Database Management System

Data :- It is collection of information which are stored in database.
Eg :- Natural data, Scientific data, financial data etc.

DBMS :- Collection of interrelated data and a set of programmes to access those data.

Database + Set of programs → DBMS

Database ⇒ The main goal is to provide a way to store and retrieve database information that is both convenient and efficient.

Relational database :- When information are saved in the form of tables, rows and columns.
For example, a typical business order entry database would include a table that describes a customer with columns of name, address, phone, number etc.

SQL → Structured Query Language (Relational database)

It is a standard query language for storing, manipulating and retrieving data in databases.

Database Management Systems

Database System Versus File Systems

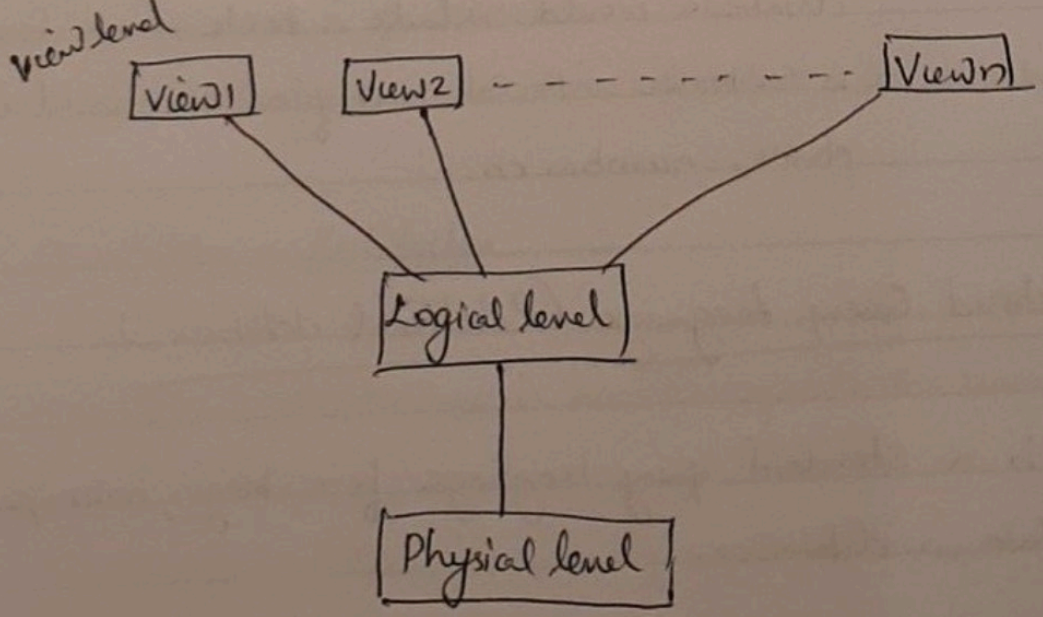
1. Data redundancy and inconsistency
2. Difficulty in accessing data
3. Integrity problems
4. Atomicity problems
5. Concurrent access anomalies
6. Security problems.

Database Abstraction

→ It is a process of hiding unwanted or irrelevant details from the end user.

→ Mainly there are three levels of abstraction for DBMS.

1. Physical level
2. Logical level
3. View level.



① Physical level

→ lowest level of abstraction

→ describes how the data are stored

→ [data is stored in the form of blocks of storage such as bytes, gigabytes etc]

→ describes complex low level data structures.

② Logical level

→ Describes what data are stored in the database, and what relationships exist among those data.

Eg:- Student (StudentID, Student name, age, course)

Course (CourseID, Coursename)

③ View level

→ The highest level of abstraction

→ This level describes the user interaction with database system.

