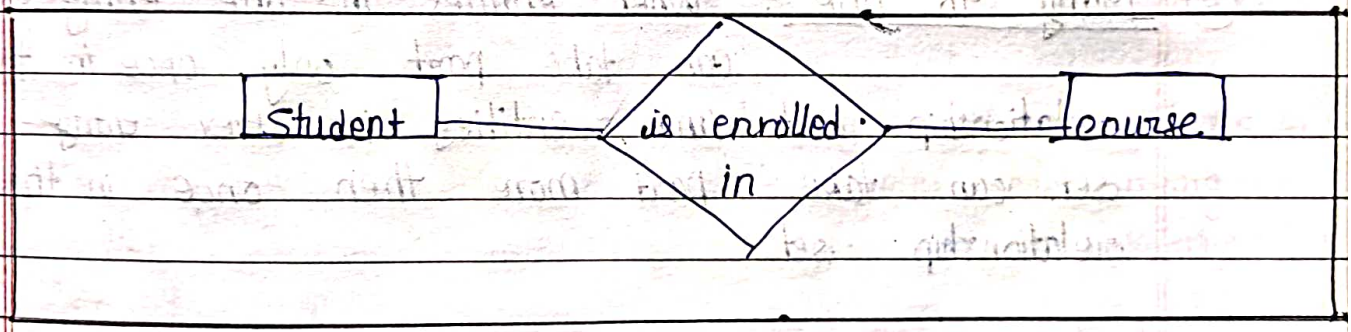
The number of diffrent entity participating in a relationship is known as degree of a relationship.

1. Unary Relationship : When there is only one entity participating in a relationship then this is known as unary relationship.



Unary Relationship.

2. Binary Relationship: When there are two entities participating in a relation then this is called binary relationship.



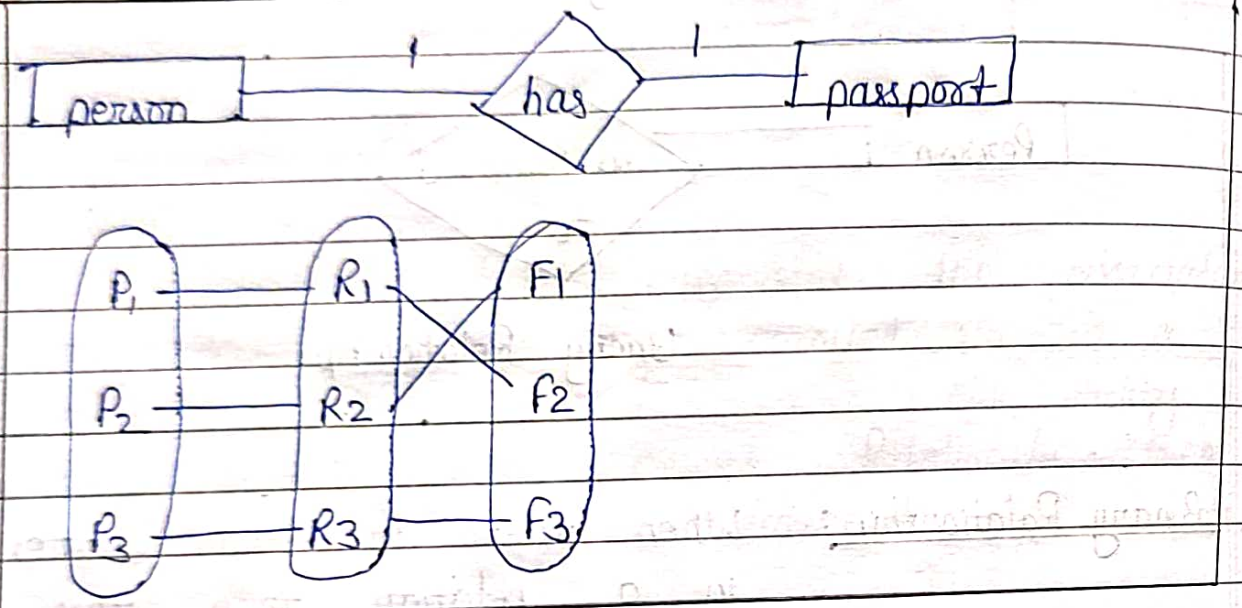
07/04 Cardinality :

The number of times an entity of an entity set participate in a relationship set is known cardinality.

