→ DBMS
↓
(Database Management Systems)

Section A :

Overview of DBMS, Various views of data, data models, Introduction to database language, Advantage of DBMS over fil processing systems, responsibility of database administrator, introduction to client / server architecture, 3 levels architecture of Database systems, ER diagram (entity relationship), mapping constraints, keys, reduction of ER ~~table~~ diagram into tables.

X X

- Overview of DBMS :-

1) Data → (raw facts and figures) (which has some existence in real world)
- data is any known facts or any smallest information that can be recorded and have implicit meaning.
- or we can say data is fact and figures about an entity that can be recorded.
- eg → Deepak, BTech, audio, Number, map, etc.

2) Database →
- it is a collection of inter-related data.
- it is organised collection of data generally stored and accessed electronically from a computer system.
- eg → name, age, salary are the collection of related data.

Name	age	Salary
Ram	31	19000
Shyam	28	14000

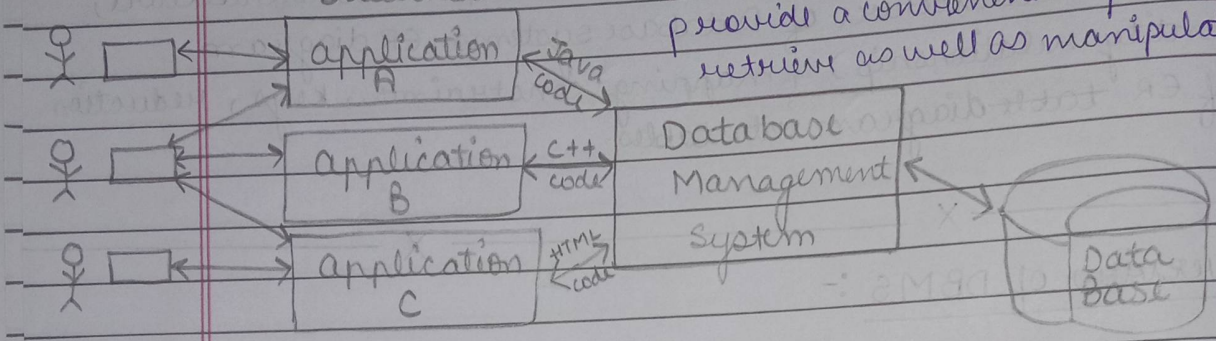
collection of data ↓

collection of age data.

2023.03.12 12:56

3) Database System: • it is a system in which an user uses the database technology in order to achieve an organized, store a large number of dynamic associated data with the help of Hardware, software (DBMS), operating system.

• Or ~~an application software~~ which provide a convinient way to store, retrieve as well as manipulate data



• So, database system consists of 5 major parts :-

Hardware, software (DBMS), data, Procedures, people.

4) Database Management System:

(mainly DBMS is used to manage the database).

• it is a set of software programs that allows users to create, edit and update data in database files, and store and retrieve data from those database files.

eg → oracle, MYSQL, MS SQL Server, SQL, DB2 (IBM)

• database management system (DBMS) is used to organised the data in the form of a table, schema, view and report etc, graph, tree, document.

• The primary goal of a DBMS is to provide a

way to store and retrieve database information that is both convenient and efficient.

• Database management system is the combination of 2 words: Database + Management system = DBMS

• DBMS is a collection of programs that enables users to create and maintain the database.

→ DBMS can also be define as an interface b/w n users, the application program and the operating system to access and manipulate that database.

↓

DBMS Application:

1) Banking : all transactions
2) airlines : reservations, schedule
3) universities : registration, grades
4) Sales : customer, products, purchases
5) online retailers : order tracking
6) manufacturing : production, orders, supply chain, inventory
7) Human resources : employee records, salaries, tax deductions.
8) Many more...

(we need data to derive some information from it.)

5) information:

• when data is processed, organised, structured or presented in a given context to make it more useful. It is called as information.

eg : 10 employee has salary 20K per month.

(we process data for future analysis.)

2023.03.12 12:56

→ difference between Data and information:-

Data	information
• derived from latin word "Datum".	• derived from latin word "informare".
• data is raw fact	• processed form of data.
• may or may not be meaningful.	• always meaningful.
• understanding is difficult.	• understanding is easy.
• data may not be in order.	• information should be in order.
• input to any system may be treated as data.	• output after processing system is information.
• eg: survey data	• eg: census report

example: data → Student's attendence data.

information → % of lectures attended by students.

$$\% = \frac{\text{Total Lectures attended}}{\text{Total no. of lectures conducted}} \times 100$$

eg:

if % < 40 then grade F
≥ 40 & < 50 then grade D
≥ 50 & < 60 then grade C
≥ 60 & < 70 then grade B
≥ 70 then grade A.

2023.03.12 12:56

↓ why DBMS?

- to develop software application in less time.
- for data integrity and security.
  (accuracy, consistent, completeness)
- data independent and effective use of data.
- for uniform data administration.
- to use user-friendly declarative query language.
- for concurrent access of data (more than 1 user can use), and data recovery from crashes.

↓

Advantage of DBMS :-

① data independence - application programs should be free & independent as possible from details of data representation and storage.

② efficient data access - DBMS contains technique for storing and retrieving data efficiently.

③ data integrity and security - if data is accessed through the DBMS which enforce integrity constraints of earth.

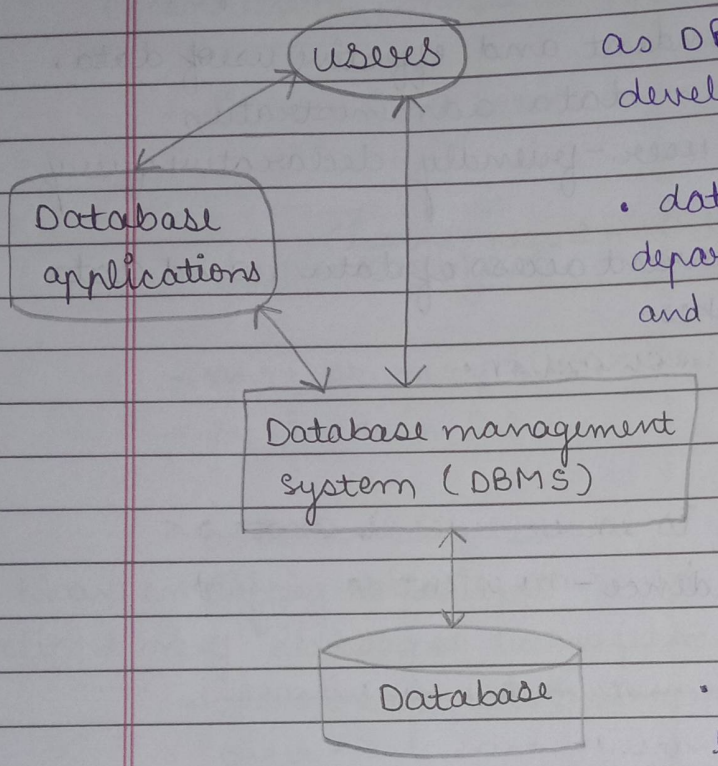
④ data administration - it is a high level function that is responsible for overall management of data resources in an organization which includes database planning, analysis, design, implementation, maintainance and data protection.

↓

it reduces data redundancy and make data to retrieve efficiently.

2023.03.12 12:56

Components of DBMS:-



- users can be of any type or kind such as DB administrator, system developer or database users.
- database application may be departmental, personal, internal and organisational.
- DBMS is a software that allows user to create and manipulate database access.
- database is collection of logical data as a single unit.

- ## Various views of data:-

- Views of data refers to how data is stored in database, what data is stored and what will be the structure of data.

- The database is highly complex data. Developers hide the data structure complexity which might not be similar to end users.

→ So, there are 2 views of data which are 1) data abstraction
2) schemas and instances.

① Data abstraction :

As we know data in database is stored in very complex structure. So, if user want to access this data then he has to go through this complex data structure.

Solution → to simplify the interaction of user and database, DBMS hides some information which is not of user interest, this is called as data abstraction.

it has 3 level of abstraction :-

a) physical level → (also known as internal level)

(memo mai kaise aapni harddisk mai data ko store karrhi hai!)

- this is lower level of abstraction.
- it tells how data is actually stored in database.

(iss level mai hum jo bhi data ko hide karte hai usko hum physical level of abstraction bolte hai!) and (till konsa access path use karenge!)

b) logical level → (also known as conceptual level)

- this level of abstraction describes what data is stored in Database and what relationship exits among those data.
- eg → there are 2 type of collection of data like of student and teacher.

Student
Name
Age
Gender

Teacher
Name
Age
Gender

→ both are stored in one database

relationship ? → Teacher teaches those students !

2023.03.12 12:56

c) View Level → different types of users interacts with different database.

- There may be many views of one data.

- higher level of data abstraction.

(means kisko kya dekhana hai, vo view level mai hota hai.)

eg: Students aur teachers dono ko result portal different views mai dekhta hai.

- Teachers un portal mai marks change kar skti hai but student sirf dekh skte hai apne marks. So, both views are different.

