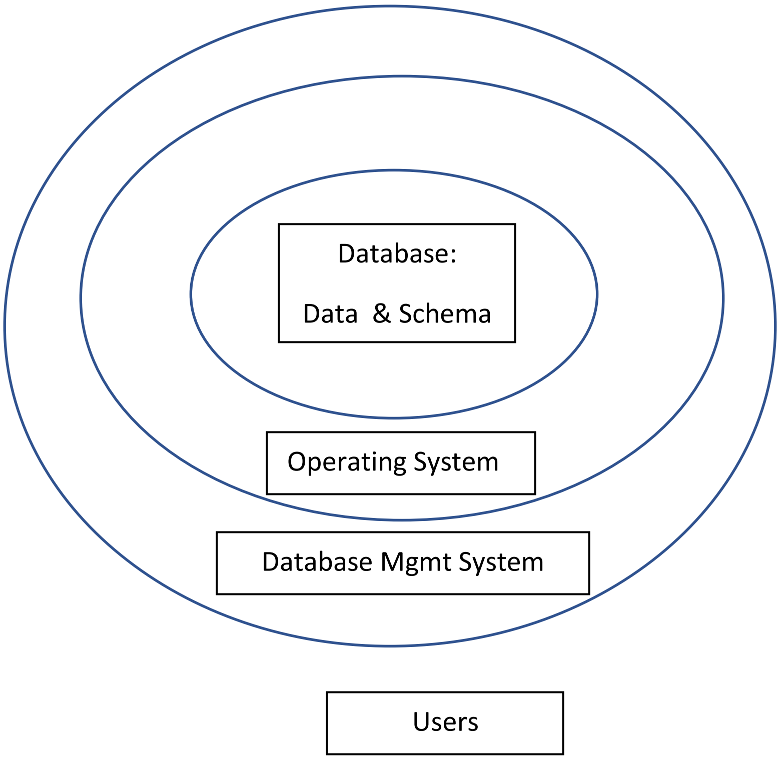
Introduction of DBMS



Data : It is representation of Facts, concepts or instructions in a formalized manner suitable for communication , interpretation of processing by humans or by automatic means.

Database: Collection of logically related data that can be recorded.

Implicit properties of Database:

1. It must represent some real-world aspect. (Source of Information).
2. It must comprise a logically clear and consistent collection of data. (Some related event)
3. The repository of data must be designed , developed and implemented for a specific purpose. (Some intended Users)

Source of Information

Some related event

Intended Users

Database Management System : Refers to set of programs for defining , creation , maintenance and manipulation of a database.

Major Functions of DBMS:

1. **Defining of Database Schema**→defining structure, relationship , integrity constraints.
2. **Manipulation of the Database**→ Insertion, deletion , updating , selection.
3. **Sharing of a Database**→ Concurrent access of shared items by multiusers.
4. **Protection of a Database** →Unauthorized Access.
5. **Database Recovery** →In case of system Failure.

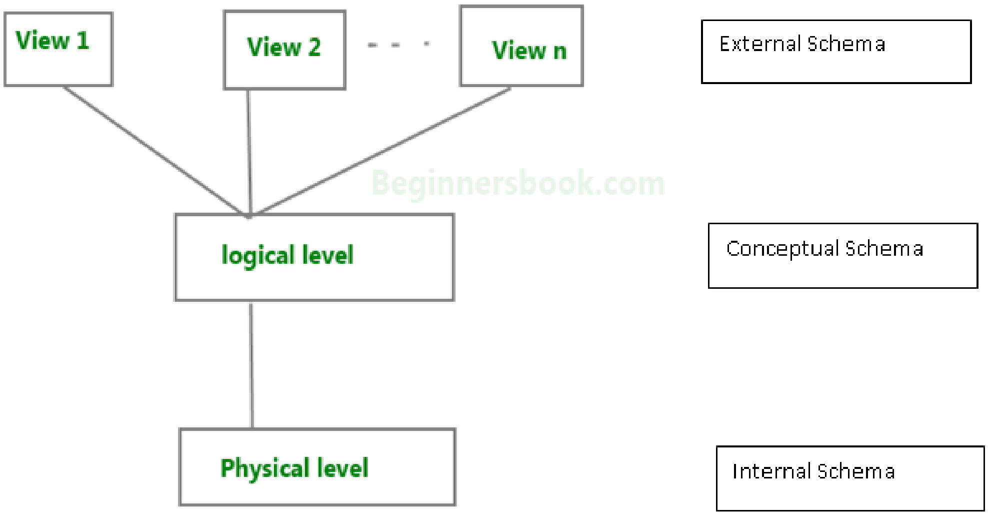
File Processing System:

A file processing system is a collection of programs that store and manage files in computer hard-disk. Such system needed a dedicated set of application program to add information to files, to extract information from the files and to update the existing information.

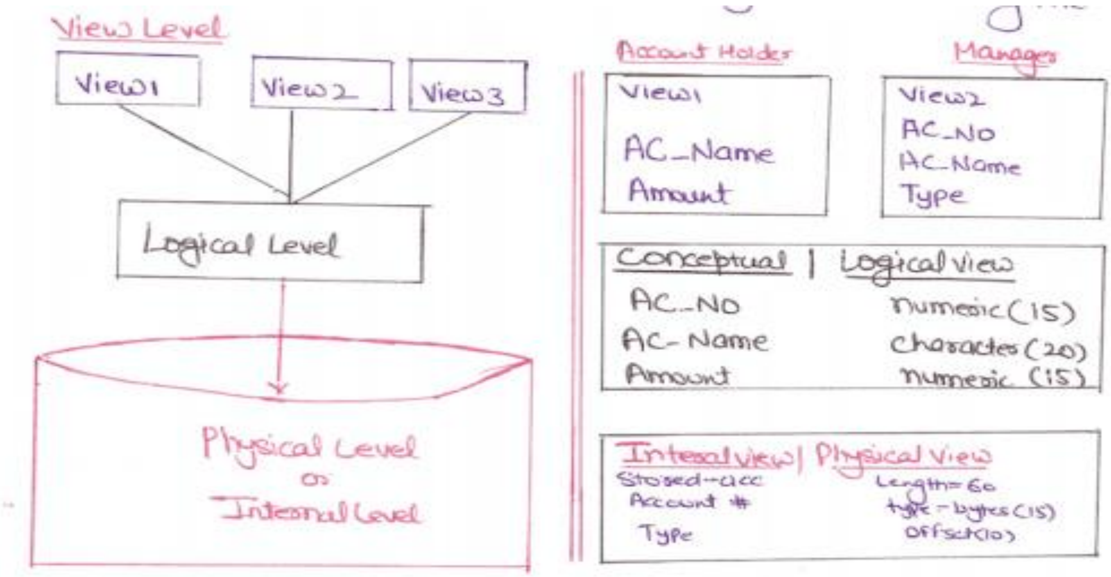
Limitation of FPS:

1. Data Redundancy and inconsistency
2. Difficulty in accessing of data
3. Data Isolation
4. Difficulty in enforcing Integrity Constraints
5. Atomicity problem
6. Difficulty in Concurrency control
7. Security Problem.

View of Data:



Three Levels of data abstraction

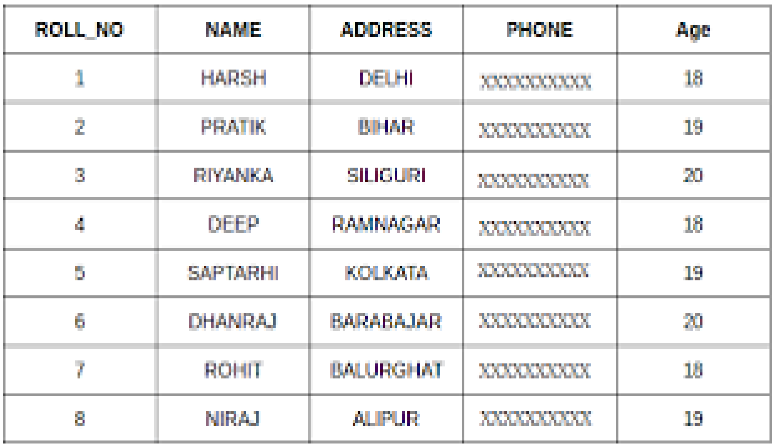


Database Independence:

- Physical Data Independence
- Logical Data Independence

Instance and Schema:

Student



ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

Instance :The data stored in database at a particular moment of time is called instance of database.