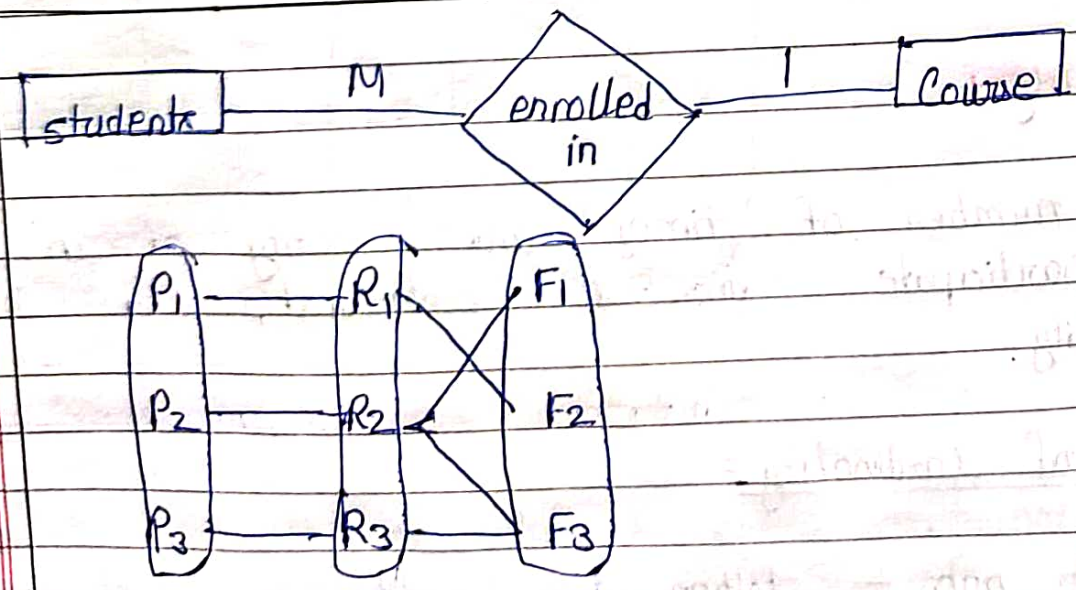
Types of Cardinality :

① One to one : When each entity in each entity

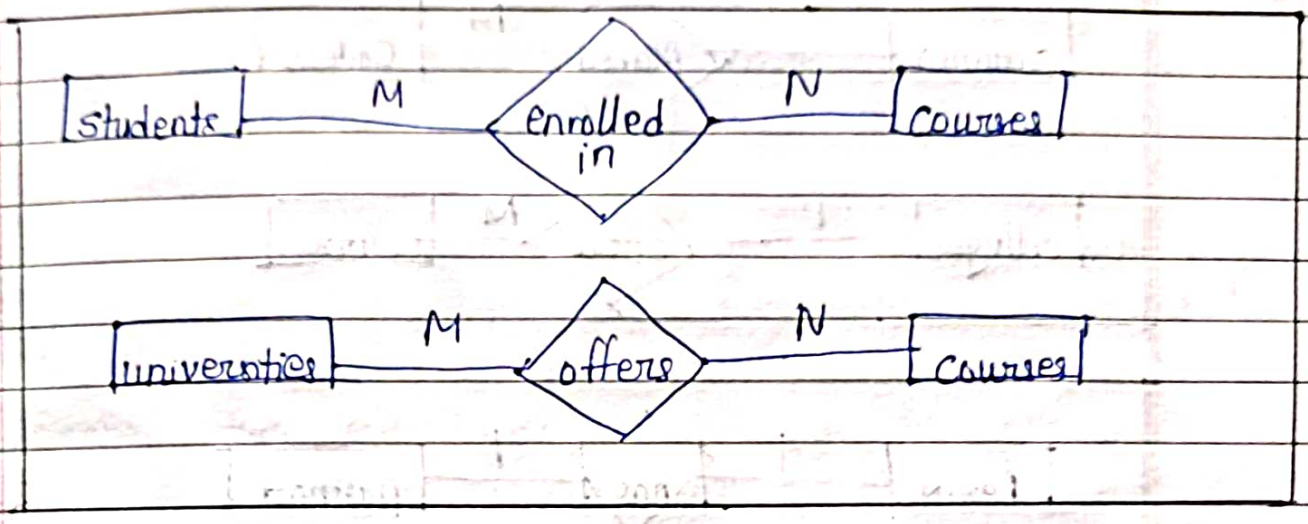
set can take part only once in the relationship. The cardinality is one to one.



