Unit 2: Data system Architecture

or

The three level of the Architecture

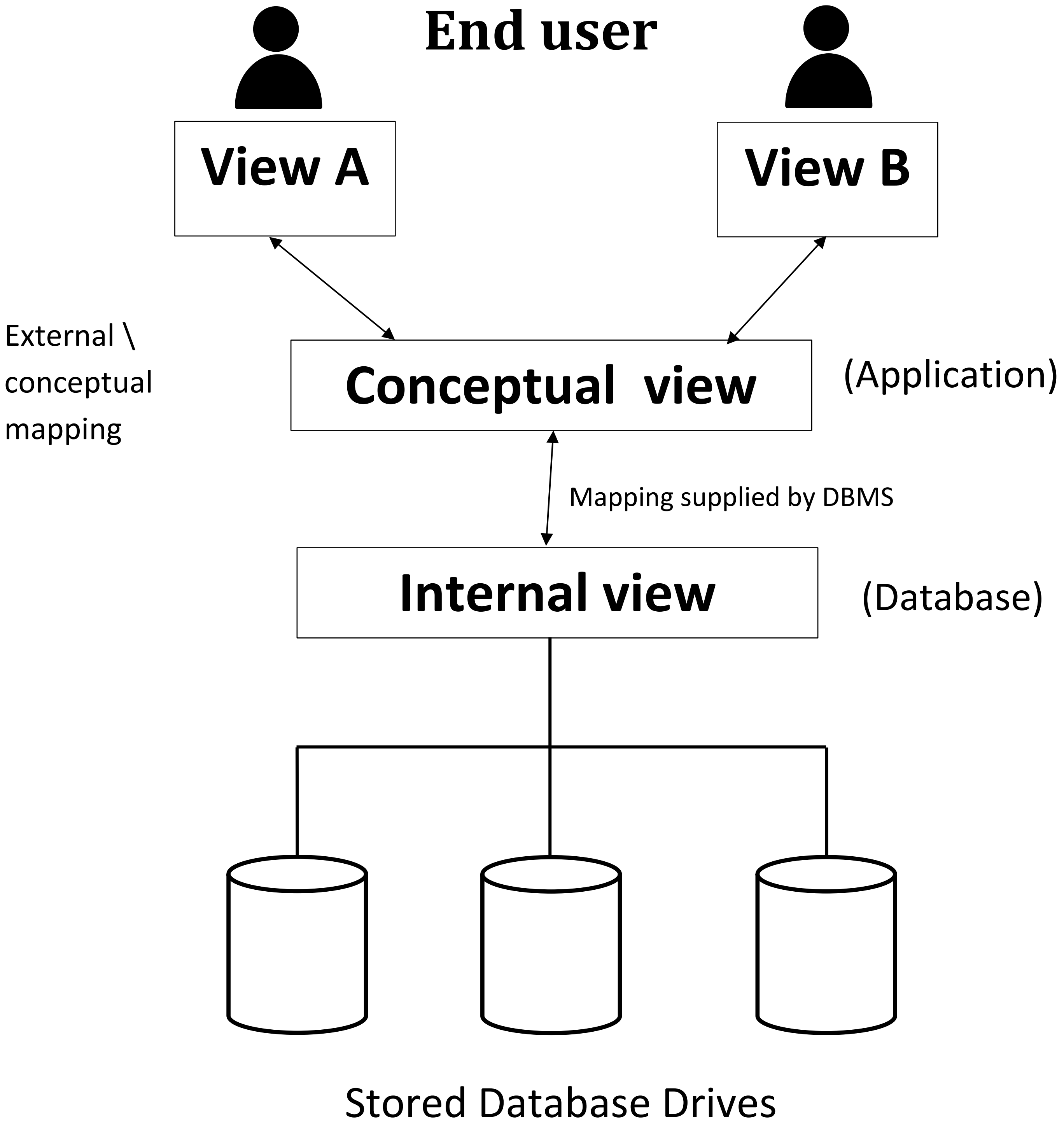
- The database system architecture is also called **ANSI\SPARC** model.
- The Three level architecture, shown in figure is divided into three levels.
  1. External level (view level)
  2. Conceptual level (Logical level)
  3. Internal level (Physical level)
- The view at each of these level is described by scheme.
- A scheme is an outline or plan that described the record & relationship existing in the view.

- **Purpose:-** Proposed to support DBMS characteristics of:
  - Program data independence.
  - Support of multiple views of the data.
- Not explicitly used in commercial DBMS products, but has been useful in explaining database system organization.