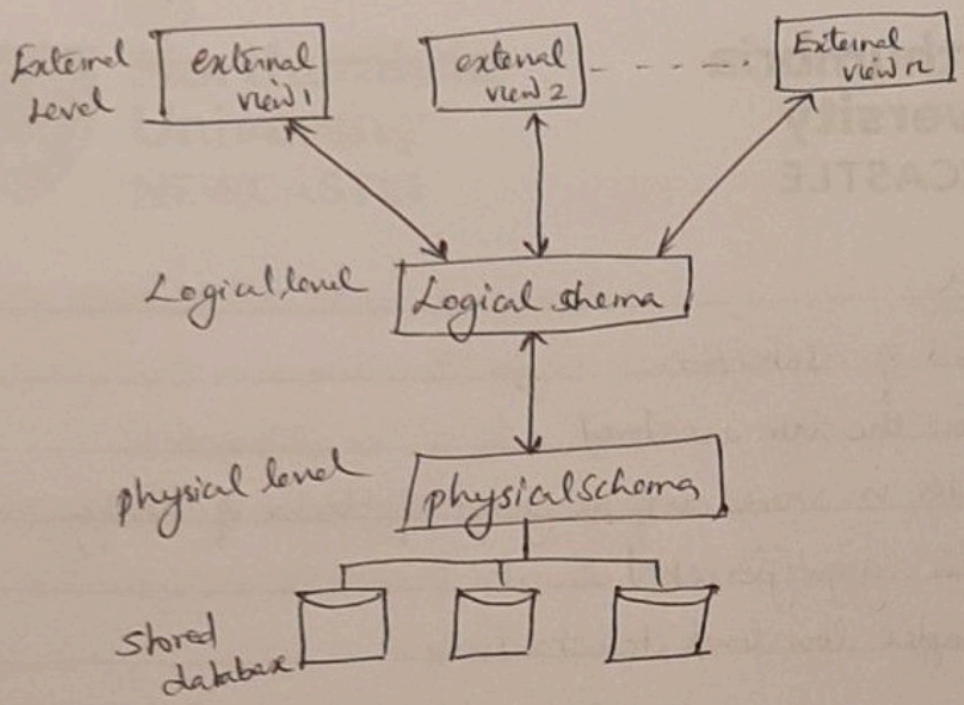
Three Schema Architecture

→ Three schema architecture is used to seperate the user applications and physical database.

① Physical schema

② Logical schema

③ External schema.



Data Independence

→ Capacity to change the schema at one level without having to change the schema at the higher level.

There are two types of data independency.

① Logical data independence :- It used to seperate the external level from logical level.

② Physical data independence :- It used to seperate the logical levels from the physical level.

ER Models (Entity Relationship Model)

→ ER models describes the structure of a database with the help of a diagram is called Entity-Relationship diagram.

→ It describes the conceptual view of a database.

→ This model is used to define the data elements and relationship for a specified system.

ER-diagram has three components:

① Entity :- It is a real world object.

→ An entity can be place, person, object, event.

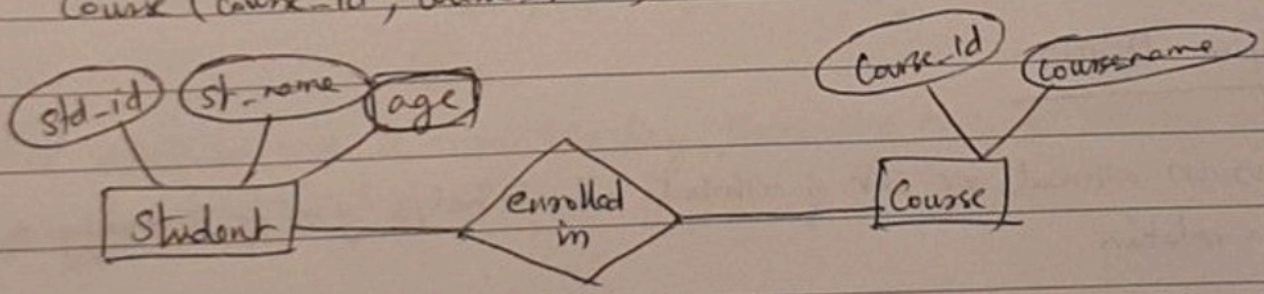
② Attributes → It describes the property of an entity.

③ Relationships → It used to describe the relationship between entities.

Eg:-

Student (st_id, st_name, age)

Course (course_id, course_name)



Attributes

These are the properties which describes an entities in an entity set.

Types of attributes

① Simple attributes :-

② Composite attributes :- It can again be divided further.

③ Single valued attribute :-

④ Multivalued attribute

⑤ Derived attribute.

⑥ Key attribute.