(2) Many to one :- When entities in one entity set can take part only once in the relationship set and entities in other entity set can take part more then once in the relationship set.



③ Many to Many :- When entities in all entity set can take path more then once in the relationship. then cardinality is many to many.



Types of Relationship :-

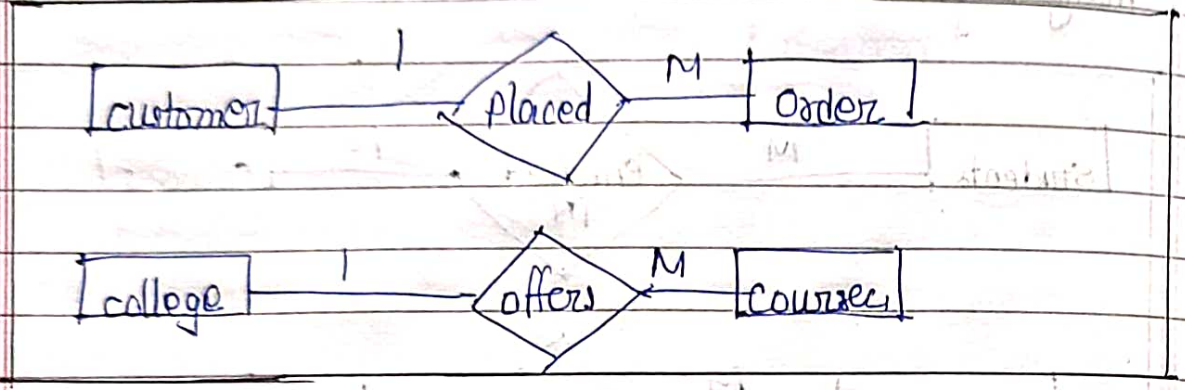
A relationship is represented by diamond shape in ER diagram. It shows the relationship among entities. There are four types of relationship :

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

① One to one relationship :- When a single instance of an entity is associated with a single instance of another entity then it is called as one to one relationship.

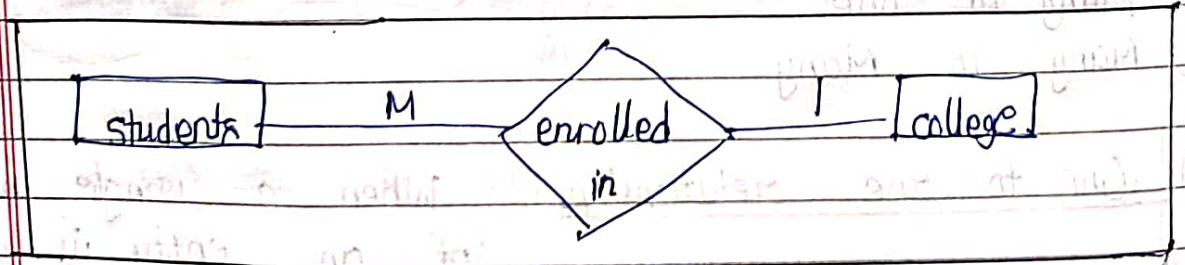
② One to many relationship :- When a single instance of

An entity is associated is associated with more then one instance of another entity then this relationship is called as one to many.