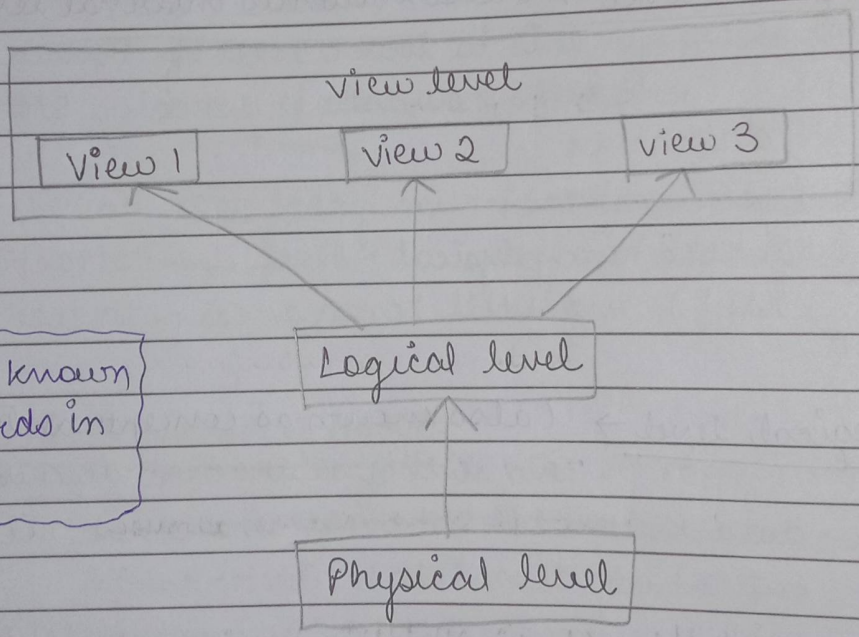


Table is known as records in DBMS.

② Schemas and instances:-

- design of data is known as schemas. it is basically a skeleton structure that represents the logical view of entire database.

→ it defines how the data is organised and apply constraints on the data.

constraints → eg: age cannot be 500, Name (ABC.

• Database system has various schema :-

① Physical schema → database design at physical levels.
• it tells form of storage like files, indexes, word, byte, blocks.

② Logical schema → design at logical levels.
• defines logical constraints, tables, views. & define relationship b/w tables and keys.

③ View Schema → design at view levels.
• we can have different views of database at different views levels.
↓
also known as Sub schema.

→ instances are collection of information stored in database at a particular moment is called instances.

eg :- in B.Tech admission :-

6/3/22 Ko 4 students ne admission liya

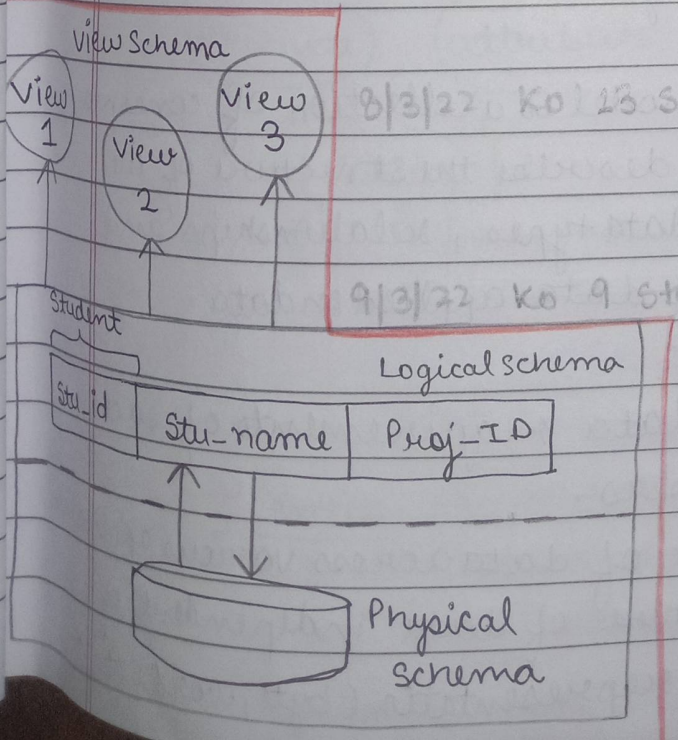
7/3/22 Ko 10 students ne admission liya → toh instance is 14

8/3/22 Ko 23 students ne admission liya → toh particular moment mai no. of admission is 14+23 = 37

9/3/22 Ko 9 students cancel the admission then instance will be 37 - 9 = 28

↓

(So, these are collection of information at particular moment/time!)



2023.03.12 12:57

Database administration:

- database administration refers to the whole set of activities performed by a database administrator to ensure that a database is always available as needed.

→ Software installation & maintenance.
→ managing and integrity.
→ Takes care of data extraction, transformation and loading.
→ monitoring performance.
→ data handling.
→ create database backup plan.
→ database recovery
→ database accessibility
→ provide support to users.
→ troubleshooting.

• Data models:-

- data models define how data is connected to each other and how they are processed and stored inside the system.

- data model is a collection of concepts that can be used to describe the structure of the database including data types, relationships and the constraints ~~of data~~ applied on data.

↳ it ensures: • the data requirements of each users.
• the use of data across various application
• the nature of data independent of its physical representation (byte, word, etc.)

2023.03.12 12:57

• The data model support communication b/w the users and database designers.

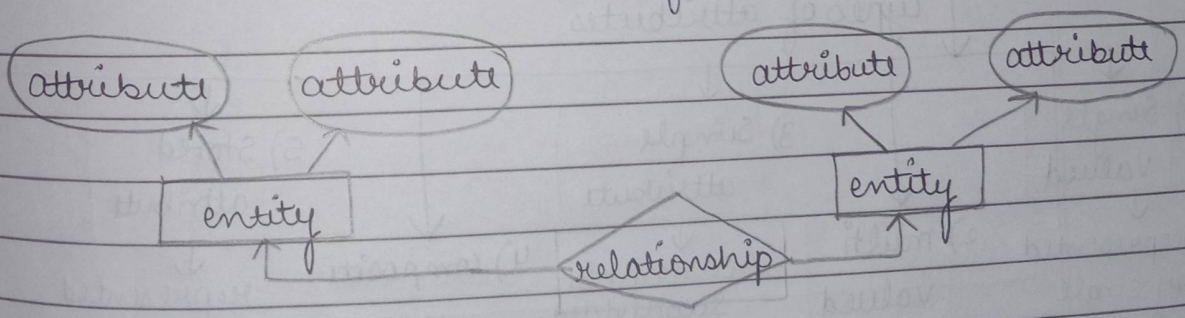
↳ characteristics:

1) bottom up approach must be followed.
2) diagrammatic representation.
3) application independent → data & their requirement can be expressed and distinguished easily.
4) data representation must not be duplicate.
↓
means jo data ek baar likh diya usko wapas nhi likhna.
5) consistency and standard validation must be maintained.
means constraints

→ There are 2 types of data models :-

① entity relationship model →

• entity relationship (ER) model is based on the notion of real world entity and relationship among them.



• ER was introduced by Chen in 1976.

→ entity → an entity in data model is a real world entity having properties called attributes. every attribute is defined by a set of values

2023.03.12 12:57

called domain.

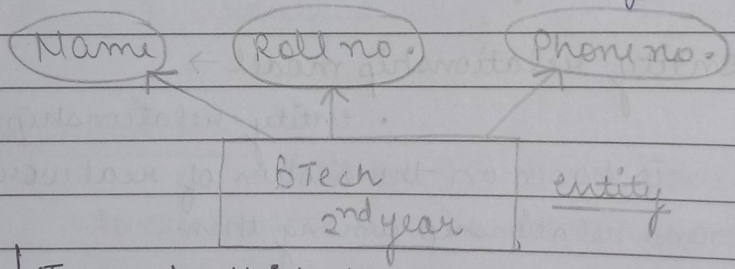
eg → in a school database, a student is called as entity. Students has various attributes like name, age, class, gender, etc.

→ relationships → the logical association among entities is called relationship. Relationship are mapped with entity in various ways.

→ attributes → • these are the units which describes the properties and characteristics of the entity.

• These are described property by each members of an entity set.

• it is represented by connected oval or ellipse to the concerned entity set.



Types of attributes

1) Single valued
↓
represented by single oval
eg: passport no, aadhar card no.
Passport no.

2) multi valued
↓
represented by double oval.
eg:
Phone no.

3) Simple attribute
↓
cannot be subdivided.
eg → roll no.

4) composite attribute
↓
can be subdivided
eg: Name
First name, middle name, Last name

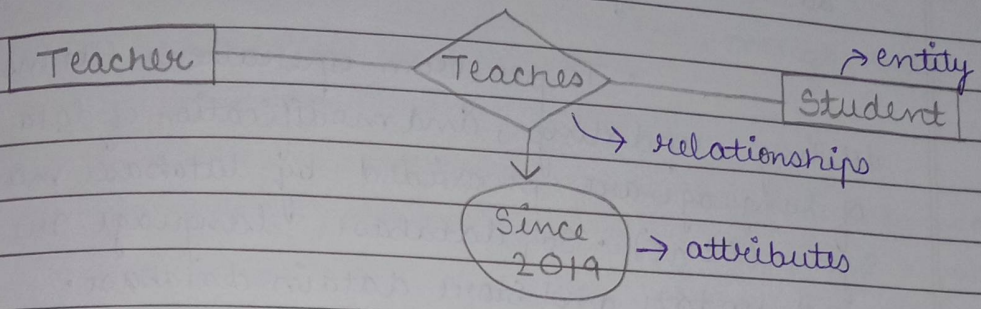
5) Stored attribute
↓
represented as
DOB
eg: DOB, age, name.
(data which is constant & fixed.)

6) Derived attribute
↓
derived from stored attribute
eg: represented as
DOB

2023.03.12 12:57

→ 7) Descriptive attributes → • the attributes which describes the relations. An attributes takes a null values when the entity does not have values for it.

eg:



② Relational model →

• The most popular data models in DBMS is relational model. It is more scientific model than others.

• The relational model in DBMS is an abstract model used to organise and manage the data stored in a database.

• It stores data in 2D inter related tables also known as relations in which each row represents an entity and each row represents an entity & each column represents the property of the entity.

eg: →

Student Table (relations)

Roll No.	Name	class	CGPA
001	Alix	9	9.1
002	Bob	9	8.5
003	Maya	9	9.5

Row (Tuples) → contains unique value!