1. External view level:-

- The external or user view is the highest level of database abstraction where only those portion of the database of concern to user or application programs are include.
- Any number of user views may exist for a given conceptual view.
- Each external view is described by means of scheme called an external scheme.

- The external level contains the definition the logical records & the relationship in the external view.
- The external level also contains the method of deriving the object n the external view from the object in the external view from the object in the conceptual view.
- The object include entities, relationship & attributes.



2. Conceptual view:- At this level of database all the database entities & relationship among them are included.

- The conceptual view represents the entire database.
- The conceptual view is defined by the conceptual schema.
- It describes all the records & relationship included in the conceptual view & therefore in the database.
- There is only one conceptual schema as per database.
- This schema also contains the method of deriving in the conceptual view from the object in the internal view.

3. Internal view:-

- It is the lowest level of abstraction closest to the physical storage method used.
- It indicates how the data will be stored & describes the data structures and access methods to be used by database.
- The internal view is expressed by the internal schema which contains the definition of the stores record the method of representing the data fields & the access used.

❖ Concept of data models?/ What is data models? Explain the each type of data models or category of data models.

Ans: Definition of data models:-

A data model is a set of concepts to describe the structure of a database.

The operation for manipulating these structures & certain constraints that data base should obey.

❖ Data model structure & constraints:-

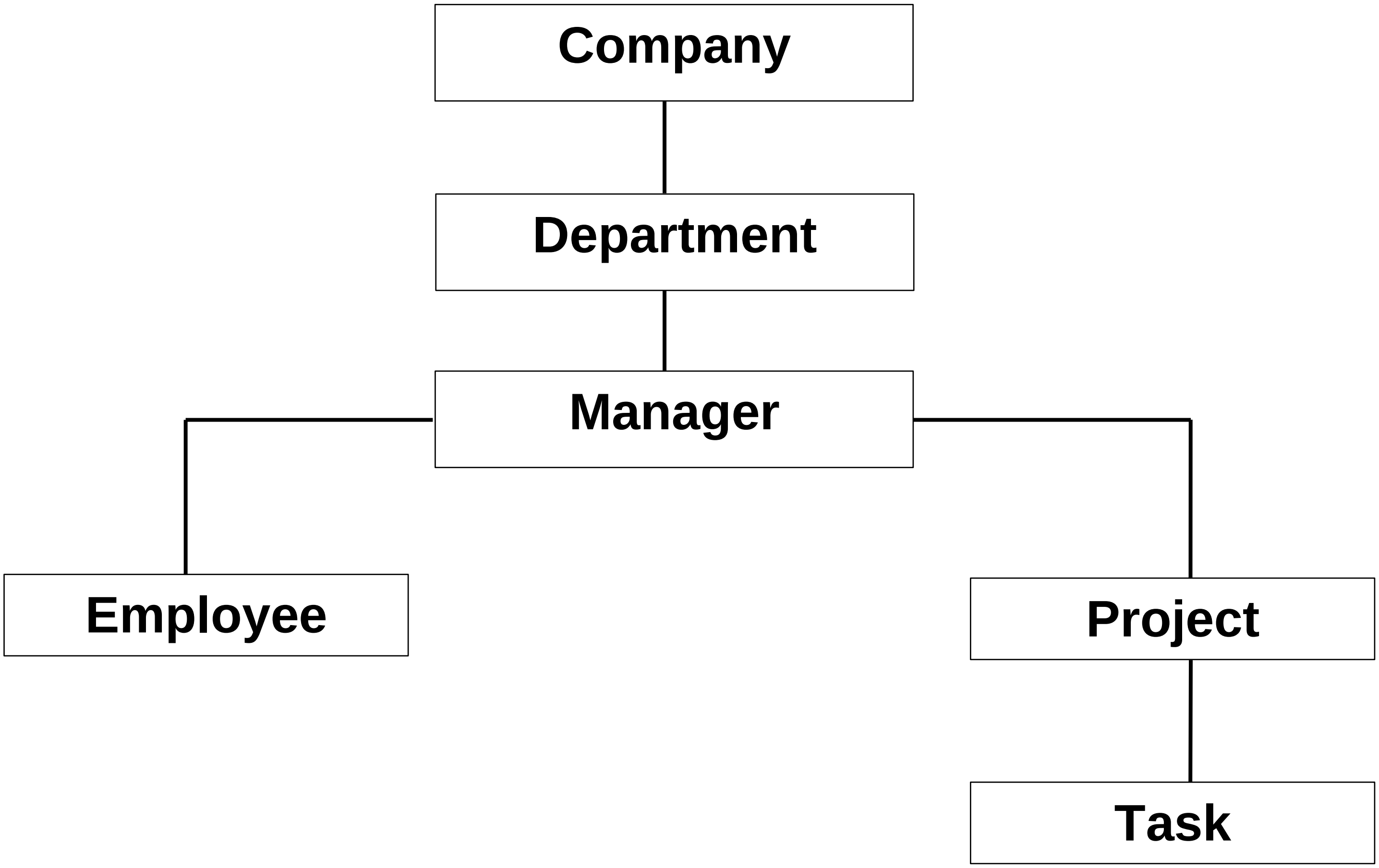
- Construct are used to define the database structure.
- Constructs typically include elements & their datatypes & group of elements (Ex: entity, record, table & relationships as such groups.)
- Constraints specify some restrictions on valid data, these constraints must be enforced at all times.