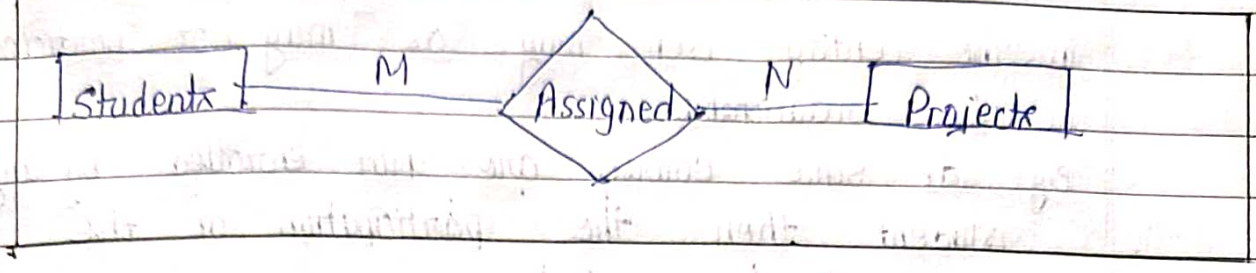
Example of one to one.

3) Many to one relationship :- When more then one instance of an entity is associated with a single instance of another entity then it is called as many to one relationship.



4) Many to many relationship :- When more then one instance of an entity is associated with more then one instance of another entity then

it is called many to many relationship.

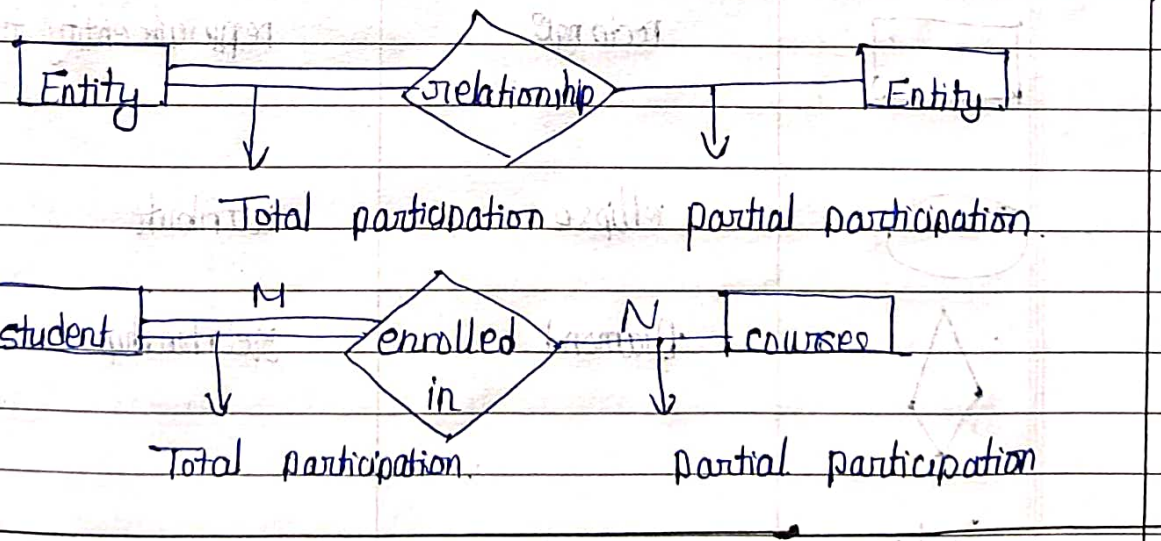


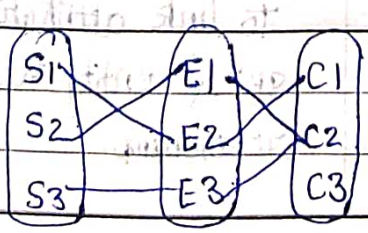
Participation Constraint :-

There are two types of participation constraint :

① Total Participation
② Partial Participation

① Total participation :- Each entity is ~~enrolled in~~ involved in the relationship. Each entity in the entity set must participate in the relationship. Total participation is represented using double lines.