Schema : Overall design of the database. A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Features of Database Management System:

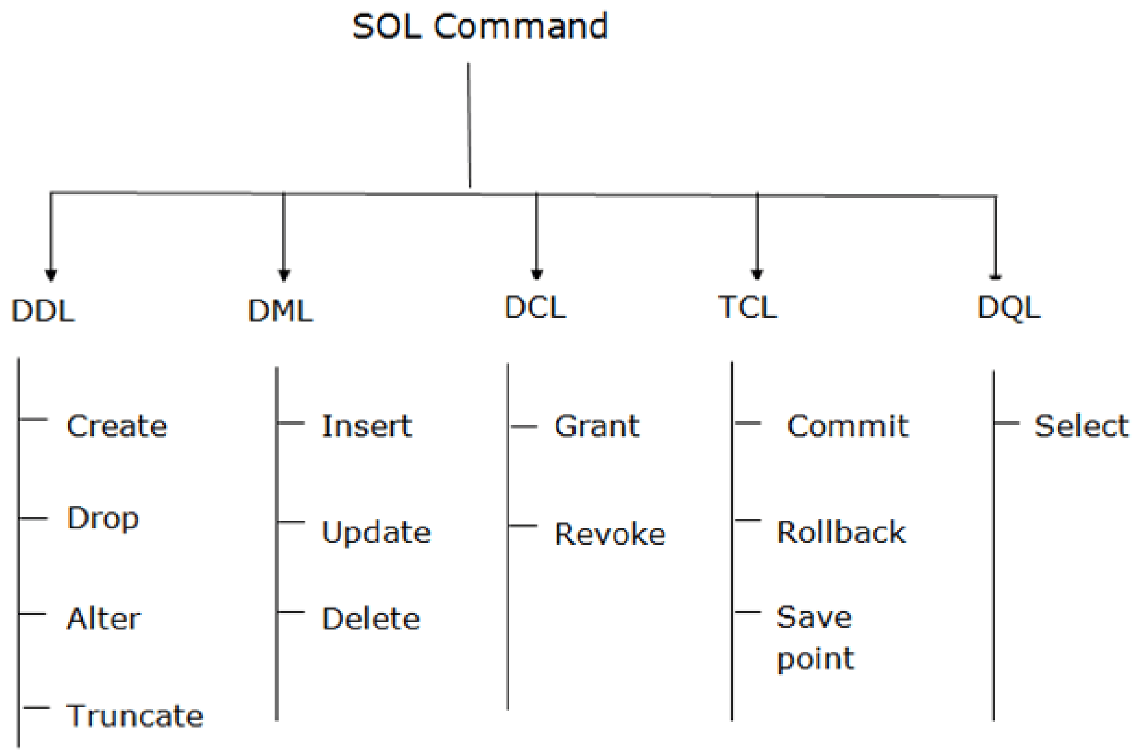
Physical Level:

- **Data files:** Stored in the database itself.
- **Data dictionary:** Stores metadata about the structure of the database.
- **Indices:** Provide fast access to data items.

Logical Level/ Storage Management:

- **File manager:** Manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- **Buffer manager:** It is responsible for retrieving data from disk storage into main memory. It enables the database to handle data sizes that are much larger than the size of main memory.
- **Disk Space Manager:**Allocate, deallocate , read and write pages through this.
- **Query Evaluation Engine :**
  - Parser
  - Query Optimizer
  - Plan executor
  - Operator Evaluation
- **Transaction& Lock manager:** Ensures that the database remains in a consistent state.
- **Recovery Manager:** Maintains a log of all the transactions being executed, it will enable recovery of the underlying database, in the event of failure.

SQL Commands:



Database Interfaces:

- **Menu-Based Interfaces for Web Clients or Browsing** – These interfaces present the user with lists of options (called menus) that lead the user through the formation of a request.
- **Forms-Based Interfaces** –A forms-based interface displays a form to each user. Users can fill out all of the form entries to insert a new data, or they can fill out only certain entries, in which case the DBMS will redeem same type of data for other remaining entries.
- Graphical User Interface – A GUI typically displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram.

- **Natural language Interfaces** –
  These interfaces accept request written in English or some other language and attempt to understand them. A Natural language interface has its own schema, which is similar to the database conceptual schema as well as a dictionary of important words.
- **Interfaces for parametric Users** – Parametric users like bank tellers , often have a small set of operation that they must have performed repeatedly. Usually a small set of abbreviated commands is included with the goal of minimizing the number of keystrokes required for each request.
- **Interfaces for DBA** –
  Most database system contains privileged commands that can be used only by the DBA's staff. These include commands for creating accounts, setting system parameters, granting account authorization, changing a schema, reorganizing the storage structures of a databases.

Database Users:

- **Naive / Parametric End Users** :Parametric End Users are the unsophisticated who don't have any DBMS knowledge but they frequently use the data base applications in their daily life to get the desired results.For examples, Railway's ticket booking users , Clerks

- **System Analyst :**System Analyst determines the requirement of end users especially naïve users and developers specification of transaction that meets these requirements. They play major role in the database design.
- **Sophisticated Users / Programmers :**Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own data base applications according to their requirement. They don't write the program code but they interact the data base by writing SQL queries directly through the query processor.
- **Casual Users / Temporary Users :**Casual Users are the users who occasionally use/access the data base but each time when they access the data base they require the new information, for example, Middle or higher level manager.
- Database Administrator (DBA) :
    - Schema Definition
    - Storage Structure and Access Management
    - Availability of DBMS
    - System Security (Grant/Revoke)
    - Physical Security
    - Periodic backup
    - Immediate Recovery
    - System Performance

Advantages of DBMS:

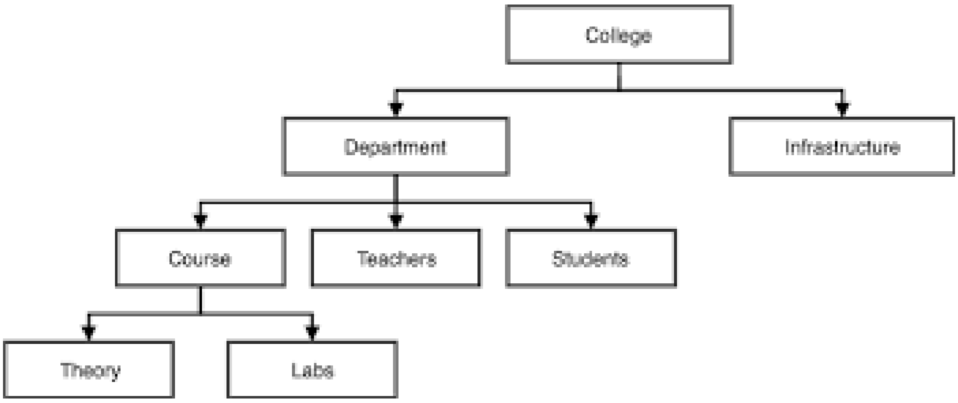
- Improved data sharing
- Improved data security
- Better data integration
- Minimized data inconsistency
- Improved Data Access

Disadvantages of Database Management System:

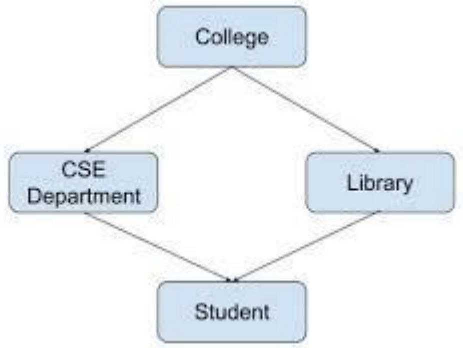
- Increased Costs
- Management complexity
- Frequent upgrade/replacement cycles
- Large memory space required

Database Models:

- Hierarchical Model

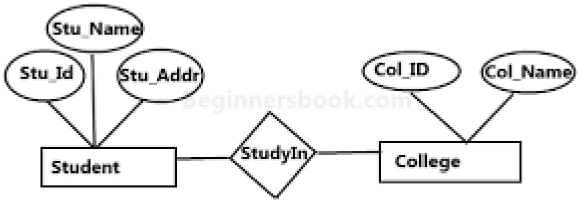


- Network Model



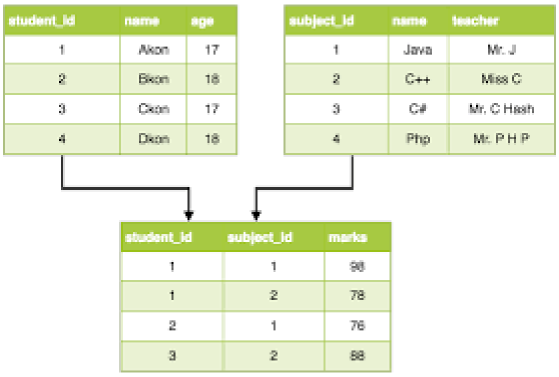
Network Model

- Entity Relationship Database Model



Sample E-R Diagram

- Relationship Database Model



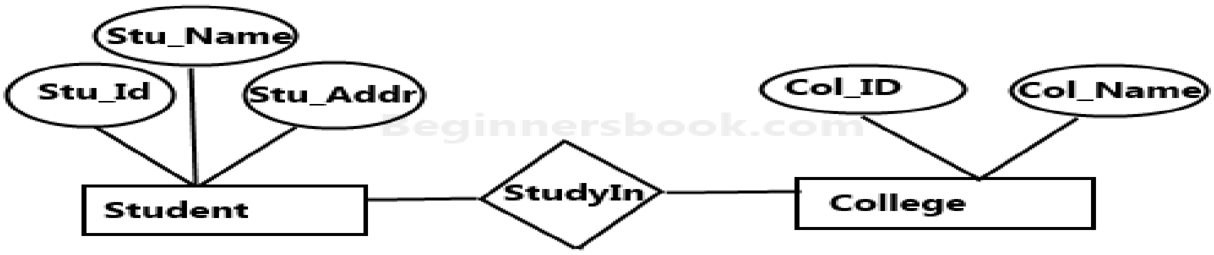
student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Entity Relationship Modelling

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.