❖Types of data models:-

1) Hierarchical data model
2) Network data model
3) Relation model
4) The Entity relationship model
5) Object- oriented data model
6) Semi structured data model

1. Hierarchical data model:-

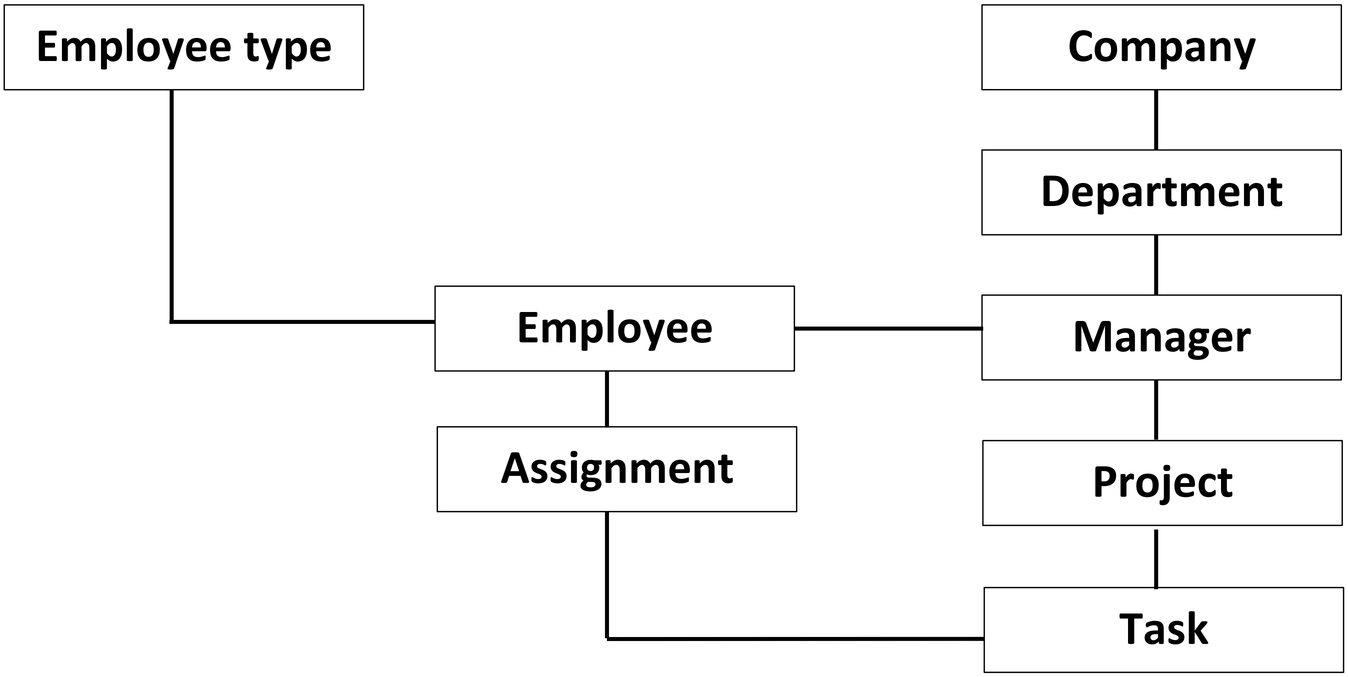
- The hierarchical database model is an inverted tree-like structure.
- The table of this model take on a child parent relationship.
- Child table are completely dependent on parent tables, therefore a child table can exist only if its parent table does.
- It follows that any entris in child table can only exits where corresponding parent entris exist in parent table.
- The result of this structure is that the hierarchical database model supports one-to many relationships.