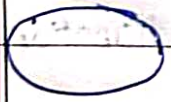
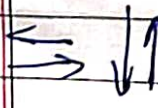


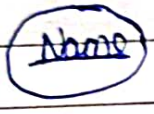
(2) Partial Participation :- Not all entities are involved in the relationship. the entities in the entity set may or may not participate in the relationship.

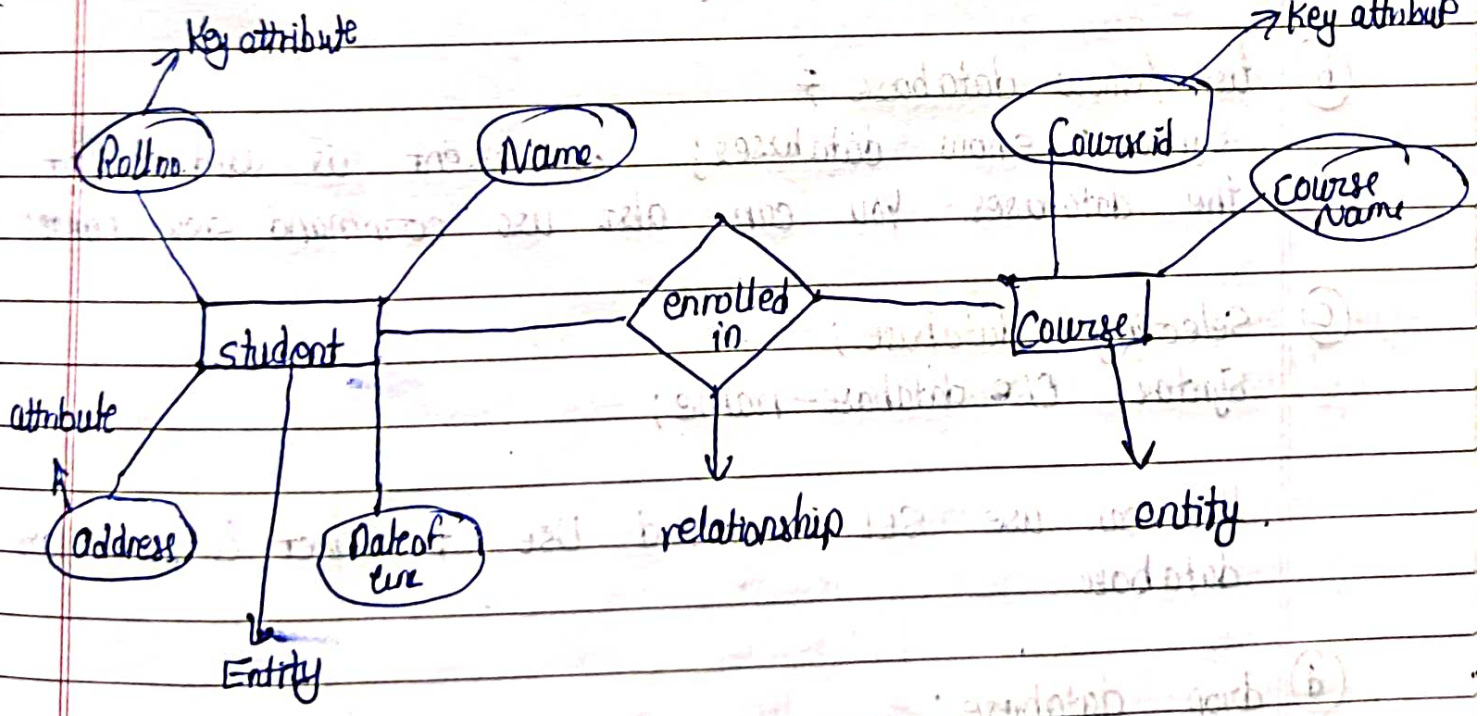
e.g:- If some courses are not enrolled by any student then the participation of the course will be partial.