Relational Model

→ Proposed by E. F. Codd

→ Relational model represents the database as a collection of relationships

Properties of Relations

① Name of the relation is distinct from all other relations.

② Cells of the table should hold a single value.

③ Each attribute contains a distinct name.

④ Each row is unique

⑤ The values of an attribute should be from the same domain.

Key in DBMS

→ It is an attribute or set of attribute which helps you to identify a row in in a relation

① Candidate key :- All unique keys in table are candidate key

② Primary key :- It is unique but it should be not null

③ Alternate key :-

④ Super key

⑤ Foreign key

⑥ ~~Partial~~ Partial key.

④ Super key

→ Set of attributes that can identify each tuple uniquely.

→ Superkey consists of any number of attributes.

→ ~~Superkey~~ superset of any candidate key is superkey.

→ Candidate key + Anything.

Eg:

- Let a relation R have attributes (a_1, a_2, a_3) and a_1 is the candidate key.

Superkeys are $\{a_1, a_1a_2, a_1a_3, a_1a_2a_3\}$

- If 'N' attributes with one candidate key then the number of possible superkeys

2^{N-1} $\quad R(a_1 a_2 a_3 \ldots a_n)$

(a)

- If 'N' attributes with 2 candidate key, then the no: of possible superkeys.

Superkeys $= 2^{N-1} + 2^{N-1} - 2^{N-2}$

Foreign key

- In the relational databases, a foreign key is a field or a column that is used to establish a link btn two tables.
- A foreign key in one table used to point primary key in another table.
- An attribute 'x' is called as a foreign key to some other attribute 'y' when its values are dependent on the values of attribute 'y'.
- The attribute 'x' can assume only those values which are assumed by the attribute 'y'.
- Here, the relation in which attribute 'y' is present is called as <u>referenced relation</u>.
- Here, the relation in which attribute 'x' is present is called the <u>referencing relation</u>.

Eg ① 'Y'

Table: Employee

Employee_ID	Employee_Name
1	John
2	Alex
3	James
4	Roy
5	Kay

↳ Referenced table.

'X'

Table: Salary

Employee_ID_Ref	Year	Month	Salary
1	2012	April	30000
1	2012	May	31000
1	2012	June	32000
2	2012	April	40000
2	2012	May	40000
2	2012	June	41000

↳ Referencing table

Eg ② Primary key & Foreign key in same table.

Primary key → Foreign key

Emp_ID	name	Manager_ID
1	A	3
2	B	3
3	C	NULL
4	A	6
5	F	6
6	E	NULL

→ Foreign key references the primary key of the table.
→ Foreign key can take only those values which are present in the primary key of the referenced relation.
→ Foreign key may have a name other than that of primary key.
→ Foreign key can take the NULL value.
→ In fact, foreign key is not unique most of the time.

Constraints in DBMS

→ In DBMS, constraints are the set of rules that ensures that when an an authorized user modifies the database they do not disturb the data consistency.

→ Constraints enforce limits to the data or type of data that can be inserted, updated, deleted from a table.

Types of Constraints

① Key Constraints :- • All the values of the primary key must be unique and not null.
• There can be multiple key in a table.

② Entity-Integrity Constraints :- It states that the primary key value cannot be null. We use the primary key to identify individual rows in the table. If the value of the primary key is null, we cannot identify those rows.

③ Domain Constraints :- Domain constraint defines the domain or set of values for an attribute.
It specifies that the value taken by the attribute must be the atomic value for its domain.

④ Referential Integrity Constraints.

→ This constraint is enforced when a foreign key references the primary key of a relation.

→ It specifies that all the values taken by the foreign key must either be available in the relation of the primary key or be null.

PK

Roll-no	name	age
1	A	20
2	B	20
3	C	21

Student table ↳ Referenced table

FK

C-id	C-name	Roll-no
C_1	DBMS	1
C_2	OS	2
C_3	CN	3

Course table ↳ Referencing table

Referenced table	Referencing table
① Insert – no violation	Insert - may cause violation
② Delete - may cause violation on delete cascade	Delete - no violation
③ update may cause violation on update cascade.	update - may cause violation

Mapping Cardinality in DBMS

- Cardinality represents the number of times an entity of an entity set participates in a relationship set.
- For Binary relationship set there are entity set A and set B then the mapping cardinality can be one of the following.