[Fig: Hierarchical data model]

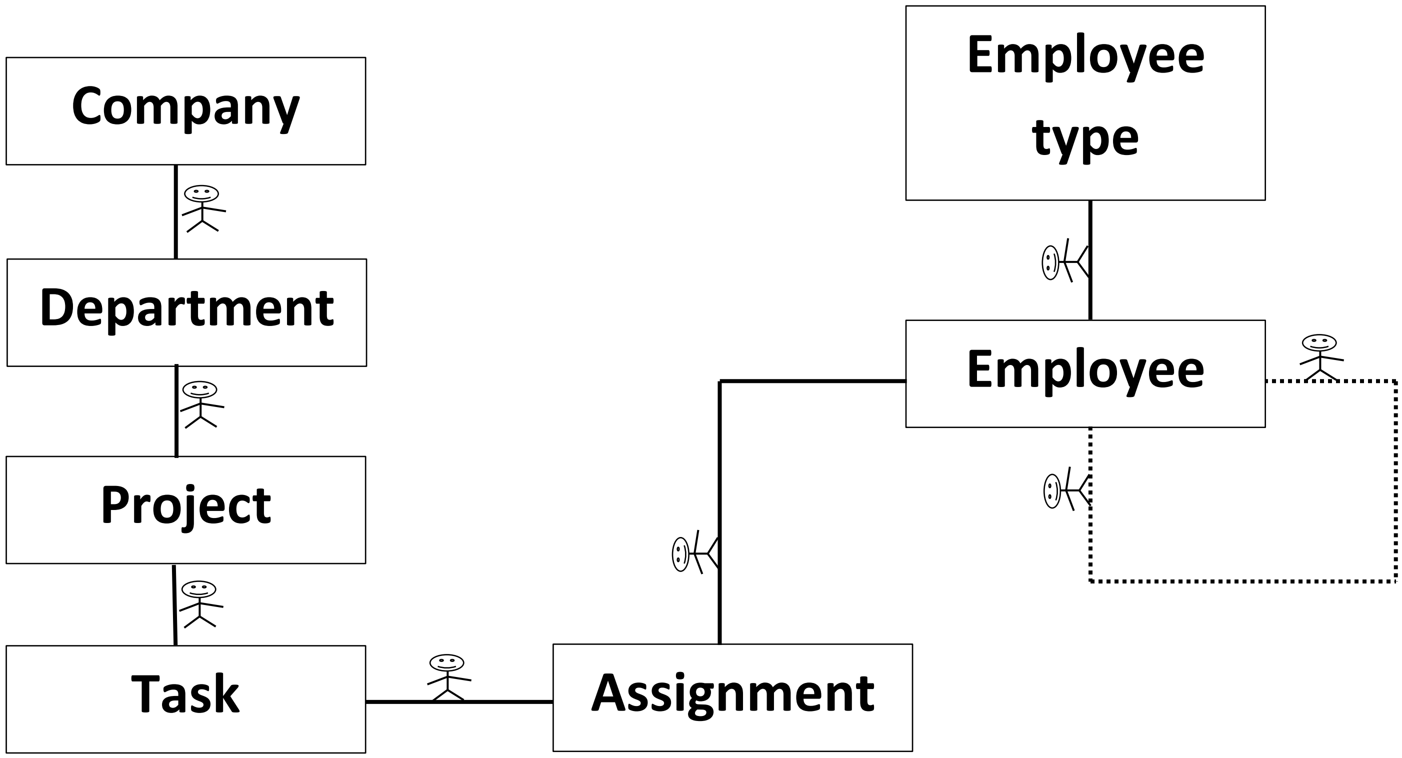
2. Network data model:-

- The network database model essentially a refinement of the hierarchical database model.
- The network model allows child tables to have more than one parent, creating a network like table structure.
- The network model multiple parent tables for each child allows for many-to-many relationships, in addition to one-to-many.