11/04 ER Diagram (Entity Relationship diagram)

An ER diagram show the relationship among entity set. This diagram shows the complete logical structure of a database.
following are the symbols, shape that are use in ER diagram.

Shape	Shape Name	Purpose
	rectangle	represents entity type
	ellipse	attribute
	Diamond	relationship
	lines	to link attribute to entity set & entity sets to relationship.

	double ellipse.	represents multivalue attribute.
	double rectangle	Weak entity set
	dashed ellipse	derived attribute.
	double line	Total participation of an entity
	ellipse with underlined name of attribute	key attribute



UNIT-5

Structure Query Language [SQL] using MySQL.

Database :-

A database is used to store the collection of records in an organized form. It allow us to hold the data in table, rows and columns. We can create database in MySQL using create database statement.

(a) Creating database :-

Syntax : Create database database-name;
example : Create database mydatabase;

(b) list / show database :-
Syntax : show databases; statement is used to list the databases. You can also use command show schemas;

(c) Selecting database ;
Syntax : Use database-name;

You can use SQL command use to select a particular database.

(d) drop database ;

We can drop, delete or remove a MySQL database

with the drop database command.
Syntax: drop database [database_name];

This command will delete the database along with all the tables or indexes.

Table :

(a) Creating a table:

A table is used to organise a data in the rows & columns My sql allow you to create a table using create table statement.

Syntax: Create table [table name] [column 1 datatype, c
column 2 datatype,
:
:
Column datatype
);

Example: Create table Student (rollno int(5) primary key
name varchar (20), branch Char (40));

(b) Views : Views are virtual tables that do not store any data of their own. But display data stored in other tables
A view can contain all ~~or~~ or a few rows from a table. It can contain all or few rows from a table.

Syntax: Create view [view name] As select statement;

eg: create view view_name As select rollno, name from student;

12/04 (i) To see created view:

Command: select * from view_name;

(ii) Update view:

The alter view command is use to modify or update the already created view.

Command: sled comma colomns from table_name.
where condition;

(iii) Drop view:

We can drop the existing view using drop view statement.

Syntax: drop view [view name];

(c) Aliases.

Aliase are used to give a table or a column in a table or tempory name. Aliase are used to make column names more readable.
An aliase is created with the As keyword

Syntax: For column select column_name as alias name
from table_name;

eg: select name as n from student;

For table;

Select column1, column2 from table name as aliasname;
select rollno, name from student as s;

(d) Index

Index is a schema object. Indexes are used to retrieve data from the database more quickly. User cannot see Indexes they are just use to speed up searches.

(i) Creating index :

Syntax: Create index [indexname] on tablename (fieldname); column;

e.g: create index i on student (rollno.);