For eg:- Employee — Works in — Department

① One-to-one

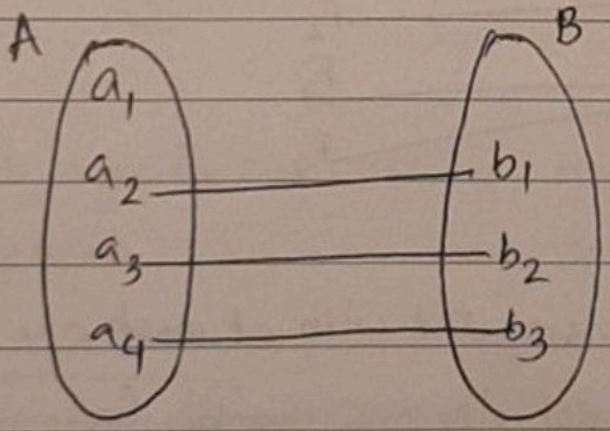
② One to many

③ Many to one

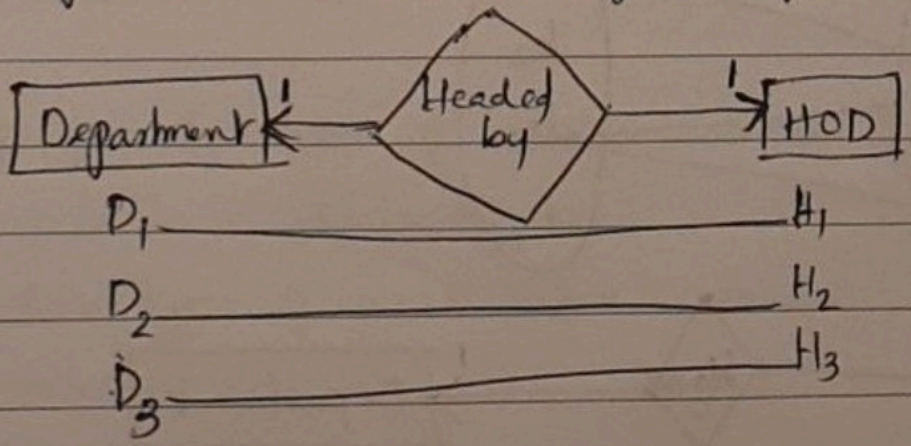
④ Many to many

① One to one relationship

- An entity in set A can be associated with atmost one entity in set B.
- An entity in set B can be associated with atmost one entity in set A.

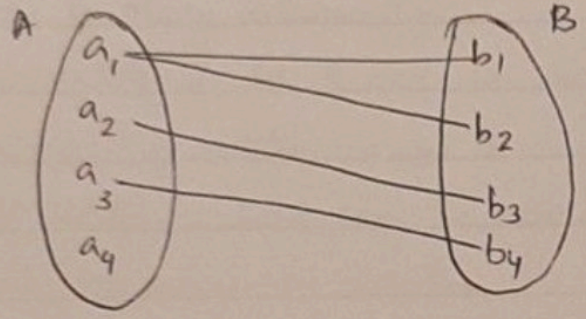


Eg:- Given below is an example of the one-to-one relationship. Here, one department has one head of the department (HOD).

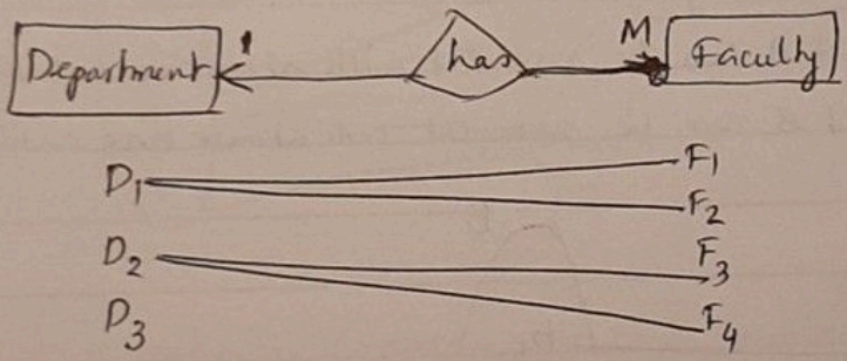


② One to many relationships

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with almost one entity in set A.

