Concept of DBMS :

(i) Data is defined as any fact or info that can be recorded which always has any implicit meaning eg. name, address of a person represents Data.

(ii) complex programs are already designed which allows the user to create, maintain and manipulate the data

(iii) Applications :
(a) Telecommunication : Cellular system
(b) Banking : Net Banking
(c) Airlines : Reservation
(d) Universities : Registration
(e) Sales : Customers
(f) Manufacturing : Production
(g) Human Resources : Employee

(iv) common Terms :
(a) Application Programs are normally used to access the database and sending request for data to the database. This is normally called Frontend.
(b) Actual database can be referred to retrieve data through application program using technique called Query known as Backend
(c) Typically a database system follows Client Server Model where
(d) A transaction refers to a process to write data into database or read data from database such that all database operations are through Transactions.

Differences between File and Database

Data can be stored in two broadways either using traditional files or using a dedicated DBMS. The difference between them or advantages of DBMS are follows -

- Concurrency : DBMS allows multiple users to access the data such that it uses a technique called concurrency control. This approach is more difficult with files because files are normally stored in disk memory, sharing of the file is possible only by creating exact copy or transferring the file using dedicated program.
DBMS provides parallel view of different data which files cannot provide.

- Difficulty in Accessing Data : when files are used they can access only specific data. Every program will have its own data unlike DBMS which provides a central facility to access the data which can be of different types. Files are locally used to store the data but DBMS and is globally used to store the data.

- Isolation of Data and Program : In traditional files data and code are always together like Java program to store record. It means there is a tight coupling between program and data. In BBMS program and data are independent. It means program refers to Application (front End) and data is backend such that data is accessed through queries. Data and program when they are independant, it adds reliability unleike files.

- Multiple Views of Data : DBMS efficiently supports many views of data like out of 10 fields from employee database it is possible to show 3,5 or 7 fields using DBMS. This requirement is difficult using Files. Also files can only provide only limited view of data and changes in program are required for limited views.

- Data Redundancy and Inconsistency : when files are used to store data different codes are used to maintain individual database which results in redundancy or duplicated. And if any record is deleted all remaining records may not be aligned properly, it results in data inconsistency.
BBMS allows data consistency and avoids redundancy by separating data from application programs and all data is accessed through query.

Advantages of Database :
(i) controlling Redundancy
(ii) Data Independence
(iii) Control unauthorised access
(iv) Persistent storage
(v) Multiple User Interface
(vi) Backup and Recovery

Disadvantages of Database :
(i) High investment in hardware, software and training
(ii) Overhead for providing security, concurrency.
(iii) Embedded systems with limited storage

Users of Database

Typically a database system users can be classified as –

(A) Database Administrator (DBA) : DBA normally handles DB servers and configures DB for list of users, level of access, security of data, different views on data, permission of data
DBA is considered as central Authority.

(B) Database Designer : Database Architecture is normally designed by them like different layers of database, response of database once query is fired and different ways to organise the data.

(C) End User : Most of the applications of DBMS are used by End User or sometimes developed by End User. There are four types –

(a) Casual User : These end users occasionally use DBMS by executing a query like managers at different level or midlevel employees. This forms very small fraction of end user.

(b) Naive or Parametric - This forms largest community of end user who access differently database system directly (GUI) like schedule of Airlines, Banking Applications, etc These users do not directly involve in query, developing or executing the query.

(c) Sophisticated - Most of the engineers and technical person form this group because they use DBMS for developing systems which are not much complex but can be used by using front end and back end combinations.

(d) Standalone - These are end user which stores data not using full fledged DBMS System but using a file approach like Excel on a dedicated machine. People who maintain data for finance, income tax returns, etc because they use already formulated excel sheets.

(D) System Analysts and Application Programmer.
These groups are mainly used to develop large scale, complex DBMS applications like Airlines, Railways reservation schedules. System Analysts prepares system requirements and plans entire application while Application Programmers develop source code for application which can be used by Parametric users.

Architecture

Most common architecture is known as 3 Schema
Schema is defined as description of database during database design.
Schema do not change normally.

Objective of Schema:

(i) All users should be able to access same data i.e. uniformity should be maintained in critical application like Banking.