3. Relational data model:-

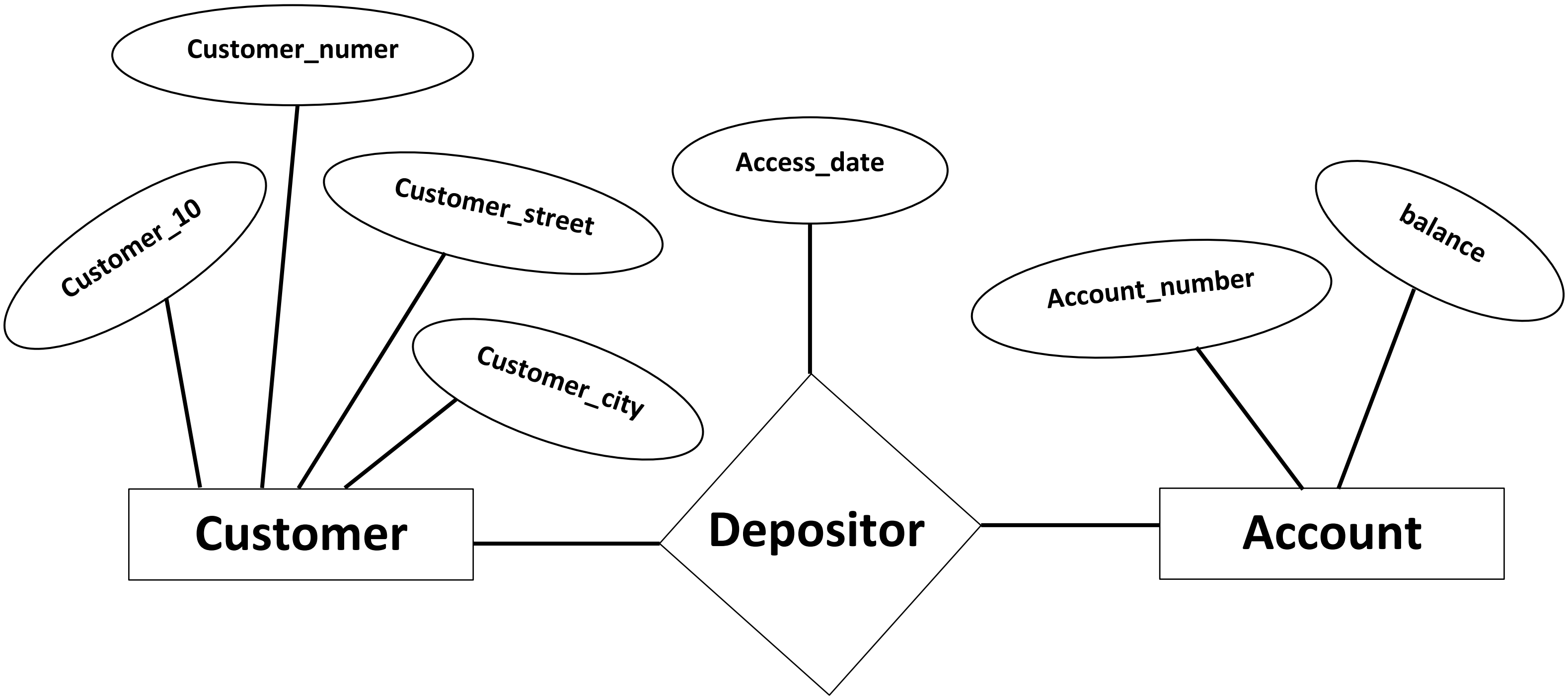
- The relational database model improves on the retriction of a hierarchical structure, not completely abandoning the hierarchy of data.
- Any table can be accessed directly without having access all parent object.
- The track is to know what to look for if you want to find the address of a specific employee, you have to know which employee to look for or you can simply examine all employees.
- You don't have to search the entire hierarchy the company down word , to find single employee.
- Another benefit of the relational database model is thar any tables can be linked together, regardless of there hierarchical position.



[Fig: Relational model]

4. The Entity relationship model:-

- The entity relationship model [E-R] data model is based on a perception of real world thar consist of collection of basic objects called entities & relationship among those objects.



[Fig: E-R Model]

5. Semi structured data model:-

- The semi structured data model permits the specification of data where individual data items of the same type may have different set of attributes.
- This is contrast to the data models mentioned where every data items of a particular type must have the same set of attributes.
- The Extensible Markup Language(XML) is widely used to represent semistrued data.

6. Object base data model:-

❖What is operators in SQL?

- Operator are some character or keyword which is use or an expression to solve any operation.
- **The operators of SQL can be categorized in to 3 types:-**
  1) **Unary operator (+, -, ++, --, !)**
  2) **Binary operator**
     - **Arithmatic Operator**
     - **Relational operator**
     - **Logical operator**
     - **Another relational operator**
  3) **Set operator**
     - **Union**
     - **Union all**
     - **Minus**
     - **Interset**

❖Another relational operator:-

- DISTINCT, ALL, IN, NOT IN, BETWEEN, LIKE, NOT LIKE.
- **Distinct:-** It allow us to retrive unique records of table.

  - Select*distinct*from emp;

- **All:-** It retrive all the information of table, including duplicates records.

  - Select all*from emp;

- **IN:-** It is used to retrive more than one record of table.
  - Select*from emp where eid in(20,70,100);

- **Not IN:-** It is used to retrive unselected records of table.