Sample E-R Diagram

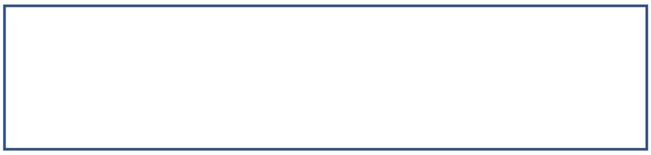
Three main components of ER diagrams are :

- **Entity**
- **Attributes**
- **Relationship**

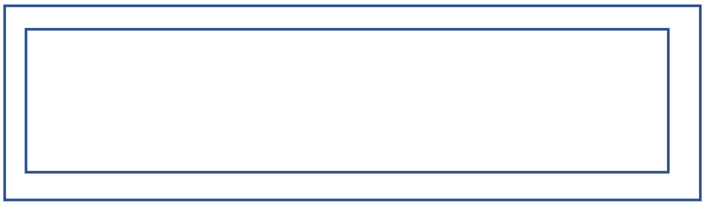
Entity: An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.
For example: In the following ER diagram we have two entities Student and College and these two entities.

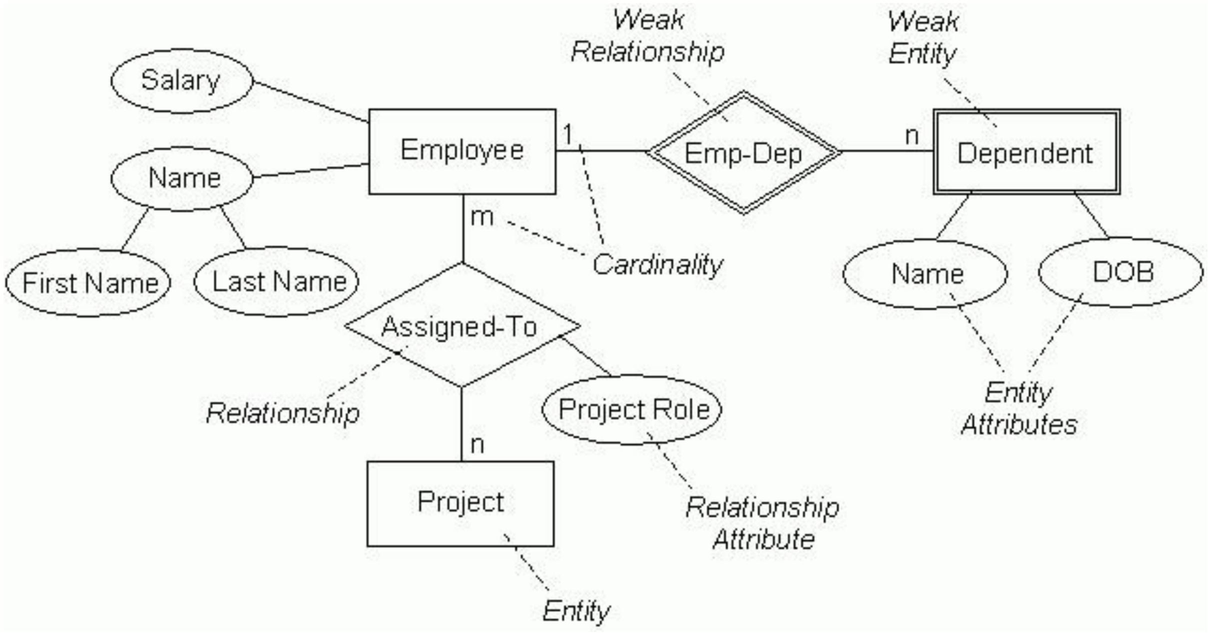
Types of Entity:

- Strong Entity

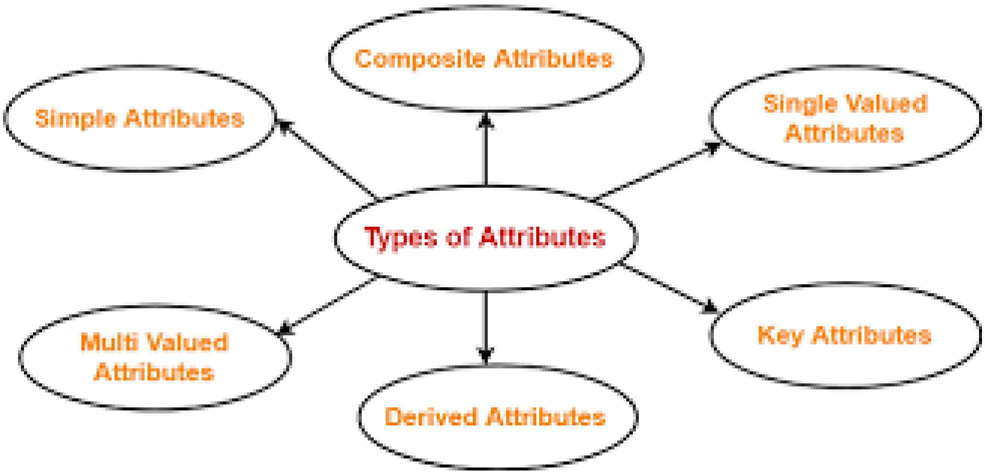


- Weak Entity

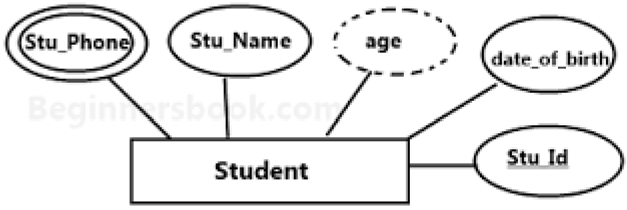




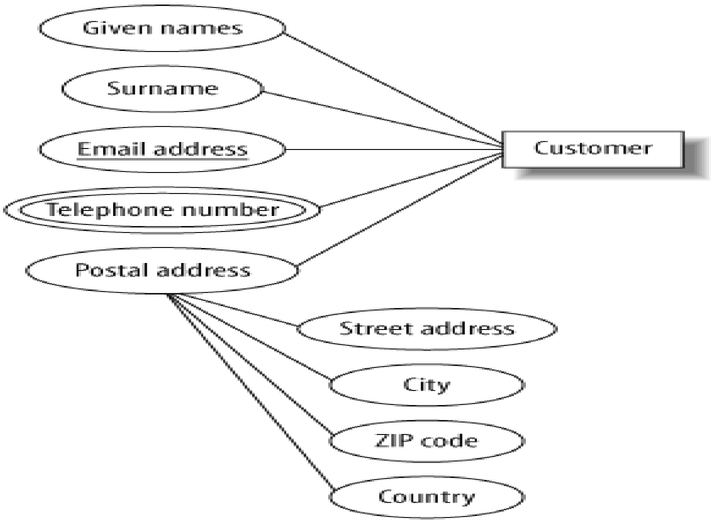
Attributes: An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram.



Example 1:



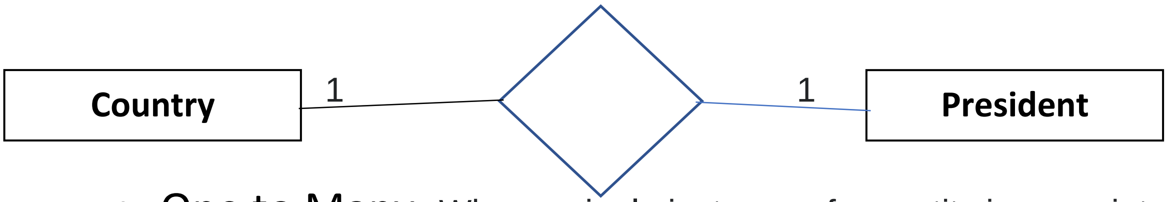
Example 2:



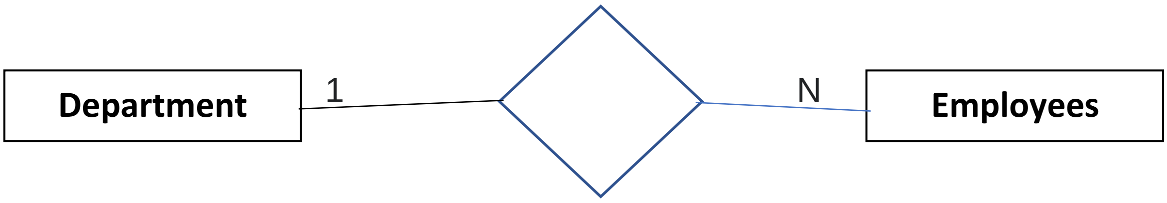
Relationship in ER diagram

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

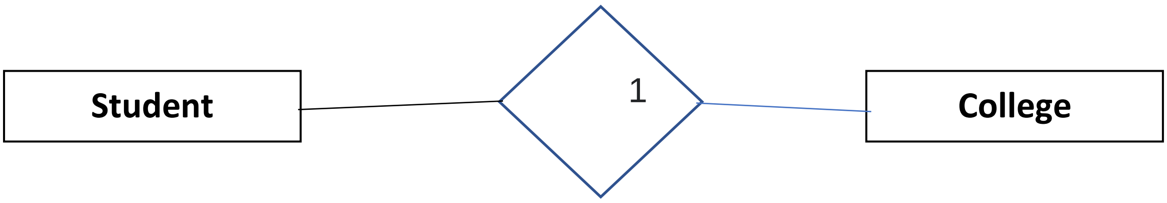
- One to One :When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.