Table (relations)

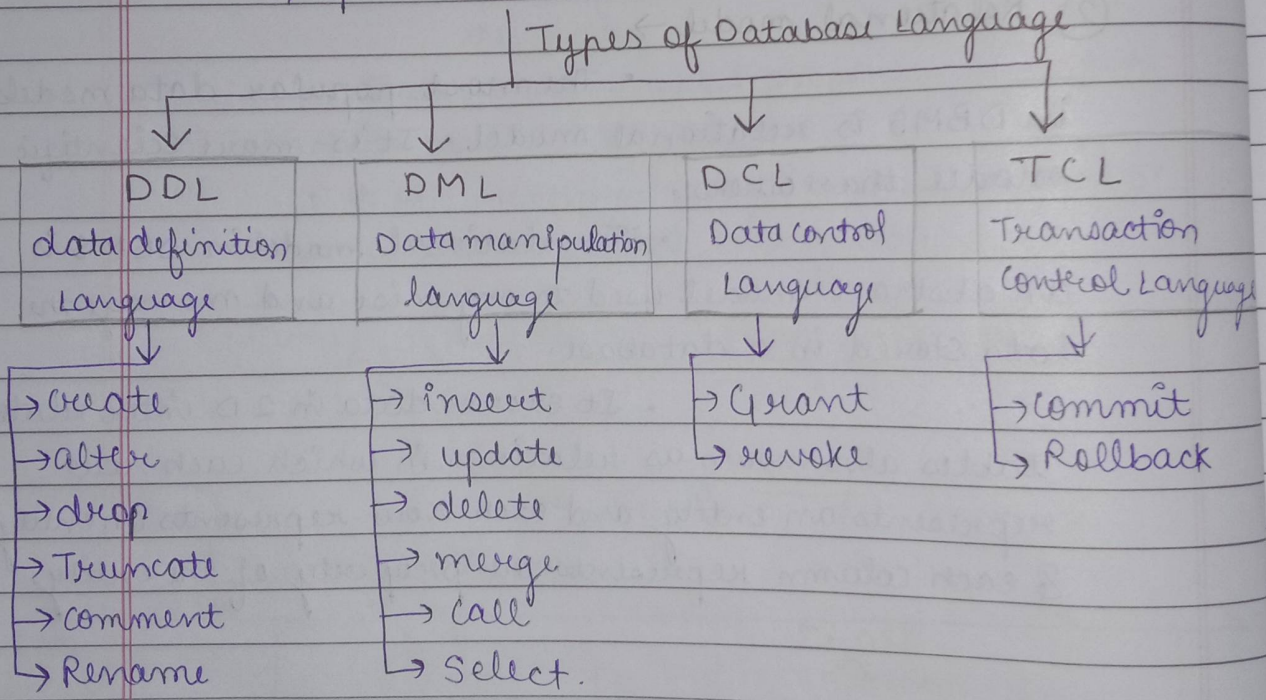
Column (attributes) → Contains values from a same domain.

2023.03.12 12:57

• Introduction to Database Language :-

• A DBMS provides appropriate language and interface for each category of users to express database queries and updates.

• To perform operations like insertion, deleting, updating, and modification of data a set of language are provided by database management system (DBMS). So, database language are used to read, update and store data in database.



① Data definition Language (DDL):

it is a language that provides / allow user to define the data and their relationship to other type of data. The DDL commands are :- create, alter (used to add, delete, modify table), Rename, Drop (remove table definition), comment, Truncate (remove all rows from a table but table structure & columns remains same).

2023.03.12 12:57

② Data manipulation Language (DML):

It is a language that provides a set of operations to support the basic data manipulation operation on data held in the database. The DML commands are :- insert, update, delete, select, merge, call.

③ Data Control language (DCL):

DCL is used to access the stored data. It is mainly used for revoke and grant the user access to a database. The DCL commands are : Grant (used to give access to the users or other rights or other opportunities to the database), Revoke (used to remove granted permission).

④ Transaction control language (TCL):

TCL language is a language which manages the transactions within the database. It is used to execute the changes made by data manipulation language statements. The TCL commands are :- Commit (used for permanently save all the changes made in the transaction of a database), Rollback (used to erase all the data modifications made from savepoint).

→ File System :-

. A file management in DBMS allows us to access to a single file or a tables at a time.

. in a file system, data is directly stored in a set of files. It contains flat file that has

2023.03.12 12:57

no relation to other file. (when only one table is stored in a single file, then this file is known as a flat file).

↓ disadvantage of file system

- data redundancy & inconsistency.
- difficulty in accessing data.
- data isolation.
- integrity problem.
- automaticity problem.
- concurrent accessing a nominee.

- **Advantage of DBMS over file system :-**

① Data redundancy and inconsistency → redundancy is the concept of repetition of data i.e each data may have more than a single copy. The file system cannot control the redundancy of data but DBMS can.

② Data sharing → The file system does not allow sharing of data or sharing is too complex. whereas in DBMS, data can be shared easily due to a centraliged system.

③ Data concurrency → concurrency means more than one user is accessing the same data at the same time. Anomalies occurs when changes made by one user get lost because of changes made by other user. The file system does not provide any procedure to stop anomalies but DBMS does.

2023.03.12 12:57

④ Data Searching → for every search operations performed on the file system, a different application program has to be written. while DBMS provides inbuilt searching operations.

⑤ Data security → a file system provides a password mechanism to protect the database.

⑥ Easy maintenance → it is easily maintainable due to its centrialized nature.

→ Types of users in database :-

① Database administrator → who has complete control over database. Database administrator is responsible for overall performances of database. He is free to take decision for database and provides technical supports. Database administrator has lots of responsibilities.

② Online users → these users communicates directly through an online terminal or directly through users interface and application program.

③ application programmers → users who are responsible for developing the application programs or user interface. It can be written in high level languages.

④ naive users → ~~xxx~~ users do not have any idea about Database and acess database by menu oriented interface.

2023.03.12 12:58

⑤ Sophisticated users → which have idea about database and access database by using queries.

⑥ Specialized users → who writes a specialized queries like queries in shell programming to access database.

- **Responsibilities of database administrator :-**

① deciding the conceptual schema:
decides the content of schema like Fields, tables, datatypes etc.

② deciding internal schema or physical storage:
how data is stored and represented.

③ deciding user view:
decides who can view how data data.

④ assisting application programmers:
the DBA provides assistance to application programmers to develop application programs.

⑤ approving data access:
the DBA decides which user needs access to which part of database.

⑥ monitoring performance: The DBA monitor

2023.03.12 12:58

performances of the system. The DBA ensures that better performance is maintained.

⑦ Backup and recovery:
The DBA ensures this periodically backup the database on remote servers. In case of failure, such as virus attack database is recovered from this backup.

⑧ Security and integrity:
A database is controlled by DBA. So, it is their duty to authenticate the user to confirms that the right person is accessing the database.

⑨ Create free space (by removing dumps) (false data)

⑩ Deciding hardware & software. (which is to be use in program)

· Introduction to client / Server architecture:-

· client / server architecture system operates in networ environment.

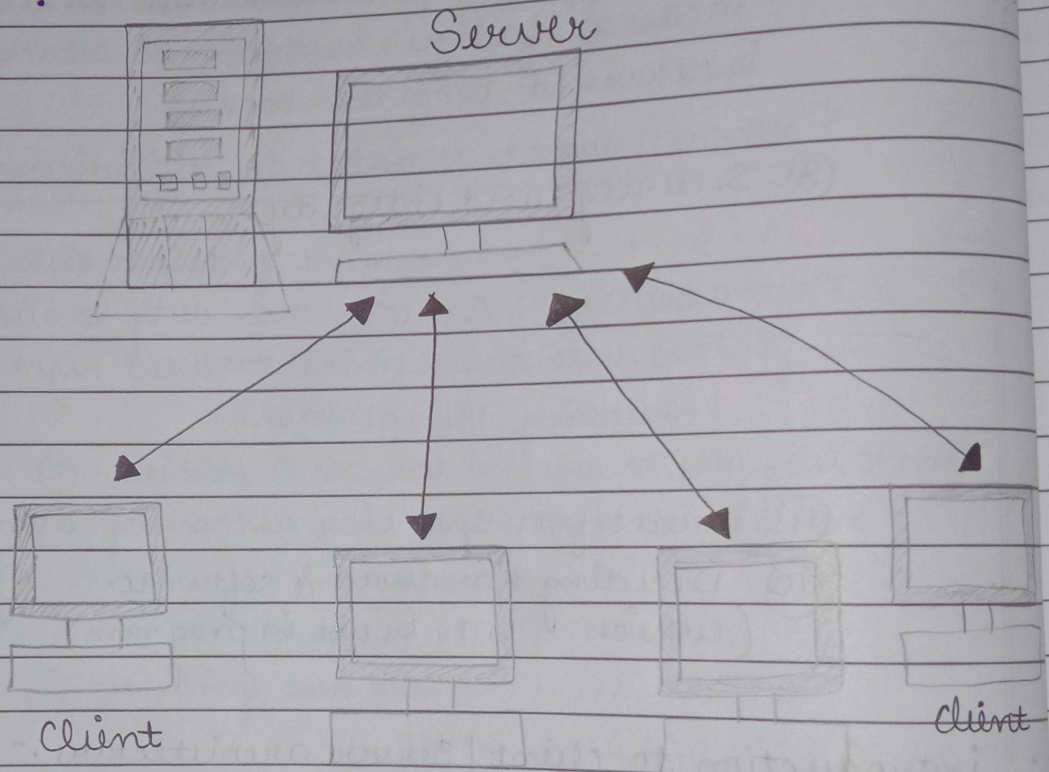
· The basic client / server architecture is used to deal with large no. of PC, web servers, database servers, and other ~~networks~~ components that are connected with networks

· Generally, the client process requires some resources which are provided by server to the client.