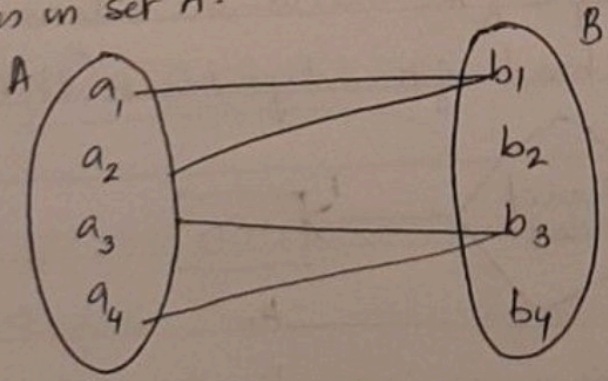


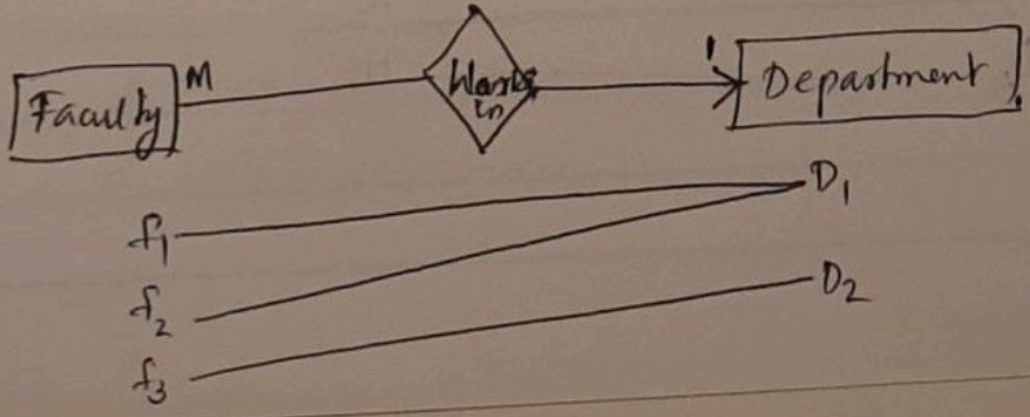
Eg:-



③ Many to one relationship

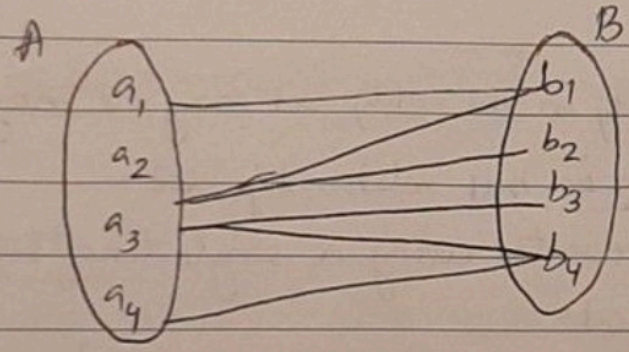
- At entity in set A can be associated with at most one entity in set B.
- At entity in set B can be associated with any number (zero or more) of entities in set A.



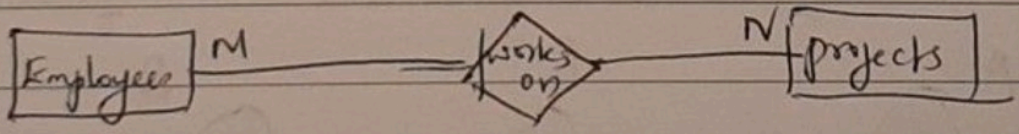


④ Many to Many Relationship

- An entity in set A can be associated with any number (zero or more) of entities in set B.
- An entity in set B can be associated with any number (zero or more) of entities in set A.



Eg:- In a particular company, multiple people work on multiple projects. They some many to many relationships.



Participation Constraints

It specifies the number of ~~separate~~ instances of an entity that can participate in a relationship set.