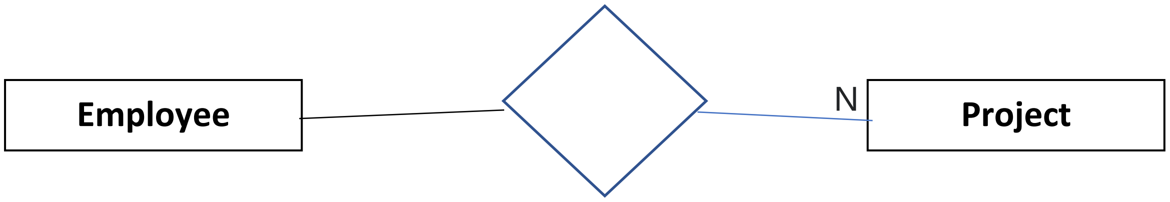
- One to Many: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.



- Many to One :When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship.

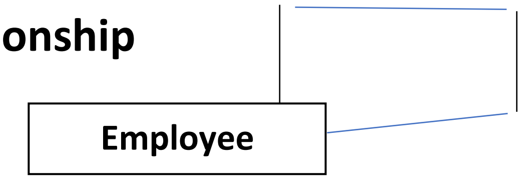


- Many to Many: When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.

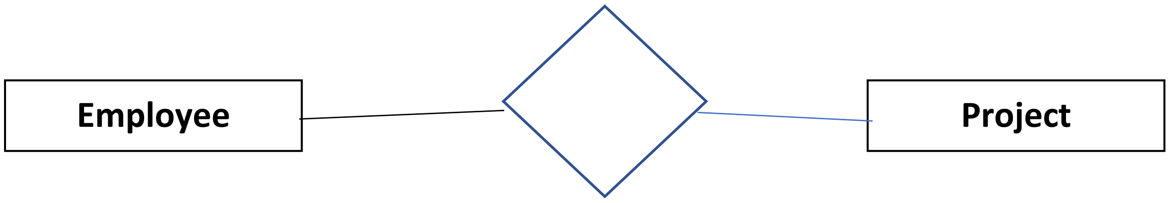


Degree of Relationship:

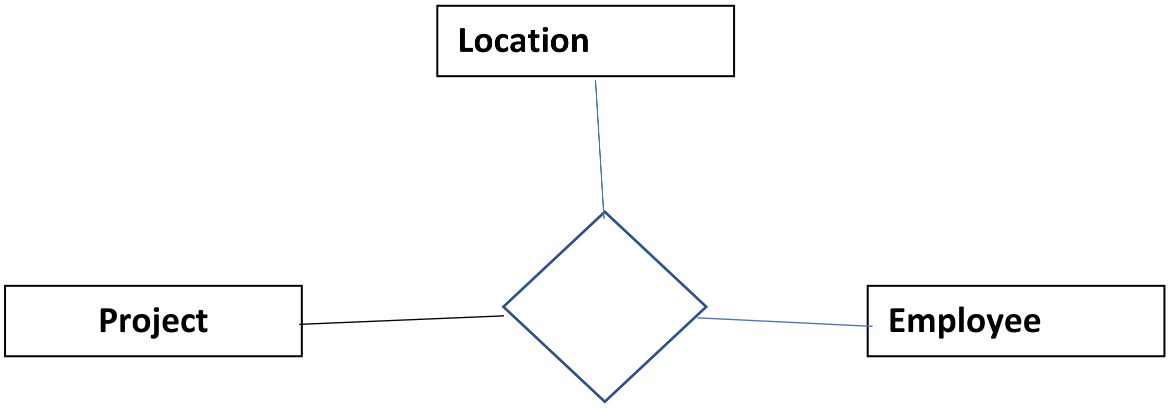
- **Unary Relationship**



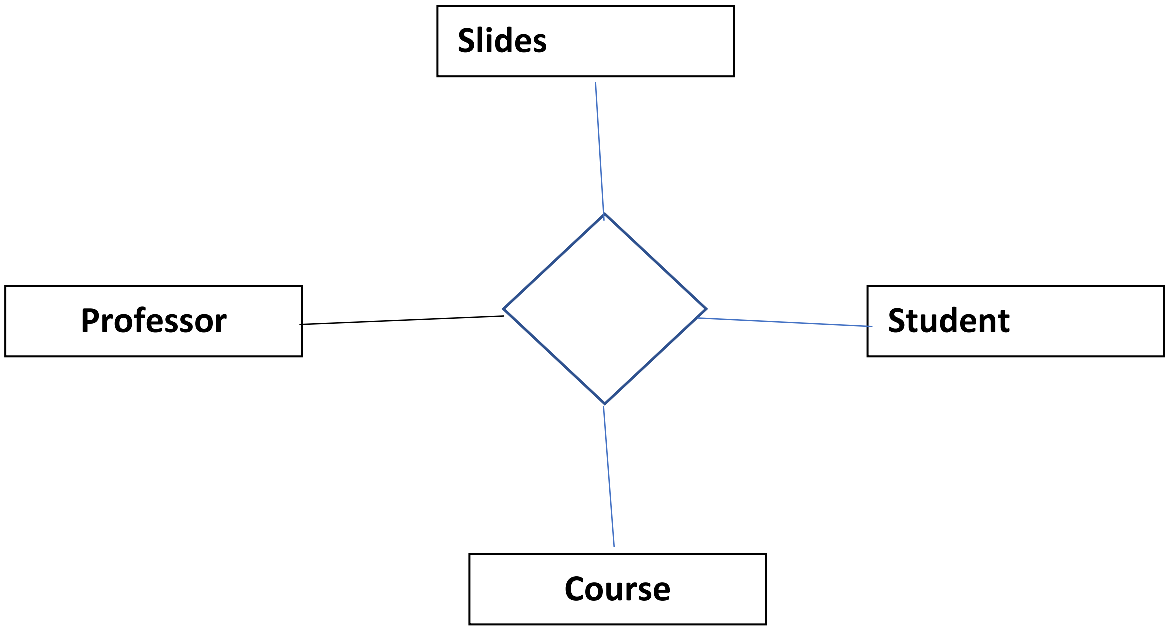
- **Binary Relationship**



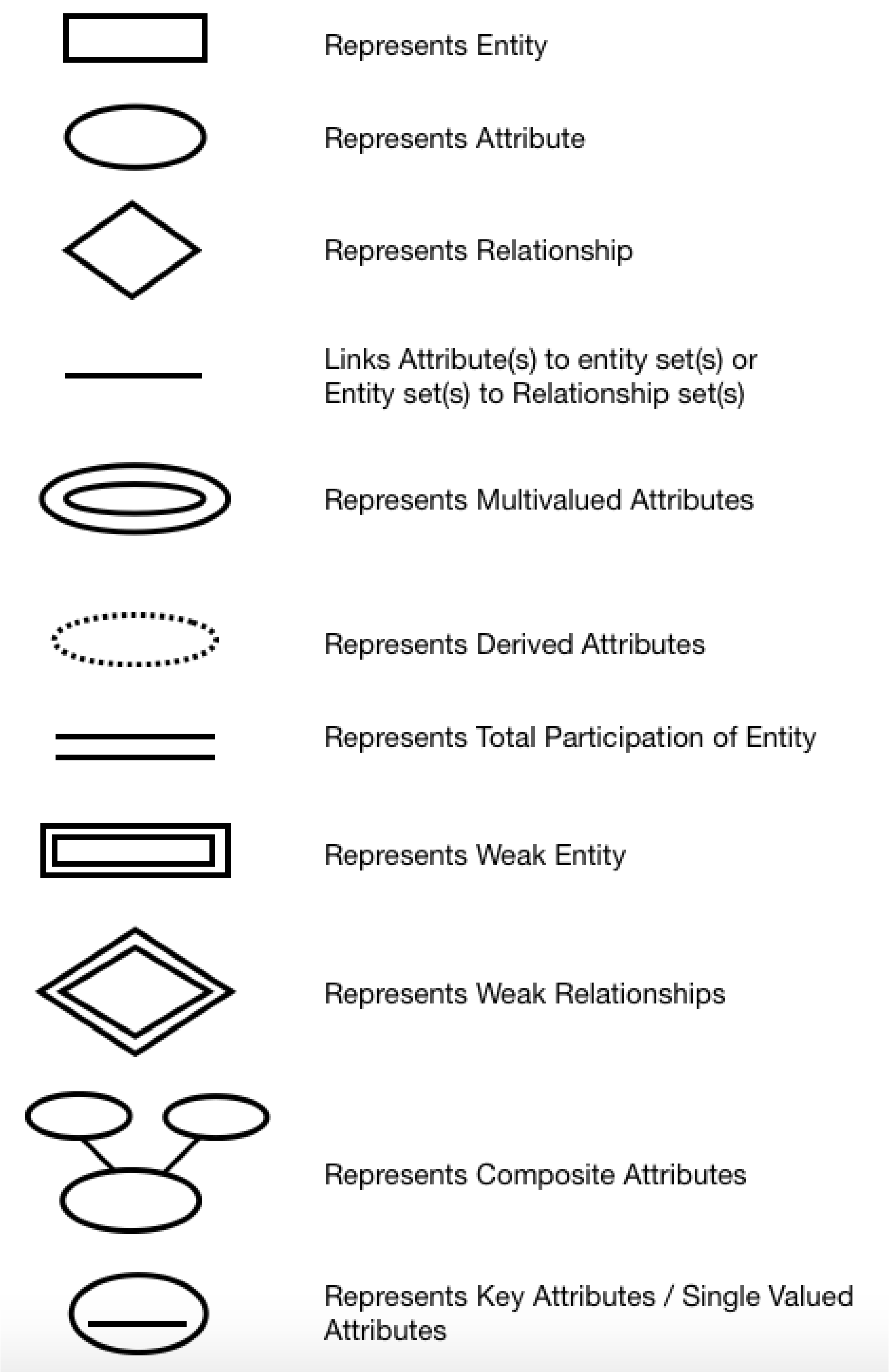
- **Ternary Relationship**



- **Quaternary Relationship**



Summary Of Symbols of ER diagram:



Represents Entity

Represents Attribute

Represents Relationship

Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)

Represents Multivalued Attributes

Represents Derived Attributes

Represents Total Participation of Entity

Represents Weak Entity

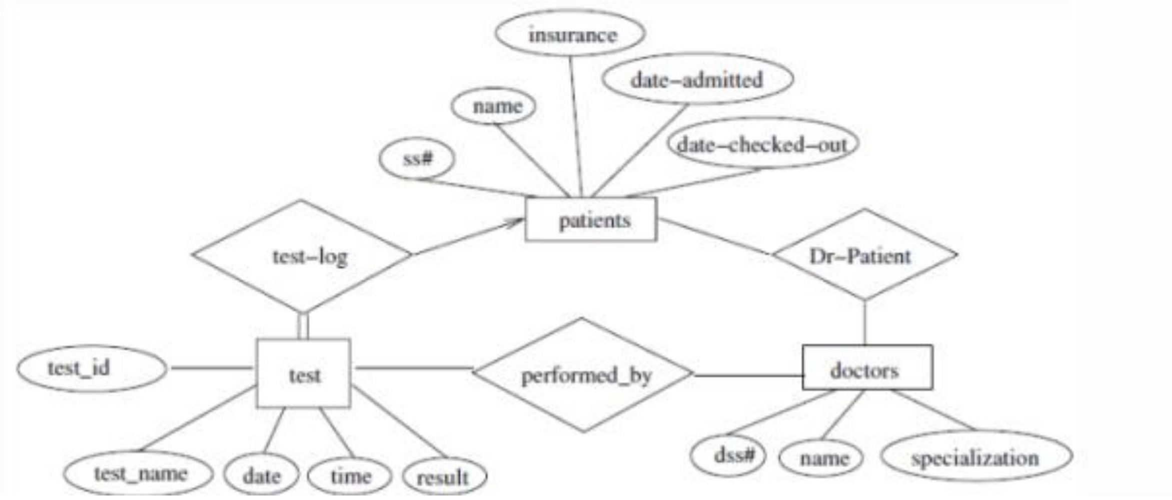
Represents Weak Relationships

Represents Composite Attributes

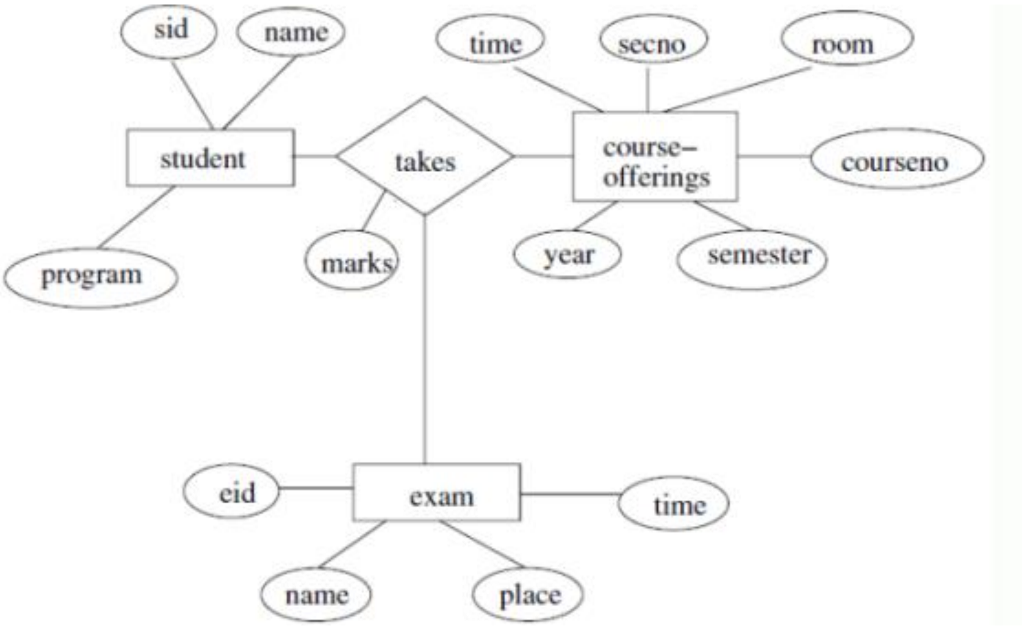
Represents Key Attributes / Single Valued Attributes

ER examples:

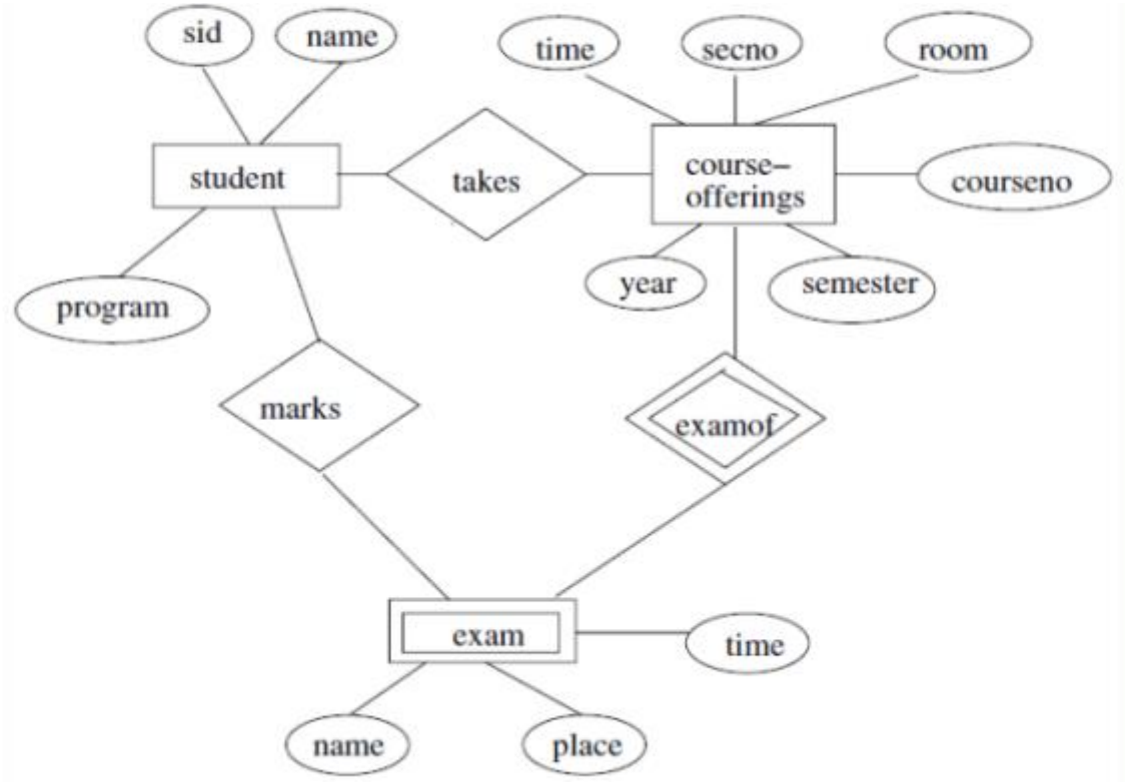
- **Construct an E-R diagram for a hospital with**
  - **a set of patients and**
  - **a set of medical doctors.**
  - **Associate with each patient a log of the various tests and examinations conducted**



- **Consider a database used to record the marks that students get in different exams of different course offerings.**
  - **a) Construct an E-R diagram that models**
    - **exams as entities, and**
      - **uses a ternary relationship, for the above database.**