· Splitting the processing of an application b/w a front-end client and back-end processor.

2023.03.12 12:58

• both client and server are intelligent and programmable, so that the computing power of both can be used to devise effective and efficient application.



Client server architecture.

Types or components mainly :

Presentation logic
↓
responsible for formatting or presenting data on user screen

Processing logic
↓
It handles data processing logic, Business rule logic and data management logic.

Storage logic
↓
responsible for data storage.

2023.03.12 12:58

for eg :- System ke aander jo bhi data humko dekh rha hai vo presentation logic ki help se ho rha hai, aur vo data manage kaise hua yeh process kaise hua vo processing logic ki help se ho rha hai, and vo data store kahan ho rha hai ... yeh storage logic dekta hai!

- **advantage →**

① Server guarantee to those client who has permission only they may access and change the data.

② provide Security, friendly interface.

- **disadvantage →**

① traffic congestion on network is an issue.

② if a critical server fails then the client's request cannot be fulfilled.

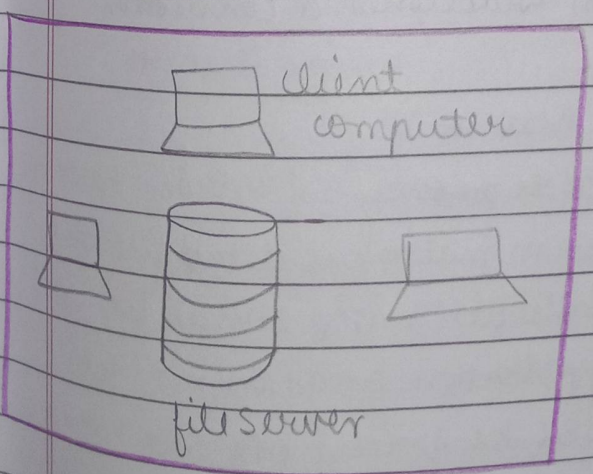
- **3 Levels architecture of database systems :-**

① **1-Tier Architecture →** • in this architecture, the database is directly available to the user. It means the users can directly sit on the DBMS and uses it.

• any changes done here will directly be done on the database itself. It doesn't provide a handly tool for each users.

• It is used for development

eg → database server, application server, etc.

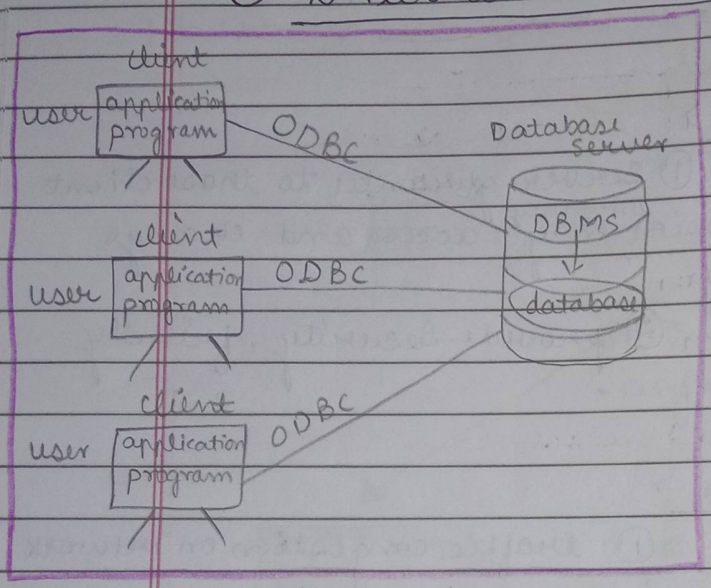


1-Tier Architecture

of the local application, where programmers can directly communicate with the database for the quick response.

② 2-Tier architecture → eg → oracle, SQL server, etc.

→ jab hum directly apne question teacher se puche bole to ye 2 tier hai



2-Tier architecture

- The 2-tier architecture is same as basic client server.
- in this, application on the client end can directly communicate with the database at the server. for this interaction we use ODBC (open database connectivity) → which help to call DBMS.

Advantage:
- modification is easy
- communication is fast

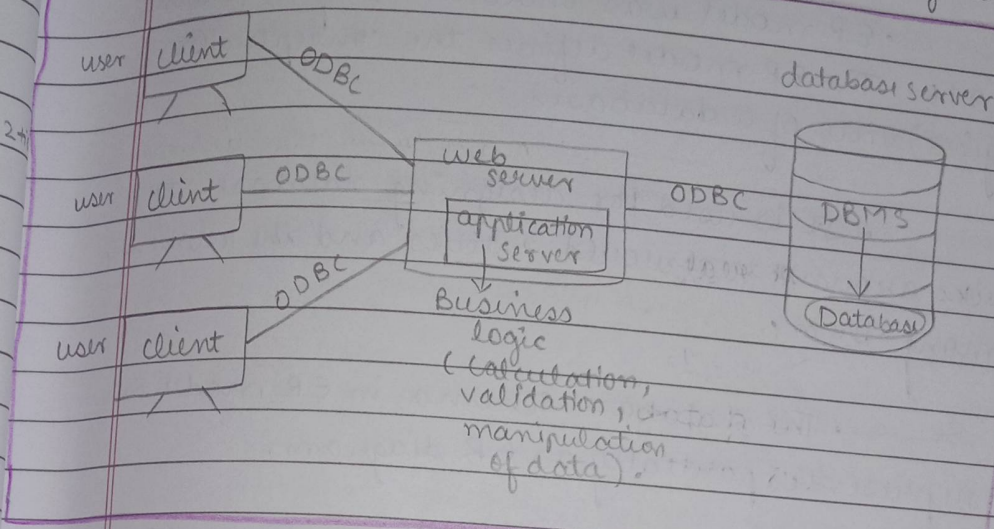
disadvantage:
- Performances degrades when no. of users increases.

- The user interfaces and application programs are run on the client side.
- The server side is responsible to provide the functionalities like: query processing & transactions managements.
- To communicate with DBMS, client side application establishes a connection with the server side.

for eg → jab hum bank se money withdraw karte hai tab hum ek form mai aapni detail fill karke wahan ke clerk ko dete hai (client) fir wo uss detail ko application program mai fill karega then ODBC ke jareye DBMS se hamara database mai se hamari detail check karega aur batayega kitne money hai hamare mai... withdraw ke layak hai yeh nhi.

2023.03.12 12:58

③ 3-Tier architecture → eg → jab hum youtube mai comments ke jaroor question puchte hai : is 3-Tier.



3-Tier Architecture

- The 3-Tier architecture contains another layer b/w the client & the server. in this, client cannot directly communicate with the server.
- The application on client end interacts with an application server which further communicates with the database system.
- end user do not has any idea about the existence of the database beyond application server. The database also has no idea about any other user beyond the application.
- it is used in case of large web applications.
- Many client can responded at an instant.

advantage:
- data security is guaranteed & scalability (no. of client can be more it can handle)
- allows complex processing to take place at server side.

disadvantage:
- create an increased need of network traffic management, fault tolerance.
- performance loss if distributed widely over internet.

2023.03.12 12:58

- **ER diagram :- (entity relationship)**

• ER model was introduced by Peter chenn in 1976. The ER model defines the conceptual(or logical) view of a database.

• It is used for designing database. It works around real world entities and the relationship among them.

• The database schema in ER model can be represented partially as ER diagram.

• By using ER models, we can easily tranforms data into relational tables. and can also used as a design plan by database developer to implement a data model in specific DBMS models.

↓ There are basic entity relationships modelling

① entities
② attributes
③ relationships

① **Entities →**

• an entity is a "thing" or "object" in the real world that is distinguishable from other object.

• It can be anything that has an independent existence

eg : in school database, students, teachers, classes, courses can be taken as an entity.

2023.03.12 12:59

- It is represented as rectangles.

Student	Teacher	Course

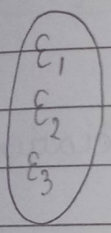
- entities has attributes (properties) that give them their identity.

eg → Student has roll no., names & addresses.

→ entities becomes table in relational model.

Entity Set: an entity set is a collection of similar type of entities that shares some attributes.

eg → an employee set contains all employee of bank, ~~the~~ a customer set contains all customer of bank.



entity set

- entity set need not to be disjoint (because employee of bank can be the customer of the same bank.)

② Attributes →

- attributes are used to describe the property of an entity.

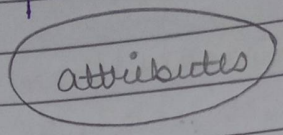
eg → a Student entity may have age, name, DOB, address, roll_number, mobile_number as attributes.

- For each attributes there is a set of permitted values called domain.

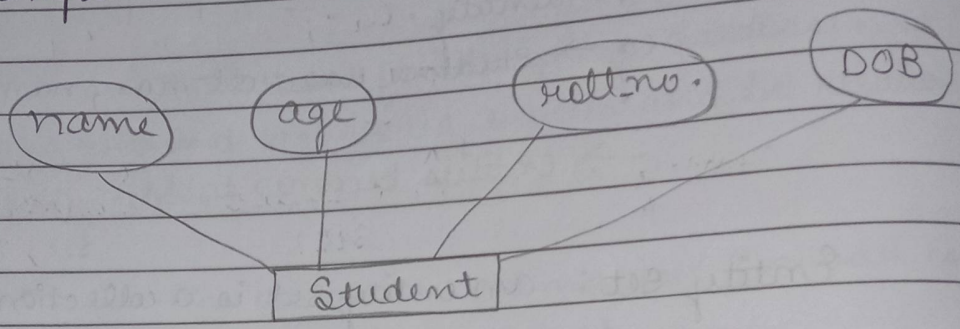
eg → Student name cannot be numeric values, age cannot be negative.