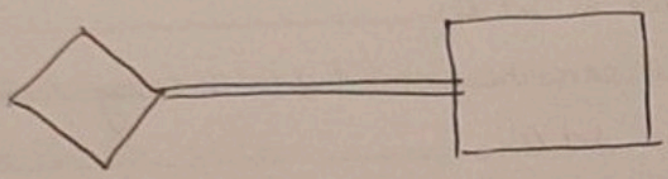
There are two types of participation constraint.

① Total participation

② Partial Participation

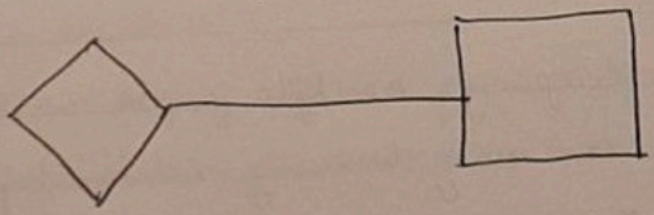
① Total Participation

- It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
- Total participation is represented using a double line b/n the entity set and relationship set.

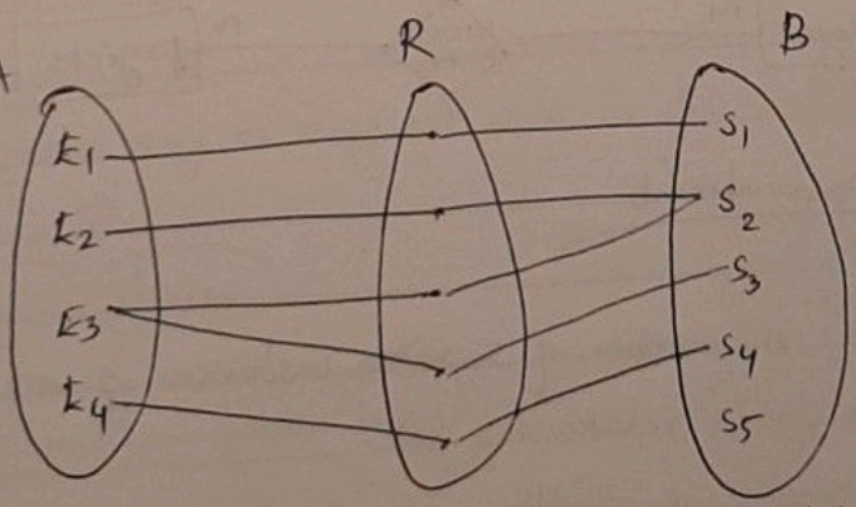


② Partial Participation

- It specifies that each entity in the entity set may or may not participate in the relationships instance in that relationship set.
- Partial participation is represented using a single line between the entity set and relationship set.



Eg:-



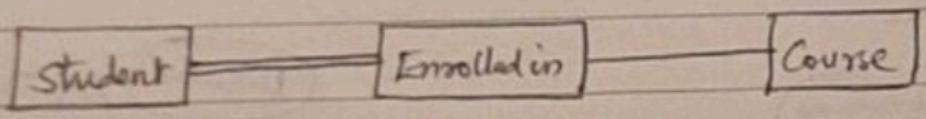
Here A & B are the entities and R is the relationship b/n them.

A to R, the participation is Total participation.

where B to R is partial participation.

Eg:- If each student must enroll in a course, the participation of student will be total