  - Select*from emp where eid not in (20,70,100)
- **BETWEEN:-** It is used to retrive records of table between two rang.

  - Select*from emp where id between 20 and 40;(only string value)
- **LIKE:-** Like operator is used to retrive a pattern wise information from a table it contains two wild card characters.
- **%-** zero or more character.
- _ only for single character.

  - Select*from emp where ename like ‘r%’;
- **Not like:-** Display unselected records.

  - Select*form emp where ename notlike ‘r%’;

❖ Relationship / Relation:-

- A logical connection b/w different entities is called Relationship.
- The relationship indicates how the entities are connected or related to each other.

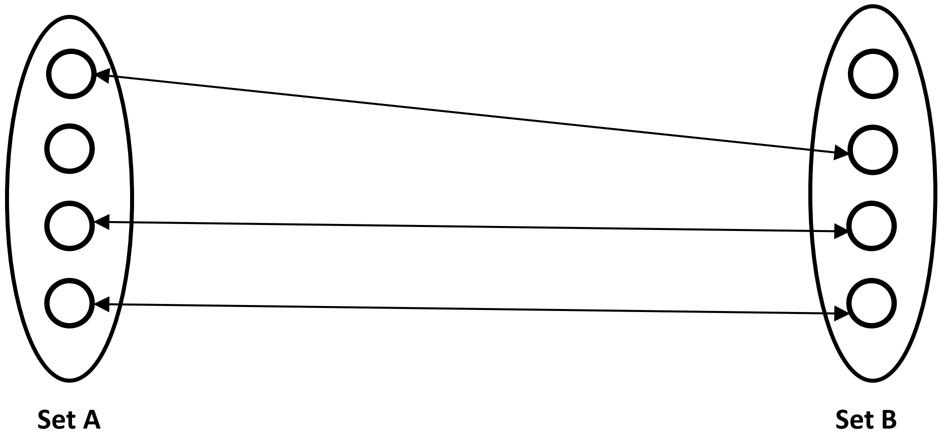
❖ Relationship set:-

- A set of relationship of similar type is called a relationship set.

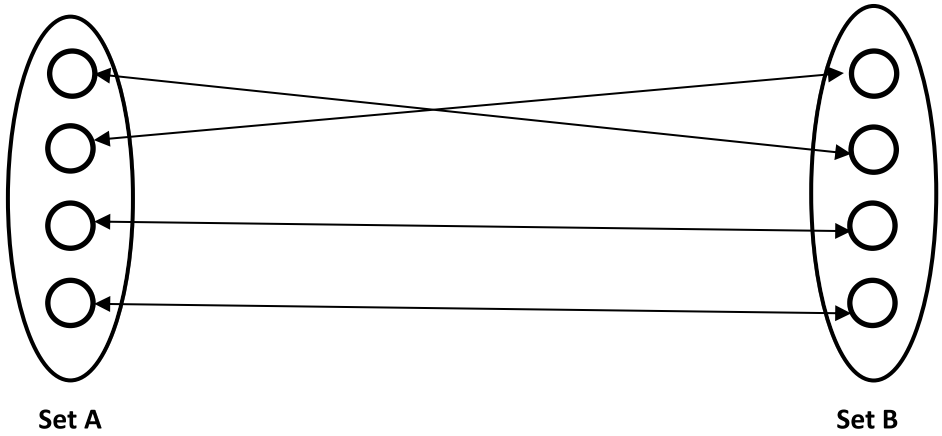
❖ Degree of relationship:-

- Binary-degree2
- Ternary-degree3
- N-ary-degree

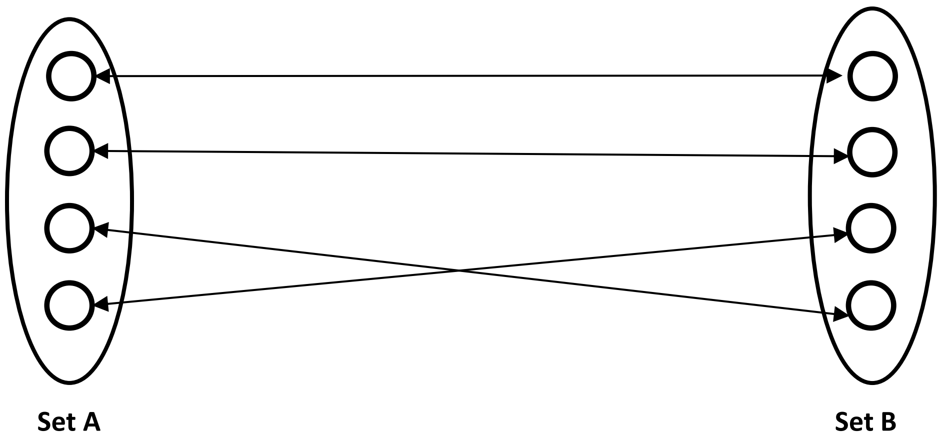
- **One-to-one:-** One entity from entity set A can be associated with at mot one entity of entity set B.