2023.03.12 12:59

• It is represented as ellipse.



example →

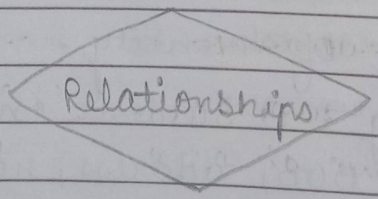


entity 1 : Rahul, 19, 101, 19-10-2003
entity 2 : Seema Singh, 20, 103, 9-5-2002

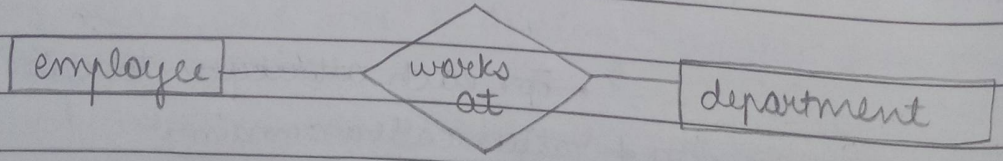
③ Relationships → • Relationship is an association among entities.

eg → a students enrolls in a course.
Rahul work at engineering department.

• It is represented as diamond shaped box.



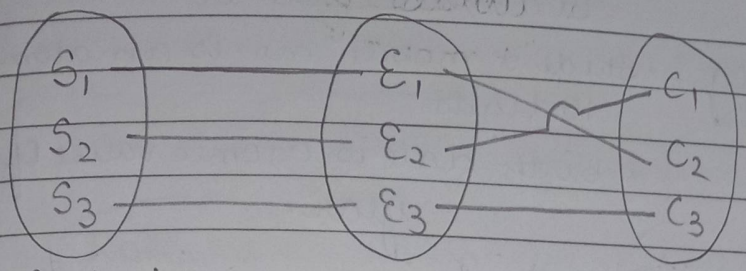
eg →



relationship set: a set of relationship of similar type is called relationship set.

2023.03.12 12:59

eg :- S_1 is enrolled at C_2, S_2 is enrolled at C_1, and S_3 is enrolled at C_3.

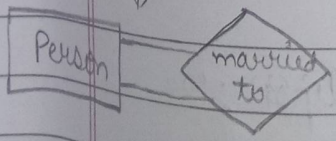


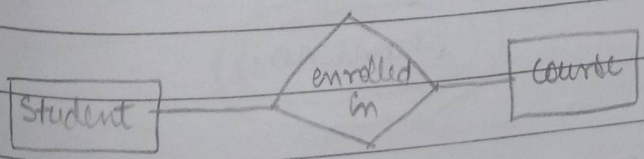
Student (entity set) | enrolled_in (relationship set) | course (entity set)

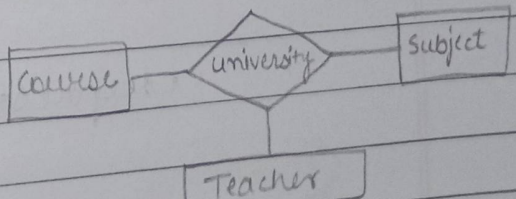
Degree of relationships :-

The number of different set of entities participating in a relationship set are called as degree of relationship set.

- **unary** (degree 1) — only 1 entity participate in relationship. eg → one person is married to only one person.
- **binary** (degree 2) — when 2 entity participates in relationship. eg → Student is enrolled in course.
- **Ternary** (degree 3) — when 3 entity participates in relationship. eg → Teacher teaches subject in a course in university.
- **n-ary** (degree n) — when n entity participates in relationship.







2023.03.12 12:59

↓ Types of attributes :-

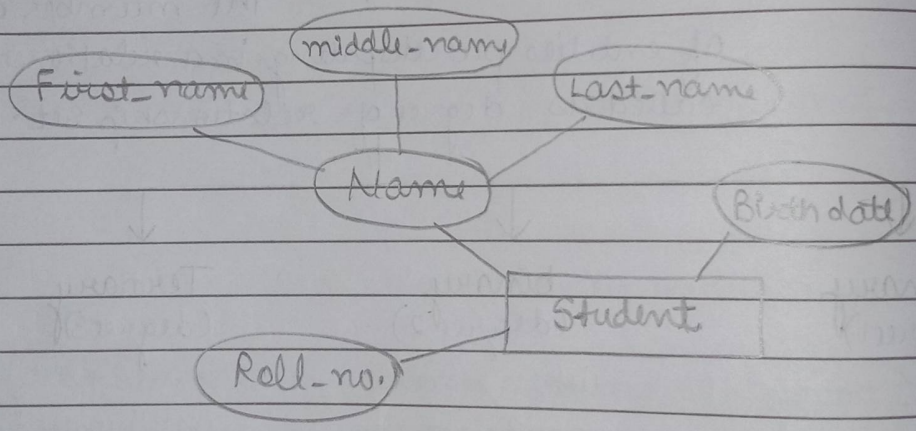
① Simple attribute → are atomic values which cannot be divided further.

eg : • Student mobile no. is an atomic value of 10 digits.
• birth date is atomic value of date-month-year.

② composite attribute → are made of more than one simple attributes.

• it is divided into tree like structures.

eg : • Students complete name have fistname, middle_name, Last_name.

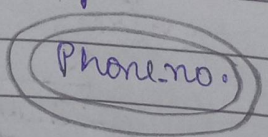


③ Single-valued attribute → it contains singled valued.

eg : Aadhar_card_no., Roll_no.

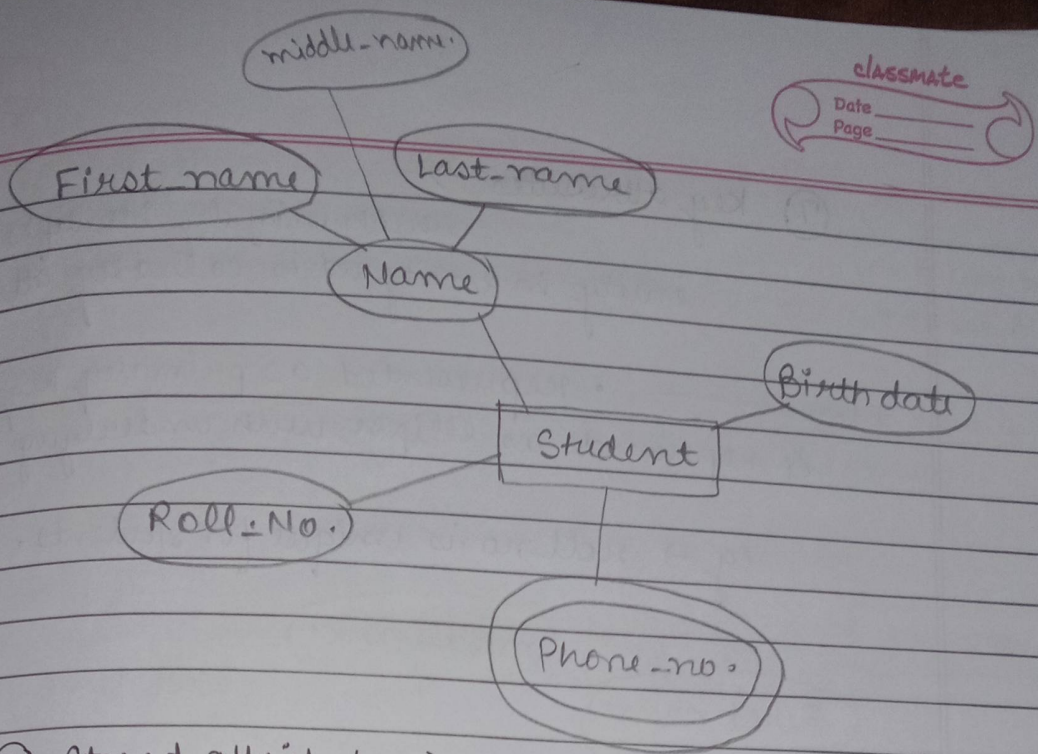
④ multi-valued attribute → may contains more than one values.

• represented by double-ellipse



eg → a person can have more than one phone no., email addresses, etc.

2023.03.12 12:59



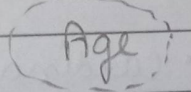
(5) Stored attribute → are physically stored in a database.

- mostly all attributes are stored in database except few ones.

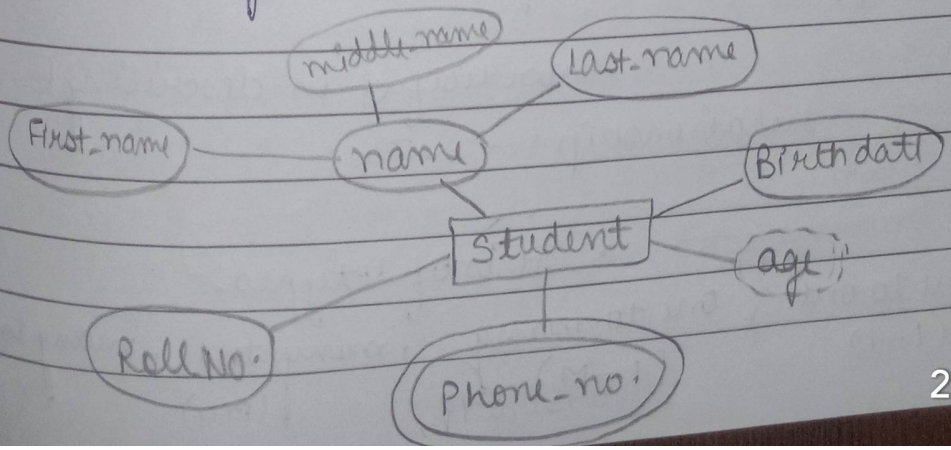
eg → roll_no., Name, birth_date, phone_no.