(ii) A users view is immune to changes made in others view, it means in concurrency applications where multiple users can get different views of data. If view of any one category is uddapated, other categories should not be updated, so that every user gets individual views.

(iii) Normally end users need not know about how data is stored in physical medium which track or which sector data is available or how data is accessed. This isolation is possible through 3 Schema Architecture.

(iv) DBA should able to make changes in th DB without affecting the views of different users and it is possible through 3-Schema Architecture.

3-Schema Architecture Layer

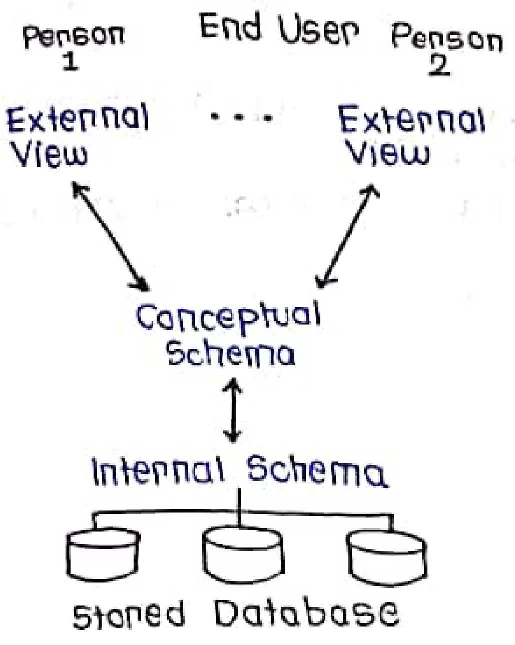
External Schema: Most visible part of DBMS in which a particular group of users like students, teachers, etc gets their own views of data like data of teacher may not be visible to student. This multiple view facility is provided by External

Conceptual or Logical Schema: Core part of DB design used to describe how data is presented in DBMS. This part starts with description of Data using a tool called ER Diagram.

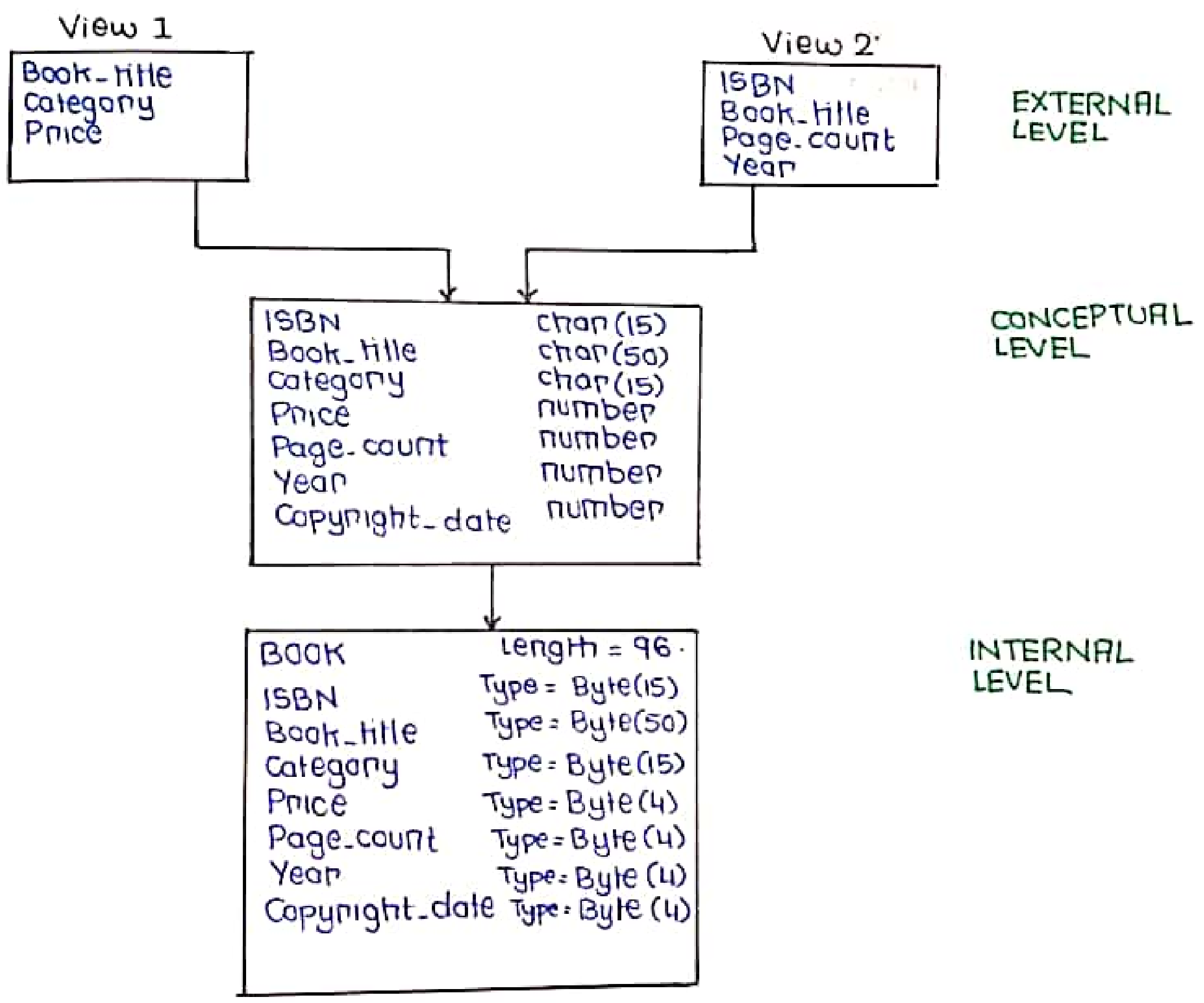
Internal Schema or Physical Schema: Lowest level of DBMS where data is physically stored in disk memory. It handles concepts like where data is stored, how data is accessed, structure for recovering of data and all these details are invisible to the user.