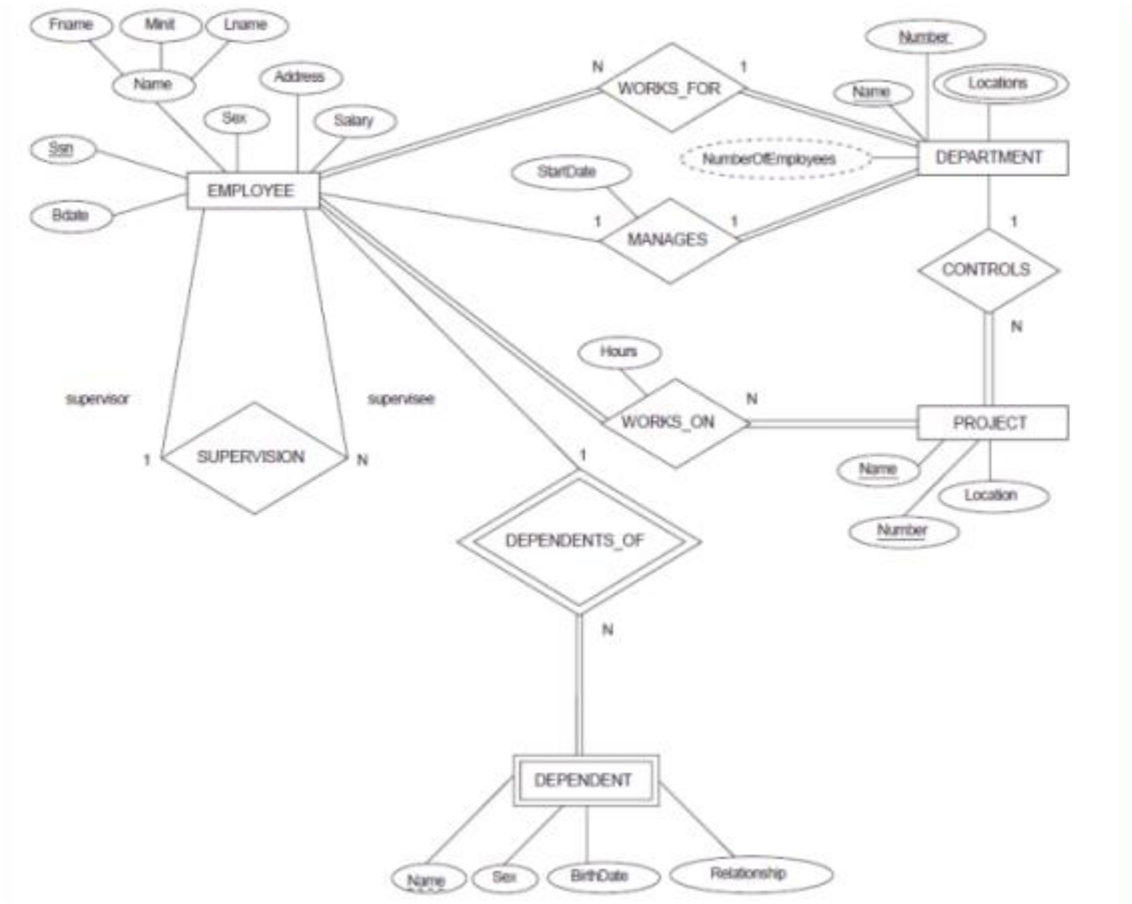


- **b) Construct an alternative E-R diagram that uses only a binary relationship between students and course-offerings.**
  Make sure that only one relationship exists between a particular student and course-offering pair, yet you can represent the marks that a student gets in different exams of a course offering.



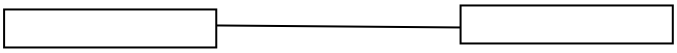
- **Construct an ER Diagram for Company having following details :**
  - **Company organized into DEPARTMENT.**
  - **Each department has unique name• and**
  - **a particular employee who manages the department.**
  - **Start date for the manager is recorded.**
  - **Department may have several locations.**
  - **A department controls a number of PROJECT.**
  - **Projects have a unique name,• number and a single location.**
  - **Company's EMPLOYEE name, ssno, address, salary, sex and birth date• are recorded.**
  - **An employee is assigned to one department, but may work for several projects (not necessarily controlled by her dept).**
  - **Number of hours/week an employee works on each project is recorded;**
  - **The immediate supervisor for the employee.**

- Employee’s DEPENDENT are tracked for health insurance purposes (dependent• name, birthdates, relationship to employee).

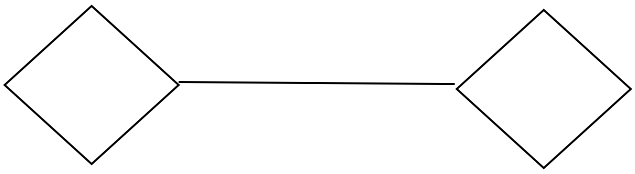


Limitations of ER diagram:

- **Two entities can not be connected without an relationship between them.**



- **ER Diagram can not be used to represent relationship among relationships.**

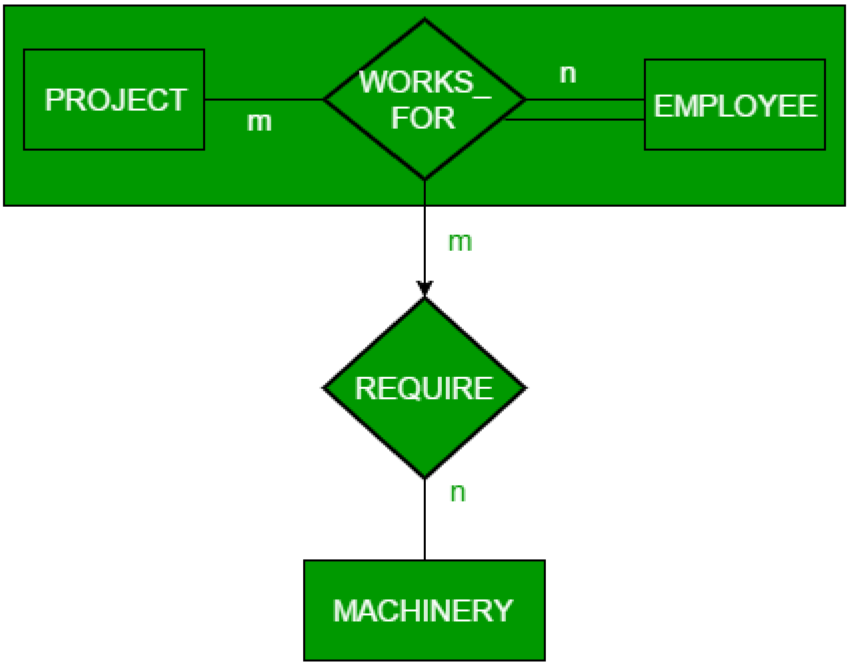


Extended ER Diagrams : EER is a high-level data model that incorporates the extensions to the original ER model. Enhanced ERD are high level models that represent the requirements and complexities of complex database.

EER includes :

- Aggregation
- Generalization
- Specialization

Aggregation:



Aggregation

Generalization & Specialization

- Generalization - the process of defining a general entity type from a collection of specialized entity types.
- Specialization - the opposite of generalization, since it defines subtypes of the supertype and determines the relationship between the two.

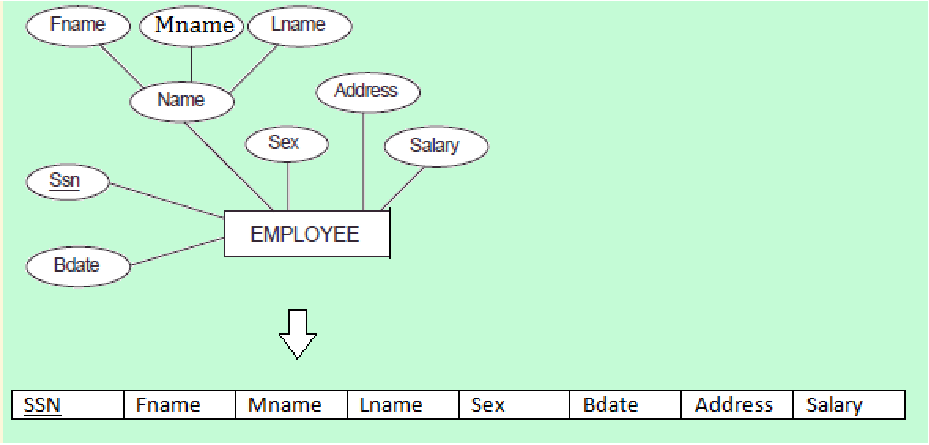
Conversion of er model to relational model – Minimisation of ER Diagram – (7-STEPS)

To convert the er model to relational model there are 7 Steps to be followed, which are

1. **Conversion of Strong Entities –**
2. **Conversion of Weak Entities**
3. **Conversion of one to one Relationships**
4. **Conversion of One to Many Relationships**
5. **Conversion of Many to Many Relationships**
6. **Conversion of n-ary Relationships**
7. **Conversion of Multivalued Attribute**

1. Conversion of Strong Entities –

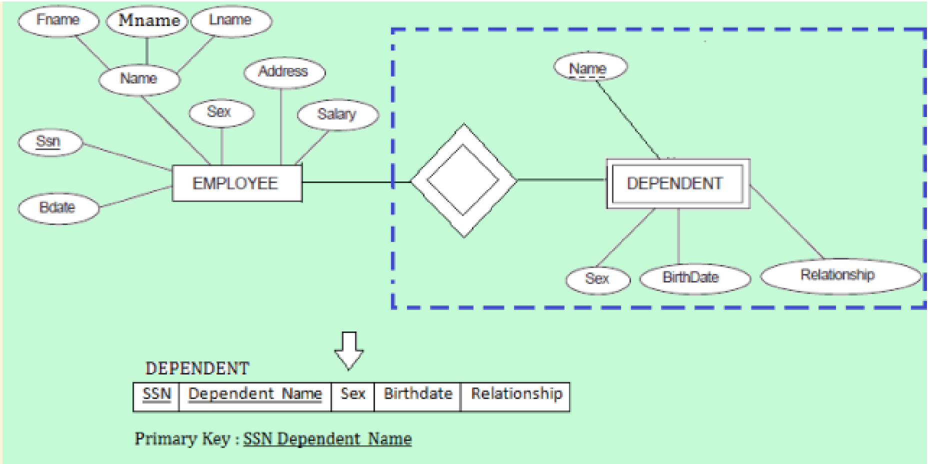
- For each strong entity in ERD, create a separate table with the same name.
- Create all simple Attributes
- Break the Composite attributes into simple attributes and create them.
- Choose a Primary Key for the table.