(6) derived attributes → are the attributes that do not exist in physical database, but their values are derived from other attributes present in the database.

- represent as dashed ellipse.



- eg → age can be derived from date_of_birth.



2023.03.12 12:59

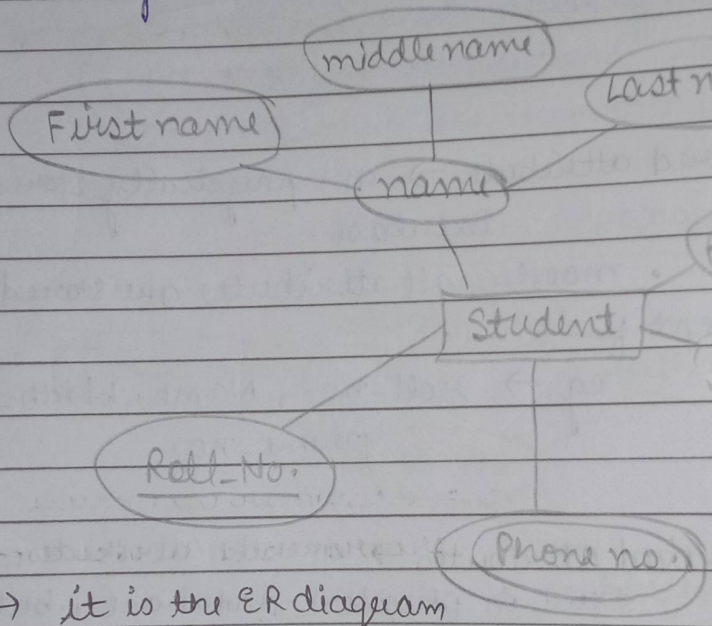
⑦ Key attributes →

which uniquely identifies each entity in entity set is called key attributes.

• represented as primary key.

& represents as ellipse with underlying lines.

eg → roll_no is unique for students.



→ it is the ER diagram of student entity.

• Mapping constraints :- (cardinality Ratio)

• Cardinality defines the number the entity of an entity set participates in a relationship set.

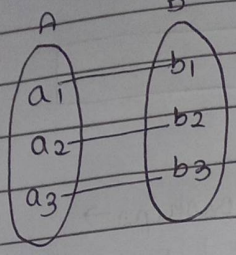
• most useful in describing binary relationship.

• It is of 4 types :-

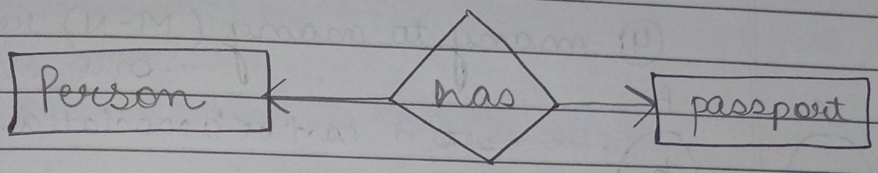
one to one (1-1), one to many (1-M), many to 1 (M-1), many to many (M-N)

① one to one (1-1) relationship →

one entity from entity set A can be associated with atmost one entity of entity set B and vice-versa.

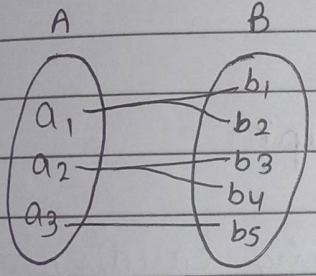


eg → a person has only 1 passport and a passport is given to one person.

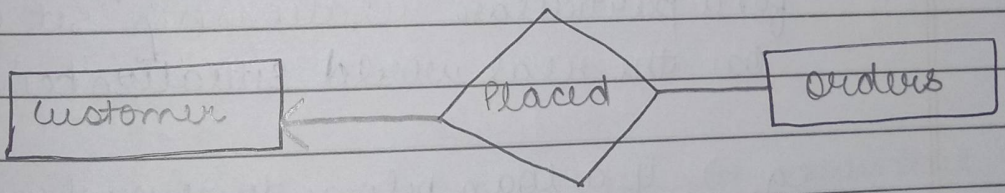


② one to many (1-M) relationship →

one entity from entity set A can be associated with more than one entity of entity set B however an entity from entity set B, can be associated ~~be~~ with atmost one entity.

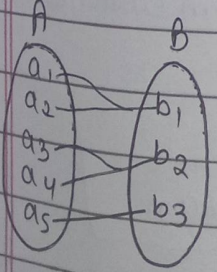


eg → a customer can place many orders but a order cannot be placed by many customer.



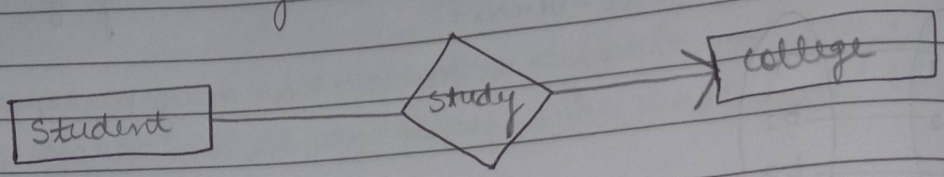
③ many to one (M-1) relationship →

more than one entity of entity set A can be associated with atmost one entity of entity set B, however an entity from entity set B can be associated to more than one entity from entity set A.



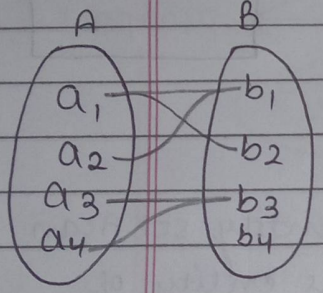
2023.03.12 12:59

eg → many students can study in single college but a student cannot study in many college at the same time.

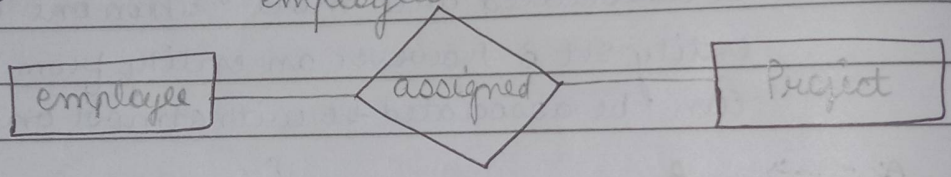


④ many to many (M-N) relationships →

Set A can be associated with more than one entity from entity set B and vice versa.



eg → an employee can be assigned to many projects & many projects can have many employees.

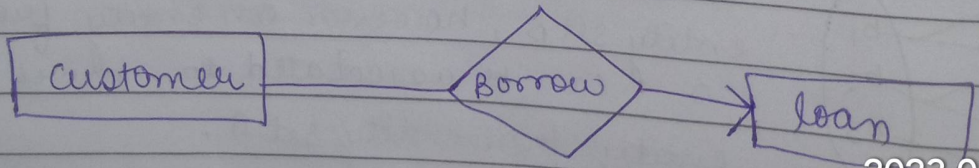


→ How to choose relationship?

↓

The appropriate mapping cardinality for a particular relationship set depends on the real world situation being molded.

eg → if a loan belong to several customers and a customer can have only one loan then the relationship set from customer to loan is M:1



2023.03.12 13:00

→ **participation constraints :-** It is applied on the entity participating in the relationship set.

Types

Total participation

- each entity is involved in the relationship.
- represented by double lines.

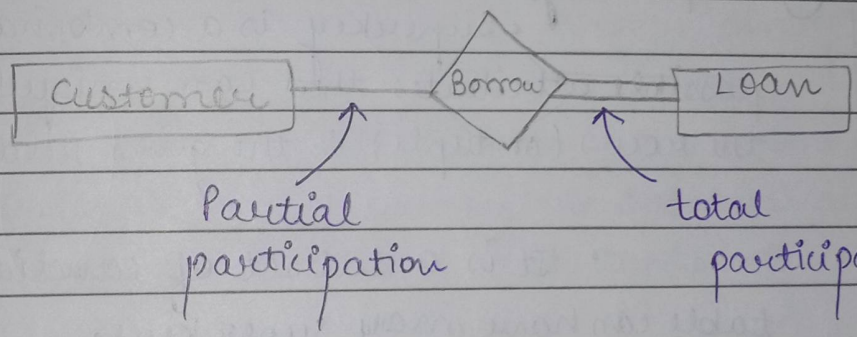
eg → every loan must be of customer

Partial participation

- not all entity are involved in the relationships.
- represented by single lines.

eg → all customers may not have loans.

eg →



- **Keys :-**
  - Keys is an attribute or set of attributes that uniquely identifies any records (or tuple) from the table.

Purpose

1) Key is used to uniquely identify any record or row of data from the table.
2) it is used to establish and identify relationships b/w tables.

2023.03.12 13:00

eg → employee table

foreign key

emp-Id	name	aadhar_no.	Dept-Id
01	Aman	775762540011	1
02	Neha	876834788522	2
03	Neha	946677898677	2
04	Vimal	796454638800	3

as here emp-Id & aadhar_no. are unique so it is a unique key.
whereas name & Dept_no are not unique.

↓ Types of keys

① Super Key →

• Superkey is a combination of all possible attributes that can uniquely identify the rows (or tuple) in the given relation.

• It is a subset of candidate key. A table can have many super keys.

• Super key may have additional attributes that are not needed for unique identity.

eg : {emp-Id} (all possible attributes)
{aadhar_no.}
{emp-Id, aadhar_no.}
{emp-Id, name}
{emp-Id, Dept-Id}
and so on....

2023.03.12 13:00

② Candidate Keys →

• A candidate keys is an attribute or set of an attributes which can uniquely identify a tuple.