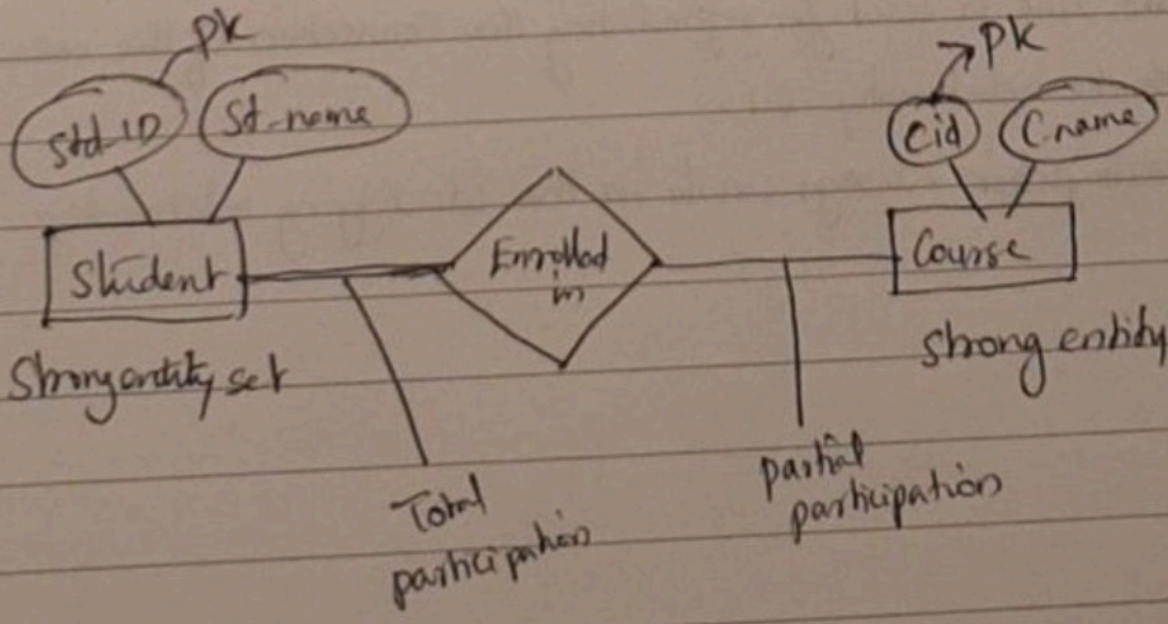
If some courses are not enrolled by any of the student, the participation of course will be partial.



Strong Entity Set

- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.
- In other words, a primary key exists for a strong entity set.
- A single rectangle is used for representing a strong entity set.
- The relationship between two strong entities is represented by a single diamond.
- Total participation may or may not exist in the relationship.

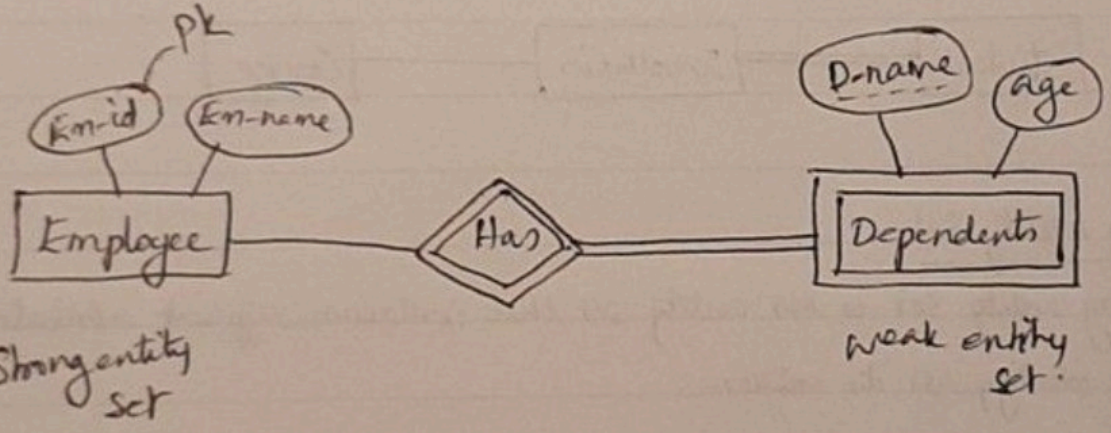
Eg:-



Weak entity set

- A weak entity set is an entity set that doesnot contain sufficient attributes to uniquely identify its entities.
- In other words, a primary key does not exist for a weak entity set.
- However, it contains a partial key called as discriminator.
- Discriminator is represented by underlying with a dashed line.
- The combination of discriminator and primary key of the strong entity set makes it possible to uniquely identify all entities of the weak entity set.

Eg:-



- The double rectangle is used for representing a weak entity set.
- A double diamond symbol is used for representing the relationship that exists bln the strong and weak entity sets and this relationship is known as identifying relationship.
- A double line is used for representing the connection of the weak entity set with the relationship set.
- Total participation always exists in the identifying relationship