3-Schema architecture provides independence of user from database and physical medium.

Representation of 3-Schema Architecture



Referring to Diagram, It is observed that in a typical DBMS there can be multiple external schema but single conceptual schema or internal such that communication between two schema always occurs using Interface. Because multiple external Schema refers to single conceptual Schema, a technique called Mapping specially between External and conceptual.

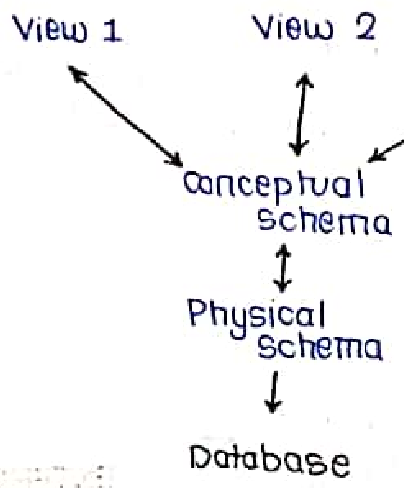


Concept of Data Independence

capacity to change a particular Schema at one level of DBMS without affecting Schema at higher level or any other level is called Data Independence.

Logical Data Independence : capacity to change conceptual Schema without changing external schema or physical device.

Physical Data Independence : capacity to change physical Schema without changing conceptual schema.