• It is minimal Super Key because we select a candidate key from a set of super key such that selected candidate key is the minimum attribute required to uniquely identify the table.

• Candidate keys are defined as distinct set of attributes from which primary key can be selected. Candidate keys are not allowed to have NULL values.

eg: {emp_Id} (select those keys from
{aadhar_no.} super key who have unique identity)

③ Primary Keys →

• A primary keys is one of the candidate key chosen by the database designer to uniquely identify the tuple in the relation.

• Primary key cannot be Null. and must always be unique (not duplicate).

• It can never be changed or updated.

• The values of primary key must be assigned when inserting a record.

eg: {emp_Id} (select any one candidate key to make it primary key)

2023.03.12 13:00

④ Alternate keys →

• out of all candidate keys, only one gets selected as primary keys, remaining keys are alternate keys.

• in employee table

1) emp-Id is best suited for the primary key.
2) rest of attributes like aadhar_no. are considered as a alternate keys.

eg → {aadhar_no.} (candidate keys mai se primary key ko chorr kar jo bacchi... vo alternate keys).

⑤ Foreign keys →

• a key used to link two tables together.

• an attribute (or set of attribute) in one table that refers to primary key of another table.

→ [purpose] • to ensure (or maintain) referential integrity of the data.

• not unique most of the time.

eg :- Primary key. department Table (referenced relation)

Dept_Id	Dept_Name
1	Sales
2	marketing
3	HR

(continue with employee table)!

• also called as master table!

→ can take null value, may have name other than primary key.

2023.03.12 13:00

⑥ Composite Keys →

• a key that has more than one attributes is known as composite key. It is also known as compound key.

eg :-

Cust-Id	Order_Id	Product_Id	Product_count
C01	001	P111	5
C02	012	P111	8
C02	012	P222	5
C01	001	P333	6

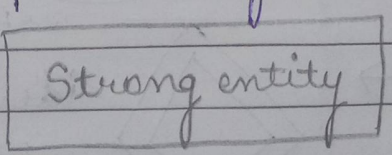
eg → {Cust-Id, Product_Id} (jab koi bhi unique nhi ho toh koi bhi 2 ko ek saath likhdo!)

• **Reduction of ER diagram into tables :-**

→ **Strong entity** → • Strong entity always have primary key. Its existence is not dependent on other entity. It is independent of other entity.

• A set of strong entity is known as strong entity set.

• represented by single rectangle.

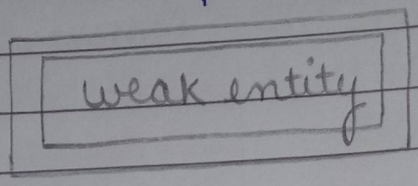


→ **weak entity** → • weak entity does not have a sufficient attributes to form a primary

2023.03.12 13:01

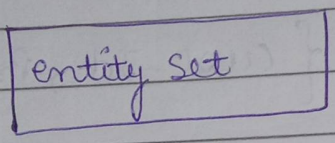
key. It is dependent on a strong entity to ensure its existence.

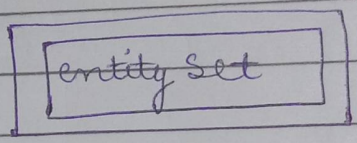
- A set of weak entity is known as weak entity set.
  → represented by double rectangle.

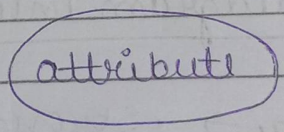


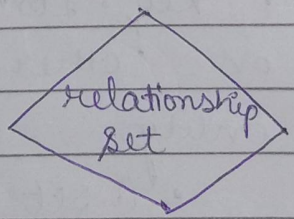
↓

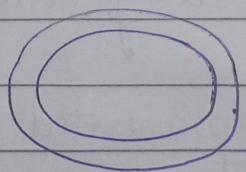
representation :

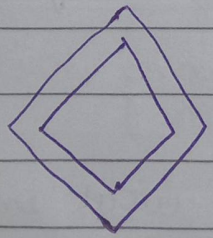
① rectangle →  Strong

② double rectangle →  weak

③ ellipse / oval → 

④ diamond → 

⑤ double ellipse →  multivalued attribute

⑥ double diamond →  identifying relationship type

2023.03.12 13:01

Date
Page

⑦ dashed ellipse → derived attribute

⑧ ellipse with line inside it → Key attribute

⑨ ellipse joined with other ellipse → Composite attribute

eg: in name → first name, Last name.

⑩ double line →

Total participation

⑪ single line →

Partial participation

⑫ Triangle → ISA — Specialisation & generalization

⑬ 1 : 1 →

⑭ 1 : M →

⑮ M : 1 →

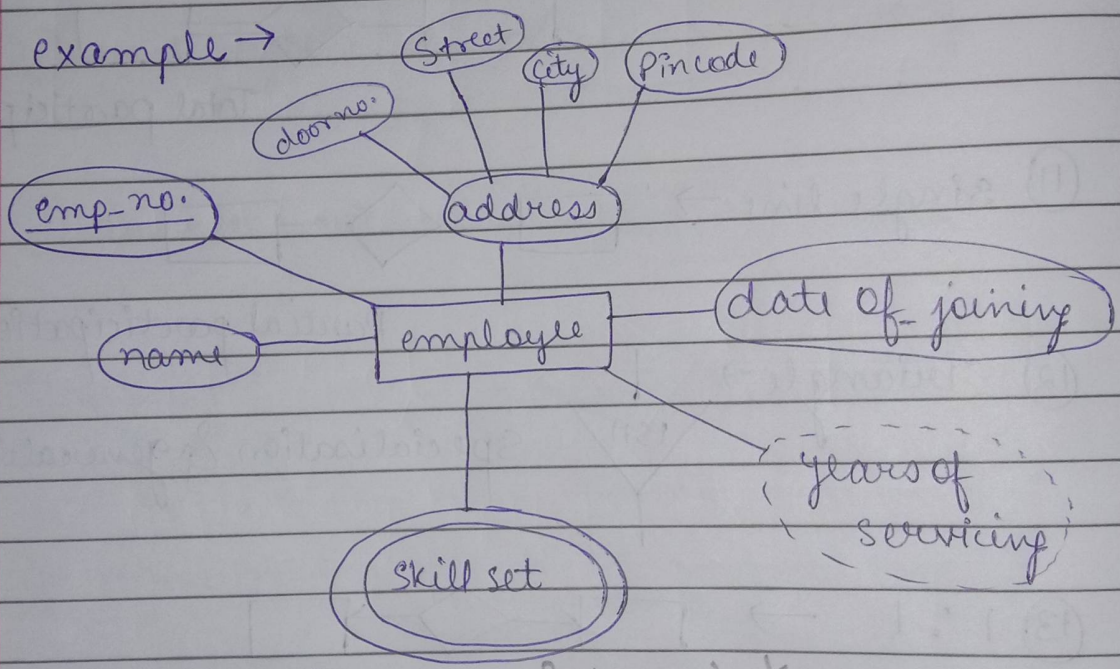
⑯ M : N →

2023.03.12 13:01

→ Rules for converting strong entity type :-

① Each entity type become a table.
② each single valued become column.
③ derived attribute are ignored.
④ composite attribute are represented by components.
⑤ multivalued attributes are represented by separate table.
⑥ Key attribute become primary key of table.

example →



as E_Id is primary key & foreign key.
employee & employee-skill set has separate table.

Employee
Emp ID. PK
Name
Door no.
Street
City
Pincode
Date of-Joining

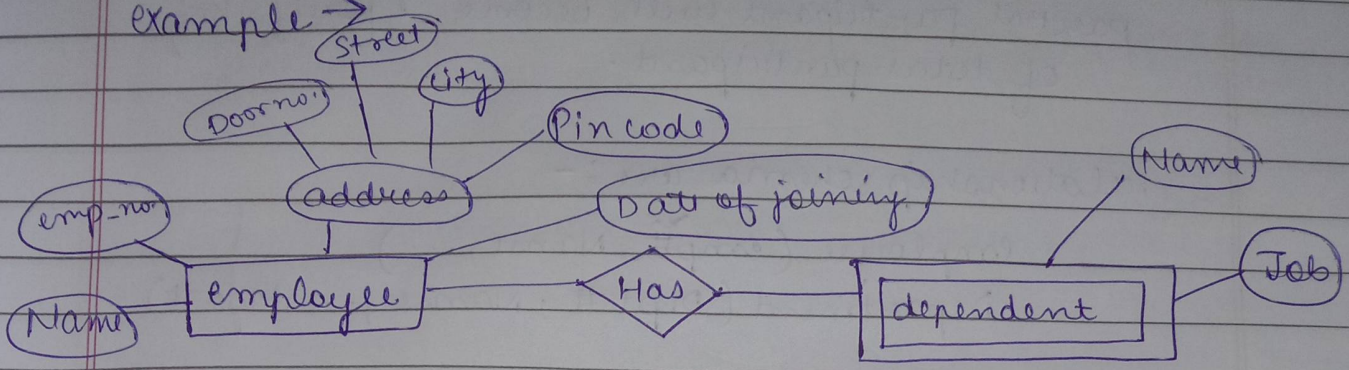
Employee-Skillset
Emp ID. FK
Skill Set

(Referential integrity)

2023.03.12 13:01

→ converting weak entity type :-

example →



emp-no. is primary key & foreign key.
(PK) (FK)

employee
emp-no. PK
Name
Door no.
Street
city
Pincode
date of joining

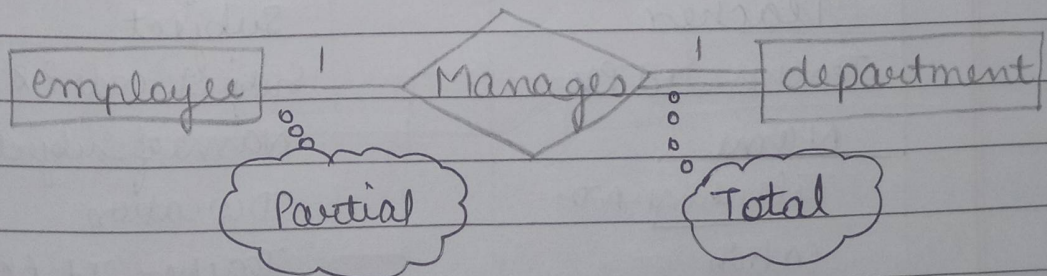
employee-dependent
emp-no. FK
Name
Job.