SSN	Fname	Mname	Lname	Sex	Bdate	Address	Salary

2. Conversion of Weak Entities –

- For each weak entity, create a separate table with the same name.
- Include Primary Key of the strong entity as a foreign key in the table.
- Select the Primary Key attributes of strong entity and the partial Key attribute of the weak entity, and declare them as primary key



DEPENDENT

SSN	Dependent_Name	Sex	Birthdate	Relationship

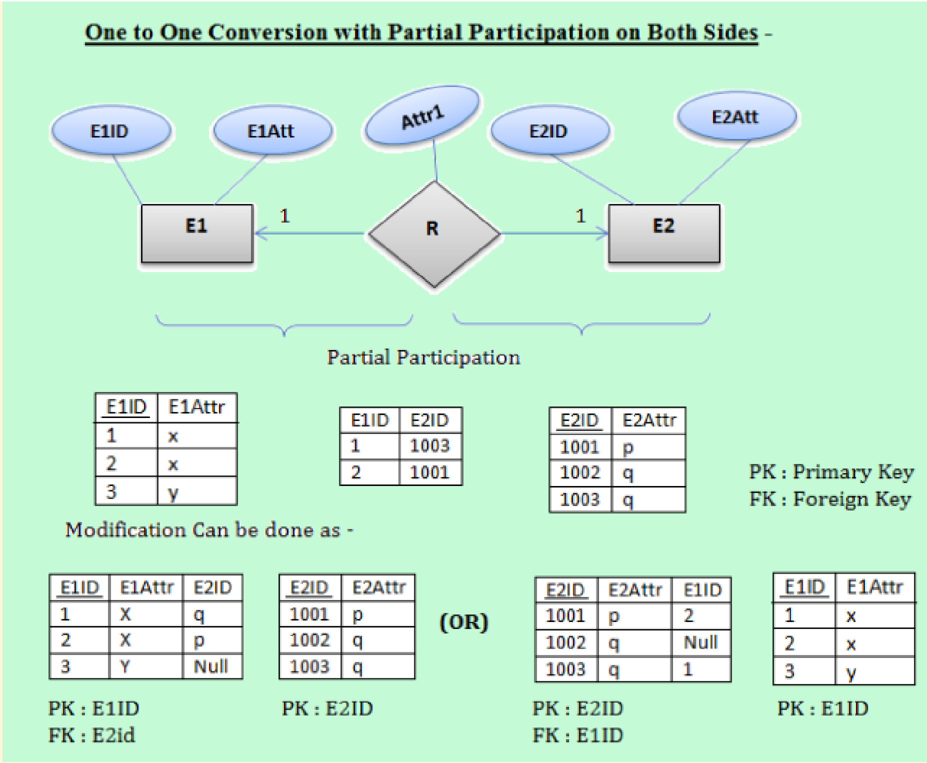
Primary Key : SSN Dependent_Name

3. Conversion of One to One Relationships –

There are two possible approaches on the basis of Participation Constraints –

- Partial Participation on Both Sides –

For each One to One Cardinality between E1 and E2 with partial participation on both sides, modify either E1 or E2 to include the primary key of other table as a foreign key.
So, 1:1 cardinality with partial participation on both sides can be minimised into two relations only.



One to One Conversion with Partial Participation on Both Sides -

E1ID	E1Attr
1	x
2	x
3	y

E1ID	E2ID
1	1003
2	1001

E2ID	E2Attr
1001	p
1002	q
1003	q

PK : Primary Key
FK : Foreign Key

Modification Can be done as -

E1ID	E1Attr	E2ID
1	X	q
2	X	p
3	Y	Null

PK : E1ID
FK : E2id

E2ID	E2Attr
1001	p
1002	q
1003	q

PK : E2ID

(OR)

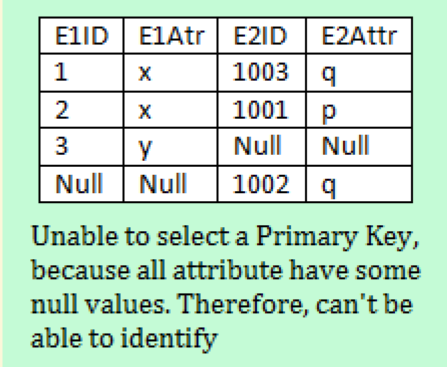
E2ID	E2Attr	E1ID
1001	p	2
1002	q	Null
1003	q	1

PK : E2ID
FK : E1ID

E1ID	E1Attr
1	x
2	x
3	y

PK : E1ID

If we try to minimize the above ERD in a single table, i.e. E1RE2, then it contains too many NULL values, and therefore, we are not be able to select a primary key. For example,

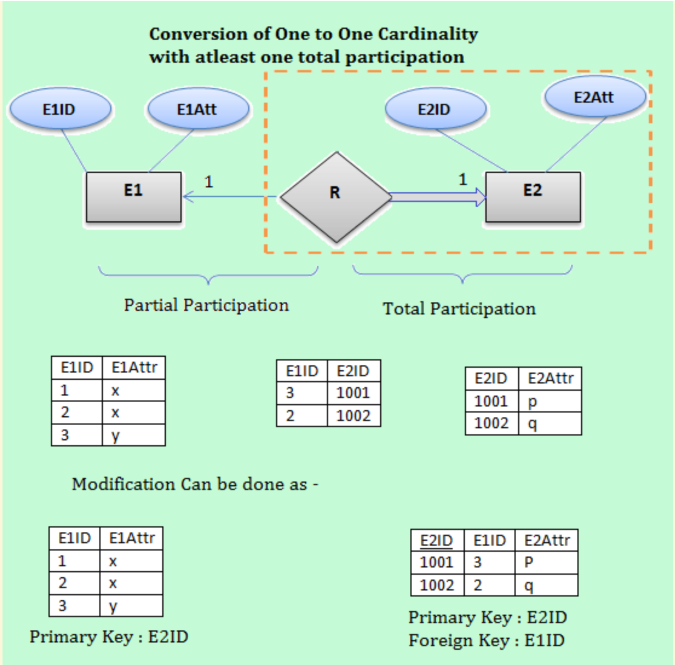


E1ID	E1Atr	E2ID	E2Attr
1	x	1003	q
2	x	1001	p
3	y	Null	Null
Null	Null	1002	q

Unable to select a Primary Key, because all attribute have some null values. Therefore, can't be able to identify

- Cardinality with atleast one Total Participation – For each One to One Cardinality between E1 and E2 with atleast one total participation, modification is done only on total participation side.
  So, One to One Cardinality with atleast one Total

Participation can be minimized into a single relation.



Conversion of One to One Cardinality with atleast one total participation

E1ID	E1Attr
1	x
2	x
3	y

E1ID	E2ID
3	1001
2	1002

E2ID	E2Attr
1001	p
1002	q

Modification Can be done as -

E1ID	E1Attr
1	x
2	x
3	y

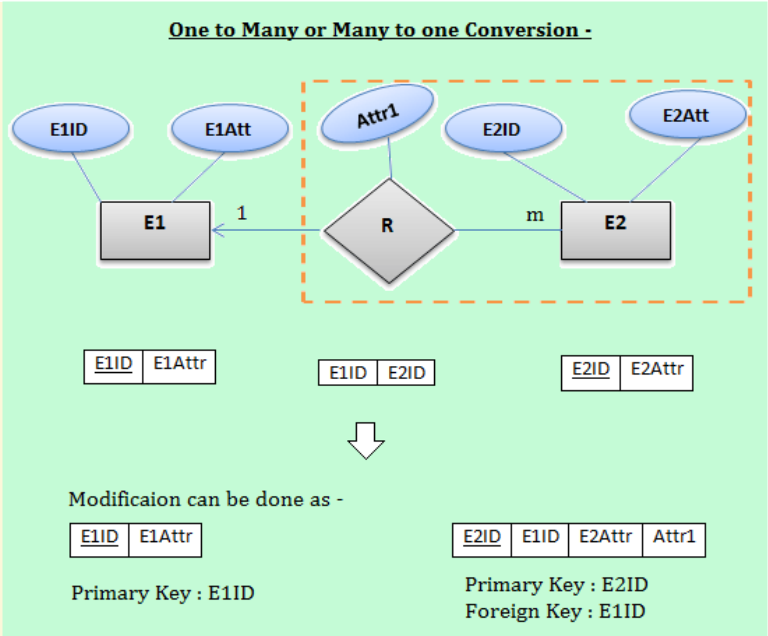
Primary Key : E2ID

E2ID	E1ID	E2Attr
1001	3	P
1002	2	q

Primary Key : E2ID
Foreign Key : E1ID

4. Conversion of One to Many or Many to One Relationship –

- For each one to many relationship between E1 and E2, modify many side relation to include from one side as a Foreign Key.