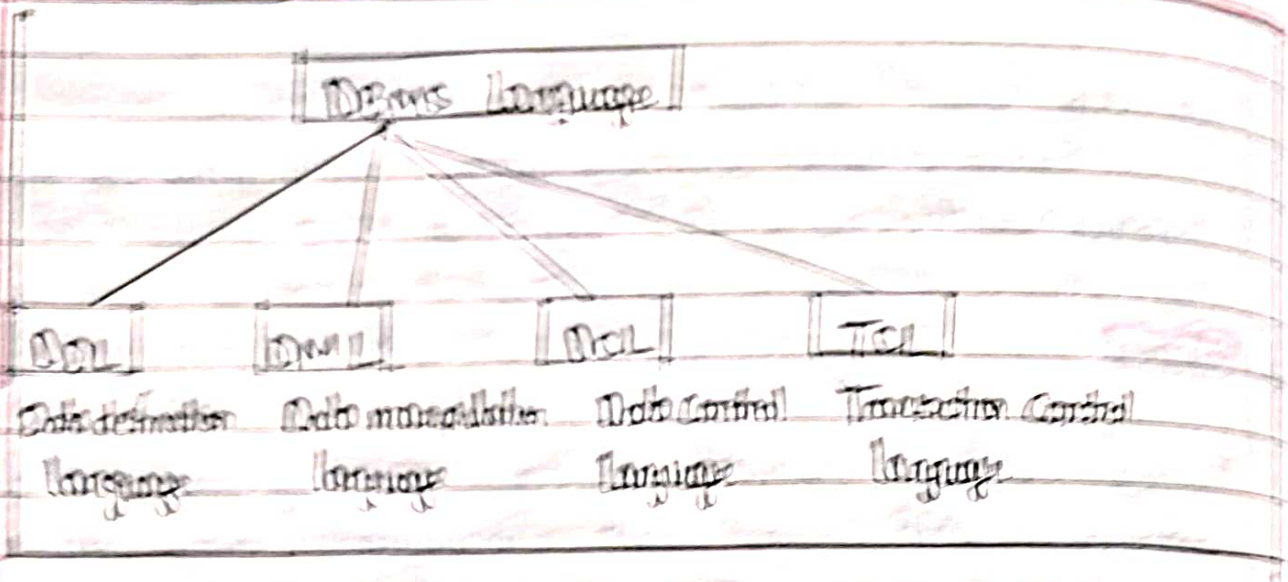
In SQL index is created on fields or columns of the table

(ii) Drop index :

alter table tablename drop index index_name;

Database languages

Database languages can be used to read stored update, the data in a database.



* 1 DDL

DDL stands for Data definition language it used to define database structure or pattern it is use to create schema, tables, indexes and constraints etc in the database it is used to create, modify the structure of your table and other objects in the database DDL commands can be used to add, remove, or modify tables within a database.

Following are the commands that are used:

A(i) **Create** : Create command is used to create the database or its objects like (table, index, views, function etc.)
This command creates a new database and a new table.

(i) Creating a database
Syntax: Create database[databasename];
e.g. Create database mydatabase;

B. **drop**: drop command is used to delete a objects from the database or schema. It can delete table, view, index and database

(i) droping a table: drop table command delete the whole table along with all data.

e.g: drop table [table name];

e.g: drop table students;

(ii) drop a database: drop database command delete the database and all the objects in it.

Syntax: drop database [databasename];

e.g: drop database mydatabase;

18/04

C. **ALTER Command**:

Alter command allows to alter or modify the structure of a database.

Using this command you can add new column, drop existing column and change the data type of column.

(i) Add column: Alter command allows us to add a column to a table.

Syntax:

alter table: < table name > add column < column name > < data type >

e.g: alter table students add column branch char(40);

(ii) Modify a column: using alter command you can

modify the type or size of the column.

Syntax: alter table <table name> modify, column <column name> <new column Name> <data type>
e.g: alter table student modify rollno. & no int(5);

(iii) Rename a table: alter command allow us to rename a table.
Syntax: alter table <tablename> rename to <newname>
e.g: alter table student rename to newstudent;

(iv) Rename a column: alter command allow us to rename a column in a table.
Syntax: alter table <tablename> rename column <old name> to <new name>.
e.g: alter table student rename column S.No. to rollno;

(v) Delete a Column: alter command allow us to delete a column and its entire data.
Syntax: alter table <tablename> drop column <column name>
eg: alter table student drop column branch;

D. Rename Command

Rename Command is used to rename an object.
~~rename a table~~
Syntax:
rename table <oldname> to <newname>;
eg: rename table student to newstudent;

E Truncate Command

It is used to delete all the rows from the table permanently. It removes all the records from a table including all the spaces allocated for the records.

Syntax: truncate table <tablename>

e.g: truncate table student;

DML [Data Manipulation Language]

DML commands are used to modify the database. DML commands deals with manipulation of data. DML commands allow us to insert, modify and delete the data. It is responsible for all form of changes in the database.

Example of DML command:

1. Insert command
2. Update Command
3. delete Command

1. Insert Command: It is used to insert the data into the row of table.

Syntax: insert into <tablename> values (val1, val2, val3 - valn);

insert into <tablename> <col1, col2, - coln) <val1, val2 - valn>;

e.g: insert into student values (1, "Name", "CE");

2. Update Command: This command is used to update or modify the value of a column

in the table.