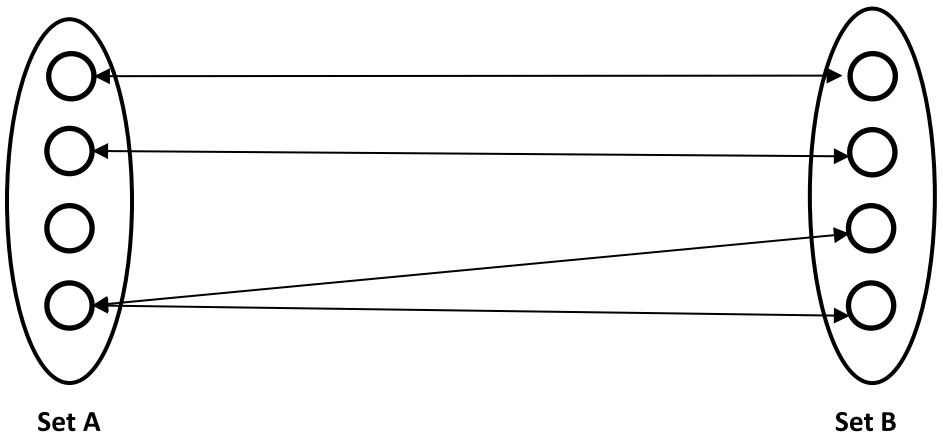
- **One-to-many:-** One entity from entity set A can be associated with more than one entities of entity set B however an entity set B, can be associated with a most in entity



- **Many-to-one:-** More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



- **Many-to-many:-** One entity from set A can be associated with more than one entity from set B and vice versa.



Entity / Object:-

- Anything that is participating in the system is known as data entity or object.
- An entity can be a person, place, thing or event for which data us collected and maintained in the system.

Attribute:-

- The characteristics of an entity is called attribute or properties.
- An entity ma have many attributes.

Entity=Customer

Customer ID
First Name
Surname
Date of birth
Address
Phone No.

There are the “attributes” for the entity ‘customer’

❖ Types of attributes:-

1. **Simple attribute:-** Simple attributes are atomic values, which can not be divided further.

   EX:- A student’s phone number is an atomic value of 10 digits.

2. **Composite attribute:-** Composite attributes are made of more than one simple attribute.

   EX:- A student’s complete name may have first name and last name.

3. **Derived attribute:-** Derived attribute are the attributes that do not exist in the physical database. But their values are derived from other attributes present in the database.

   Ex:- average_salary in a department should not be saved directly in the database instead it can be derives.

4. **Single-value attribute:-** Single- value attributes contain single value.

   Ex:- Social_security_Number

5. **Multi-value attribute:-** Multi value attributes contain more than one values.
   Ex:- Person can have more than one phone number, email, address, etc…

❖ Entity set keys:-

➢ Key is an attributes or collection of attributes that uniquely identifies an entity among entity set.
Ex:- The roll_number of a student makes him/her identifiable among students.

- **Super key:-** A set of attributes (one or more) that collectively identifies an entity in on entity.

- **Candidate key:-** A minimal super key is called a candidate key.
  ➢ An entity set have more than one candidate key.

- **Primary key:-** A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

- **Foreign key:-** Term used in relational database for an attribute that is the primary key of another table and is used to establish a relationship with that table where it appears as an attribute also.
  ➢ So foreign key value accurs in the table & again in the other table.

❖ Normalization:-

- ### Normalization of database:-

  ➢ Normalization is a systematic approach of decomposing table to eliminate data redundancy & undesirable characteristics like inertion, update & deletion

1) The process of producing a simpler & more reliable database structure is called normalization.
2) It is used to create a suitable set of relations for storing data.
3) This process work through several stages knows as normal forms.
4) Each normal form has certain requirements and condition.

- ### Normalization rule:-

  ➢ Normalization rule are divided in to following normal form.

1) **First normal form**
2) **Second normal form**
3) **Third normal form**
4) **BCNF**

1. First normal form:-

➢ A relation is in first normal form if every intersection row and column contains atomic value.