One to Many or Many to one Conversion -

E1ID	E1Attr

E1ID	E2ID

E2ID	E2Attr

⇩

Modificaion can be done as -

E1ID	E1Attr

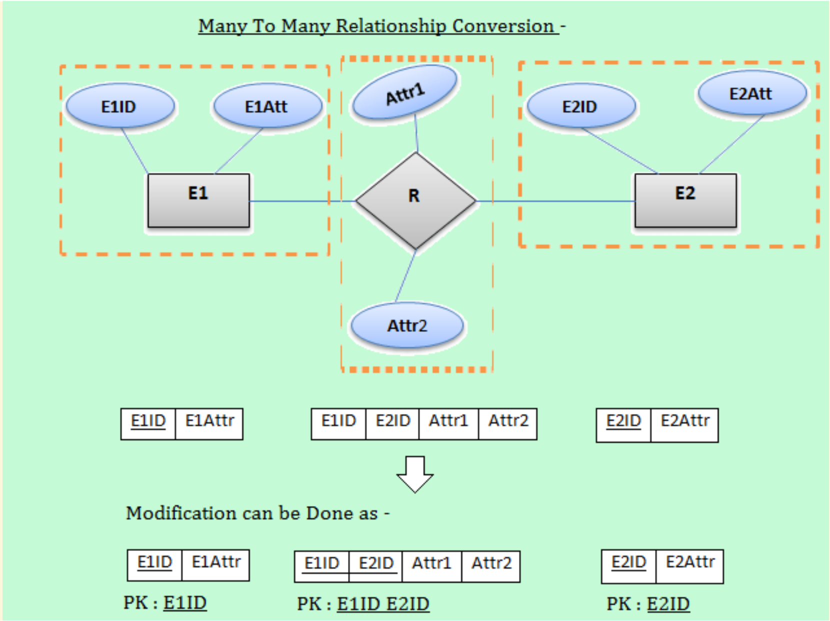
Primary Key : E1ID

E2ID	E1ID	E2Attr	Attr1

Primary Key : E2ID
Foreign Key : E1ID

5. Conversion of Many to Many Relationship –

- For each one to many relationship between E1 and E2, create a separate table and include primary key of both the tables as a Foreign Key.
- If relationship is having one or more attributes, then these must also be included in the table.


Many To Many Relationship Conversion -

E1ID	E1Attr

E1ID	E2ID	Attr1	Attr2

E2ID	E2Attr

Modification can be Done as -

E1ID	E1Attr

PK : E1ID

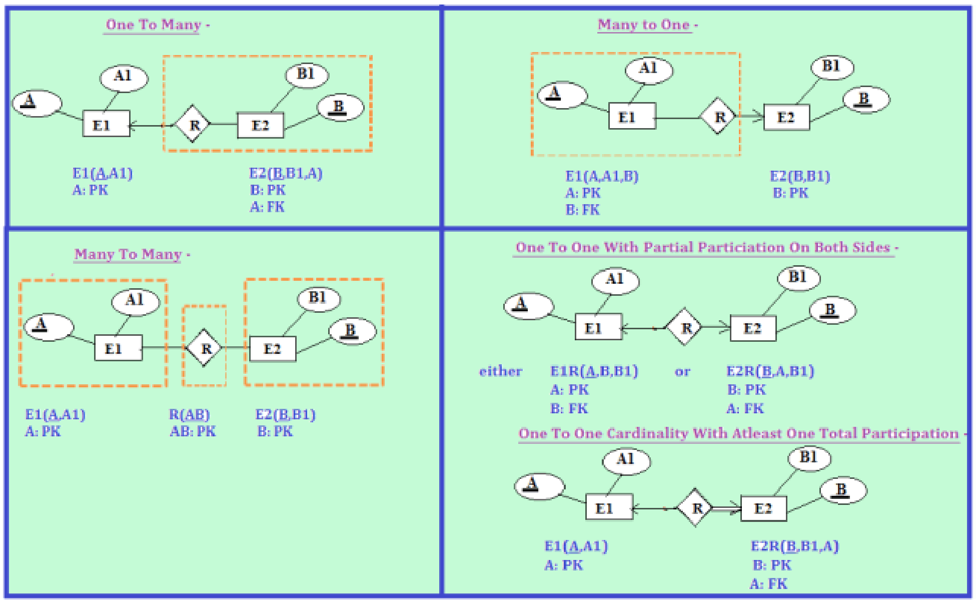
E1ID	E2ID	Attr1	Attr2

PK : E1ID E2ID

E2ID	E2Attr

PK : E2ID

In short, The Conversion of Relationships will be done as –


One To Many -

E1(A,A1)
A: PK

E2(B,B1,A)
B: PK
A: FK

Many to One -

E1(A,A1,B)
A: PK
B: FK

E2(B,B1)
B: PK

Many To Many -

E1(A,A1)
A: PK

R(AB)
AB: PK

E2(B,B1)
B: PK

One To One With Partial Particiation On Both Sides -

either E1R(A,B,B1)
A: PK
B: FK

or E2R(B,A,B1)
B: PK
A: FK

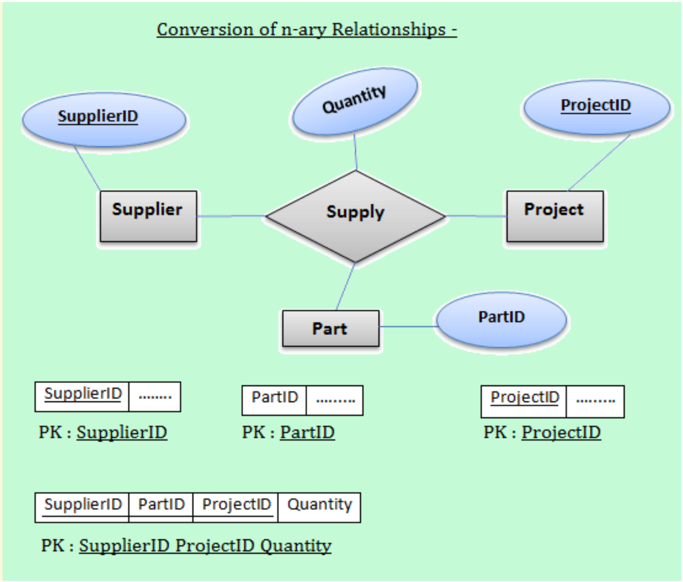
One To One Cardinality With Atleast One Total Participation -

E1(A,A1)
A: PK

E2R(B,B1,A)
B: PK
A: FK

6. Conversion of n-ary Relationship –

- For each n-ary relationship, Create a separate table and include primary keys of all other entities as a foreign key.
- If the relationships has some attributes, then these must also be included in the table.



Conversion of n-ary Relationships -

SupplierID	....

PK : SupplierID

PartID	....

PK : PartID

ProjectID	....

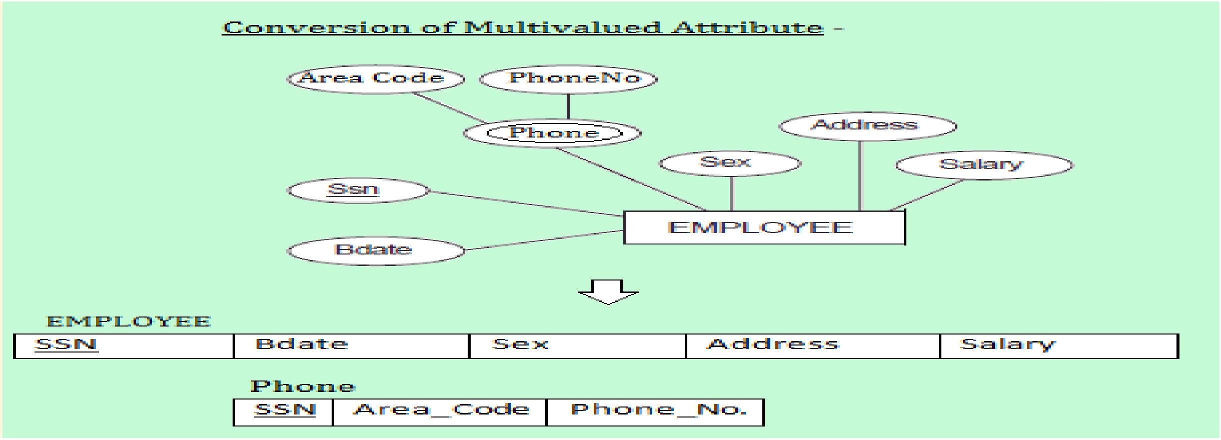
PK : ProjectID

SupplierID	PartID	ProjectID	Quantity

PK : SupplierID ProjectID Quantity

7. Conversion of multivalued Attributes –

- For each multivalued attributes, create a separate table, then include all of its simple attributes.
- Include the primary key of the original table as a foreign key.



Conversion of Multivalued Attribute -

EMPLOYEE

SSN	Bdate	Sex	Address	Salary

Phone

SSN	Area_Code	Phone_No.