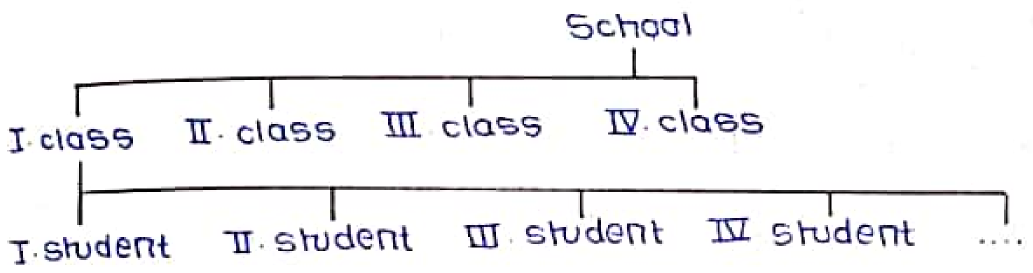


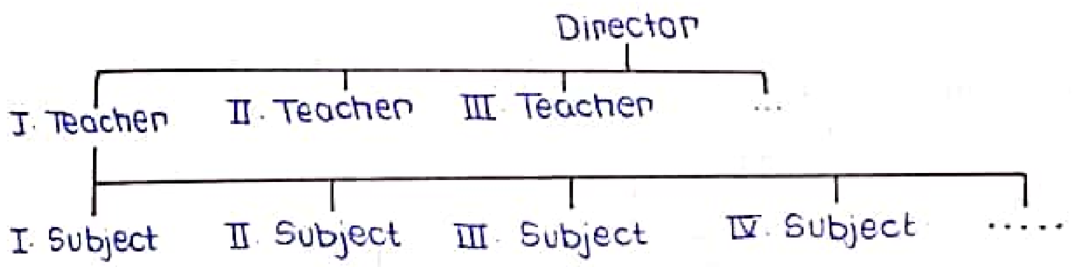
Data Modelling

Collection of concepts applied at conceptual schema to describe structure of entire Database.

(i) Hierarchial
(ii) Network
(iii) Relational

Hierarchial
All the data is described in hierarchial tree structure with one referrence called Root. A parent child relation visible between two levels of data such that at the intersection of tree a record is visible which normally forms leaves. Main advantage is it is comparatively simple to design because all standard data structures are available. Main limitation is all the data is always accessed from a fixed reference (Root) and Location of a data in a tree decides how much time is required to access the data or path of the data which may vary application to application. Another limitation is limited Relationship is possible because it is a tree structure and it is difficult to provide relationship between root levels.





Concept of Data Integrity and Constraints

Data Integrity refers to Authorization of Data such that redundancy of data or invalid operation on data can be avoided using different types of Constraints and business rules.

Domain Constraint :
Always refers to column of a table such that only the valid data as per datatype of column is entered like a column with numeric datatype will note get values like Date, etc.

Entity Integrity :

can be unique (no redundancy) or values should be NOT NULL like mobile number which must be not null or unique.

Column Constraint :
Applicable on specific column value but unlike domain, column constraints are applied according to business rule like age of specific employee will not be null (Entity Integrity) and age must be between 18-60 which is a business rule.

User Defined Integrity
As per business rule, specially coded like Salary of grade I employee must be greater than grade II.

Referential Integrity
Applied between two different tables which are related, such relationship can be defined in terms of Foreign and Primary key for consistency of data.

Table DEPT

DEPTNO	DNAME	LOC
20	Research	Dallas
30	Sales	Chicago

TABLE EMP

EMPNO	ENAME	JOB	MGR	IMREDATE	BAL	COMM	DEPTNO
7340	Vignesh	Data	731	13-05-1999	800.00	100.00	20
7541	Yash	Science	741	14-06-2005	900.00	200.00	40

Refering to example, it is observed that there are two tables such that DEPTNO is a primary key becoming Foreign key by relating two tables.
If attempt is done to insert records with DEPTNO 40 or NULL, it will not be because of Referential constraints as only 20 and 30 are allowed.
This constraint helps for consistency of data across all tables.

All these constraints can be implemented in DATABASE using Standard keywords in SQL Query.

NOT NULL
UNIQUE KEY
PRIMARY KEY
REFERENTIAL
CHECK

Three tables are created to maintain record of Bank customers such that using customer ID it is possible to find details of specific customer. using agent ID it is possible to find details of specific agent. This process it is comparatively difficult and not very practical in Hierarchial Model.

ER

Graphical Representation of data which is required by Database such that ER model represents conceptual Schema. Typically there are three important requirements -
(A) Entity :   (B)

Entity : Defined as a real world object indistinguishable from other such that such object has physical presense. Eg Person is an Entity like Student, Customer, etc such that collection of all such Entities is called Entity Set.
There are two types -
(a) Strong Entity : Entity which is independent in existence, can be uniquely identified is called Regular or Independent Entity.
(b) Weak Entity : Entity whose existence depends on other strong entities, they do not have unique existence are called dependant Entity.

Eg. Course like DBMS is strong Entity whereas modules of course like ER Model is weak Entity because they depend on Strong Entity.

Symbolically Strong Entities are represented by Rectangle and weak Entity is represented by Double Rectangle.