➢ It means that the relation doesn’t contain any repeating group.

- A repeating group is a set one or more data items that mat occur a variable number of times is a tuple.
- Each field in a relation should contain only one value.
- A relation can be converted in to 1NF using two method:-

- **Method 1:-** In this method the repeating groups are removed by entering proper data in blank fields that contains repeating data.

- **Method 2:-** In this method the repeating groups are removed by creating a separate relation for the repeating group.
  - The repeating data is stored in the new relation.

First normal form:-

Dept No.	Dept name	Emp No.	Emp name
10	Management	E01	Krishna
		E02	Radha
20	Finance	E10	Kanha
		E11	Gopi
30	IT	E25	Kishan

- **Applying method 1:-**

Dept No.	Dept name	Emp No.	Emp name
10	Management	E01	Krishna
10	Management	E02	Radha
20	Finance	E10	Kanha
20	Finance	E11	Gopi
30	IT	E25	Kishan

- **Applying method 2:-**

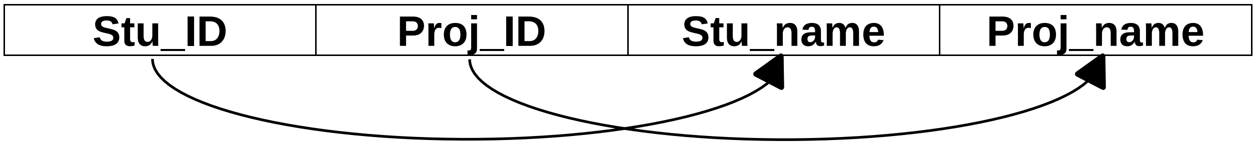
Dept No.	Dept name
10	Management
20	Finance
30	IT

Dept No.	Emp No.	Emp name
10	E01	Krishna
10	E02	Radha
20	E10	Kanha
20	E11	Gopi
30	E25	Kishan

2. Second normal form:-

- A relation is in 2NF it is in 1NF & every non-key attribute is fully functionally dependent on the primary key.
- All non-key attribute must depend on all part of the primary key.
- **Prime attribute:-** An attribute, which is a part of the prime-key is known as a prime attribute.
- **Non-prime attribute:-** An attribute, which is not a part of the prime-key is said to be a non-prime attribute.
- If we follow second normal form then every non-prime attribute should be fully functionally dependent in prime key attribute.
- That is, if X->A holds, then there should not be any proper subset Y of X for which Y->A also holds true.

- Student_project



Stu_ID	Proj_ID	Stu_name	Proj_name

[Relation not in 2Nf]

- We see here in student_project relation that the prime ky attribute are stu_ID & Proj_ID.
- According to the rule, non-key attribute i.e stu_name & Proj_name ust be dependent upon both and not on any of the prime ley attribute individually.
- But we find that stu_name can be identified by stu_ID & Proj_name can be identified by proj_ID independently.
- This is called partial dependency, Which is not allow in second normal form.

- **Student:-**

Stu_ID	Stu_name	Proj_Id

- **Project:-**

Proj_ID	Proj_name

[Relation in 2NF]

- We broke the relation in two as depicted in the above picture, so there exists no partial dependency.

❖ Condition for 2NF:-

1. The primary key consist of only one attribute.
2. No non-key attribute exist in the relation.
3. Ever non-key attribute is functionally dependent on the full set of primary key attributes.

3. Third normal form:-

- For a relation to be in third normal form, it must be in second normal form and the following must satisfy.
- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency $X \longrightarrow A$, then either.
- X is a super key or,
- A is Prime attribute.

- Student_detali:-

Stu_ID	Stu_name	City	Zip

[Relation not in 3NF]

- We find that in the above student_detail relation, stu_ID is the key & only prime key attribute.
- We find that city can be identified by stu_ID as well as zip it self.
- Neither zip is a super key nor is city a prime attribute. Additionally stu_ID->Zip->city, so there exists transitive dependency.
- To bring this relation in to third normal form, we break the relation in to two relation as follows,

student_detail:-

Stu_ID	Stu_name	Zip

Zip_codes:-

Zip	City

4. Boyce- codd normal form:-

- BCNF is an extension of third normal form on strict terms.
- BCNF states that,
- For any non-trivial functional dependency, X ⟶ A, X must be a super key.

- Stu_ID is the super key in the relation student_detail & Zip is the super-key in the relation zip codes, so,

Stu_ID ⟶ stu_name, zip

& Zip ⟶ City

- Which confirms that both the relations are in BCNF.