→ converting relationships :-

• The way relationships are represented depends on the cardinality and the degree of the relationship.

• possible cardinalities are :-

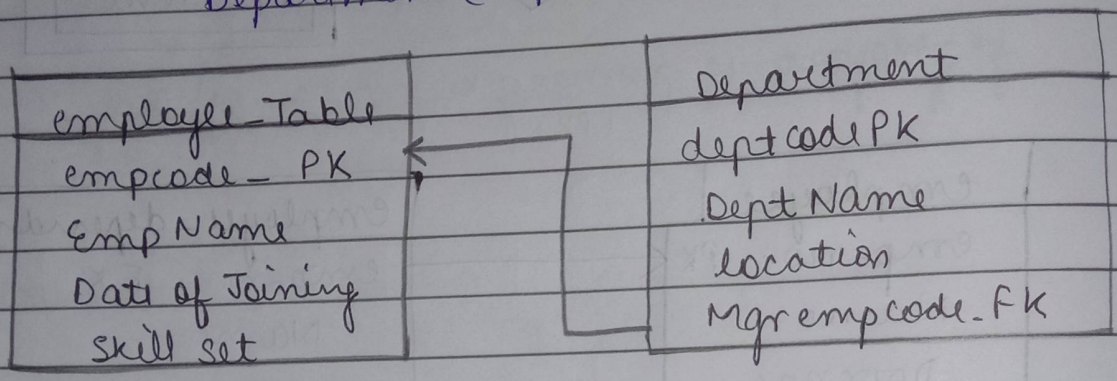
1 : 1 , 1 : M , M : 1 , M : N.



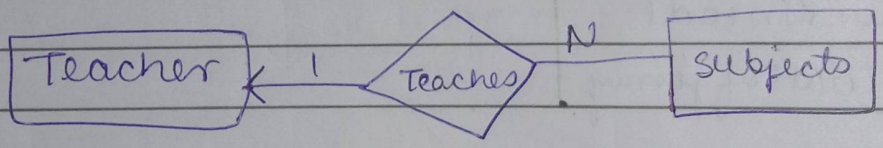
2023.03.12 13:02

a) combination of participation type →
The primary key of partial participant will become foreign key of total participant.

relationship schema are :-
employee (emp#, Name, ...)
Department (Dept #, Name, ... Mgr #)

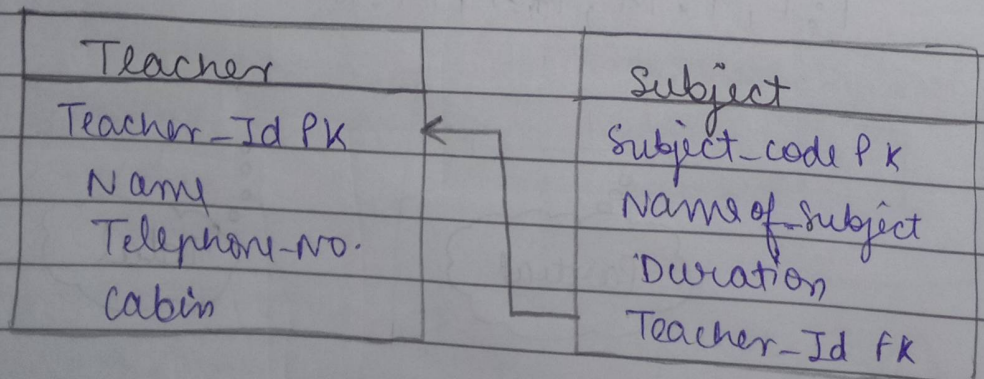


b) 1 : N (one to many)



in primary key of ~~region~~ relation on 1 side of relationship becomes foreign key on N side.

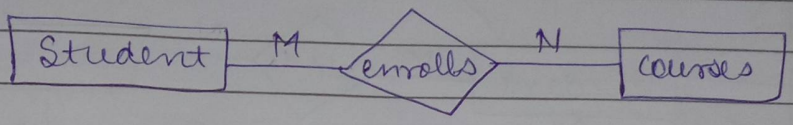
Teacher (Teacher_Id, Name, Telephone-no., ..)
Subject (code, Name, ..., Teacher_Id).



2023.03.12 13:02

c) M : N (many to many)

- a new table is created in the type of cardinality to represent the relationship.



- it has two contain 2 foreign keys – one from each table.

- The primary key of new table is the combination of the two foreign keys.

eg :-

Course table
Course_ID PK
Course Name
:
:

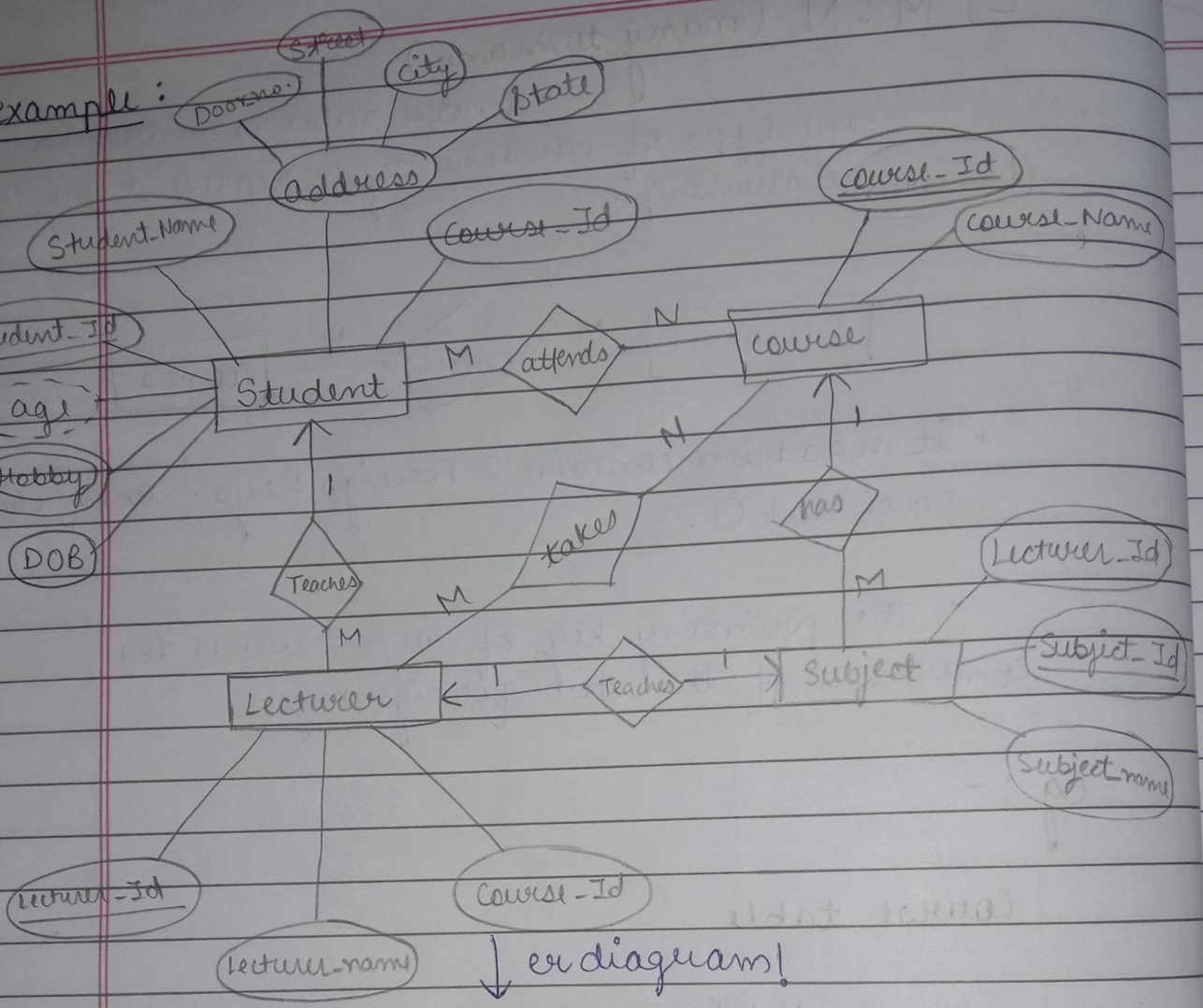
Student table.
student_ID PK
Student_Name
DOB
Address

enrollsTable
Student_Id FK
Course_Id FK
DO Issue
status
:
:

- Course table attribute Course_Id PK is a foreign key for enroll Table of attribute course_ID FK.
- Student table attribute Student_Id PK is a foreign key for enroll Table of attribute Student_Id FK.

2023.03.12 13:02

example :



er diagram!

Student
Student_Id PK
Student_Name
DOB
Door_no.
Street
City
State
Course_Id FK

Student_Hobby
Student_Id FK
Hobby

Stlident
Lecturer_Id FK
Subject_Id PK
Subject_name

Lecturer
Lecturer_Id PK
Lecturer_Name
Course_Id FK

Course
Course_Id PK
Course_Name

2023.03.12 13:03