Attributes : Every Entity can be described using different properties called Attributes such that attributes are used to identity specific Entities.
(a) Simple (Atomic) : Attributes of Entity which cannot be further divided, which are always described uniquely
Eg. Surname.
(b) Composite : Attributes which can be further divided into Atomic Attributes are called Composite Attributes.
Eg. Address (street, town, postal code)
(c) Multivalued : Attributes which may have more than one value.
Eg. Phone number, E-mail-ID.
(d) Complex : Attributes which are nested, it means its a combination of attributes like multivalued and composite.
(e) Based or Derived : Attributes which are calculated or derived from other attributes.
Eg. Average Salary (It is calculated from salary which is simple)
Age (calculated from DOB)
(f) Single Valued : Attributes which always has a unique value. Unlike multi valued.
Eg. Social Security Number like Adhaar or PAN number.

Concept

Most of the Database related operations are performed using SQL in which standard commands are available for different on database like Create Table or Select from Table.
These queries are Mathematically modelled using Relational Alegebra. Not every query are mathematically modelled but majority of queries

1. Select Operator (σ)
   A relational operator which produce a table containing subset of rows of argument table (base table) with some condition.
   Syntax : $\sigma_{condition}$ relation

Select Operators can be used with different types of comparison and logical operators -
operators : $<, \leq, \geq, >, =, \neq$
Simple selection condition :
- <attribute> operator <constant>
- <attribute> operator <attribute>

<condition> AND <condition>
<condition> OR <condition>
NOT <condition>.

Examples :

$\sigma_{id > 3000 \text{ OR } Hobby = \text{'hiking'}}$ (Person)

$\sigma_{id > 3000 \text{ AND } id < 3999}$ (Person)

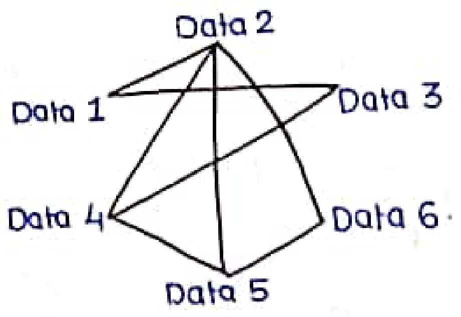
$\sigma_{\text{NOT } (Hobby = \text{'hiking'})}$ (Person)

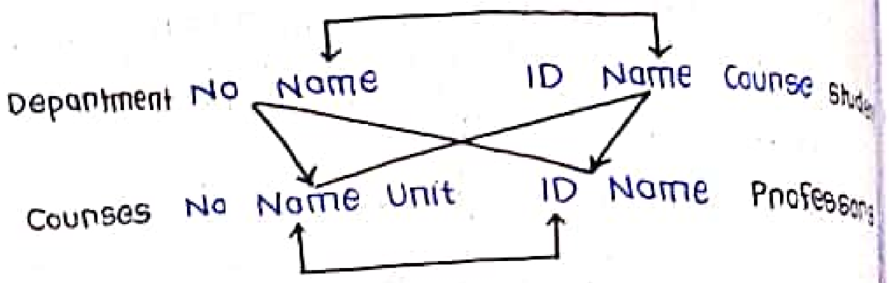
$\sigma_{Hobby \neq \text{'hiking'}}$ (Person).

2. Project
   Relational Operator which produces a table containing subset of columns of attributes of base table.
   Syntax : $\pi_{attribute\ list}$ (relation)

Network

Extension of hierarchial model is called Network Model such that unlike hierarchial model which only supports only tree model structure, network model supports multiple structures (There is no specific root node). Hierarchial model only supports 1 to N, 1 to 1 relationship but network model also supports many to many (n : m) relationship along with (1:n) and description of data is more efficient.





There are 4 different type of records visible i.e. Students, Professors, Courses and Department Records Data for every record is linked with other record in a random, like a particular student (Name) belongs to specific department, opt of specific course (Name) which is provided by specific department (Name) and taught by specific teacher (Name).

Network Model is more efficient than Hierarchial because it supports all types of relations (1:n, n:1 and m:n) but Network model is not very flexible to changes and very complex to implement as conceptual Schema.