Syntax: update < tablename > set < column = value >
[where condition]

e.g: update student set name = "Abc" where rollno=1;

19/04 3. Delete Command:

This command is used to remove one or more rows from a table.

Syntax:

delete from < table name > where < condition >;

e.g.

delete from student where rollno. = 1;

4. Select Command :

Select command is used to get data out of databases. This command will return the records.

Syntax:

select * from < table name > where < condition >;

e.g:

select * from student where rollno. = 1;

Difference b/w DDL & DML:

	DDL	DML
1.	DDL stands for data definition language	DML stands for data manipulation language

2.	DDL statements are used to create database, schema, constraints, users and tables.	DML statements are used to insert, update or delete the records.
3.	DDL has no further classification.	DML is of two types: 1. Procedural DML 2. Non-Procedural DML.
4.	Create, drop, rename and alter commands are used in DDL.	Insert, update and delete these commands are used in DML.

SELECT CLAUSES

Select query is used to fatch the data from Mysql database. It is used to return written one or more rows from the database table.

Syntax:

① select col1, col2 --- coln from <tablename> where <condns>;

② select * from tablename;

From Clause:

It is used to select some record from a table. It can also be used to get records from multiple tables using join condition.

Syntax:

① select col1, col2 --- coln from <tablename> where <condtn>;

② select * from tablename;

eg: select * from student where rollno = 1;

Where Clause:

It allows you to specify a search condition for the rows returned by a query.
Where clause is a keyword use to specify a condition.
Where clause can be used with insert, update, select and delete commands.
Syntax:
select * [from] <table name> [where] <condition>;
eg:
select * from student where branch = "IT";

Order by Clause:

Order by clause is used to sort the data in ascending and descending order based on one or more columns. The order by keyword sort the record in ascending order by default.
To sort the records in descending order use desc keyword.
Syntax:
select * [from] <table name> [where] <condition> order by col, asc/desc;
eg:
Select * from student where branch = "IT" order by age;
select * from student order by ~~name~~ marks;
select * from student order by age desc;

20/4 Group by Clause:

Mysql group by Clause is used to collect data from multiple records and group the result by one or more column.
Group by Clause returns one row for each group. It is used in select statement.
Group by Clause is used for aggregate function like (sum, count, min, max, avg.) to group the result by one or more column

Syntax:
Select column name from table name
where condition
group by column name
Order by column name;

Mysql evaluates the group by clause after the from, where and select clause and before having an order by clause.
Syntax:
from where select Group by Having order by

Having Clause:

The having Clause enables you specify conditions that filter which group result appear in the result
Having Clause is applied to summerized group.

records.
Having clause requires that a group by clause is present.

Difference b/w Where and having clause

- Where clause places conditions on the selected columns. whereas the having clause places condition on groups created by the group by clause.

Syntax:

```
Select columnname
from tablename
where condition
group by column name
having condition
Order by column
```

e.g: Select name, sum(salary) from employee
group by name
having sum(salary) > 30000;

OPERATORS : in MySQL

An operator is used to perform operations such as comparision and arithmatic operations. Operators are used to ~~perform~~ specify a condition in a statement in MySQL.

1. Arithmatic Operators

An Mysql arithmatic operators are used to perform arithmatic operations

① + Addition
② - Subtraction
③ * Multiplication
④ / Division
⑤ % modulus division

① + addition operator :- It is used to perform addition operation on the data iteams.

eg:- select 10 + 20;

+ operator perform on iteams includes either single column or multiple columns.

② - Subtraction :- It is used to perform subtraction operation on the data iteams

eg: select 10 - 20;

③ * Multiplication :- It is used to perform multiplication of data iteams.

eg: select 10 * 20;

④ / Division :- It is used to divide one number by another number.

eg select 10/20;

⑤ % Modulus :- It is used to get remainder when