Relati

Relational is the most common and visible model to handle huge records or data most efficiently. All the data is represented by 2D table with rows (called as Attributes) and rows (called as Tuples). All relationships between the data are supported (1:n, n:1, n:m) such that diverse applications can be implemented. Data is normally accessed using SQL (Structured Query Language). PLSQL is also available for programing the database.

Customer_ID	Customer_Account	Agent_ID
1224	4556	23
1225	4558	25

Agent_ID	Last_Name	First_Name	Phone
23		David	334-5678
25		Kyle	556-3421

Customer_ID	Last_Name	First_Name	Phone	Account_Balance
1224	Vina	Dyne	678-9987	1223.95
1225	Davis	Trichia	556-3342	234.25

3. Cartesian Product
If R and S are two relations or tables. R×S is called Cartesian Product. Used to concat tuples (rows) <x,y>
where x represents R and y represents S
Cross Operator is extensive to compute due to permutation and combination and hence it isn't very common like σ or π operator

Example:

a	b
x_1	x_2
x_3	x_4

R

c	d
y_1	y_2
y_3	y_4

S

a	b	c	d
x_1	x_2	y_1	y_2
x_1	x_2	y_3	y_4
x_3	x_4	y_1	y_2
x_3	x_4	y_3	y_4

R×S

4. Rename
In Cartesian Product, It is observed that if name of the attribute is common between two tables, the final outcome of cartesian product which is a table will have ambiguities which not be supported. Such ambiguities can be removed using Rename (ρ)
Rename allows to change name of attributes for a table and assign substitute names shown in Syntax.
Syntax : R(A,B,C) , ρS(X,Y,Z)(R)
Columns A,B,C are renamed to X,Y,Z in Relation R.

Example:
Transcript (StudID, Scode, Semester, Grade).
Teaching (ProfID, Scode, Semester)
$\pi_{\text{StudID, Scode}}$ (Transcript) [StudID, SCode1] ×
$\pi_{\text{ProfID, Scode}}$ (Teaching) [ProfID, Scode 2]

Q. Two tables needs to be merged such that table1 is employee with attributes empID, Name, Address and Name of Dept.
second table is Department with attributes DeptID, Dept Name, Dept Location and No of employee.
(a) Remove the ambiguity is merging.
(b) Identify Primary Key and Foreign Key if they are independent.
(c) write Relational Algebra Expression to display all employee name whose department is ABCD

One inbuilt feature of Project operator is all the records with duplicate values are automatically filtered and there is no redundancy

1123	John	123 Main
1123	John	123 Main
5556	Mary	7 Lake Dr
9876	Bart	5 Pine St

It is possible to make complex operation expression by using multiple operators.

$\Pi_{Id, Name}(\sigma_{Hobby = 'stamps' \text{ OR } Hobby = 'coins'}(Person))$

Q. Assume that a database contains a table Student with attributes student ID, Student Name, CGPA, Contact. Write Relational Algebra expressions to perform :

Display all the names of Student whose ID is more than 100
Display names of student whose CGPA is more than 6.
Display Student ID and Student Name whose CGPA is b/w 6 and 7
Display Student ID and CGPA for all the names whose mobile number is more than 9000000000.

A company has several departments. Each department has a supervisor and atleast one employee. Employee must be assigned to atleast one, but possibly more departments. At least one employee is assigned a project, but an employee may be on vacation and not assigned to any projects.

Business Rules :
Set of Rules used to describe properties of application specifically data part of application.

Examples:
The manager of the department must belong to that specific department. If there are multiple department, employee of other department is not allowed as a Manager.

ER

Graphical Representation of data which is required by Database. such that ER model represents conceptual Schema. Typically there are three important requirements –
(A) Entity : (B)

Entity : Defined as a real world object indistinguishable from other such that such object has physical presense. Eg. Person is an Entity like Student, Customer, etc such that collection of all such Entities is called Entity Set.
There are two types -
(a) Strong Entity : Entity which is independent in existence, can be uniquely identified is called Regular or Independent Entity.
(b) Weak Entity : Entity whose existence depends on other strong entities, they do not have unique existence are called dependant Entity.

Eg. course like DBMS is strong Entity whereas modules of course like ER Model is weak Entity because they depend on Strong Entity.
symbolically Strong Entities are represented by Rectangle and weak Entity is represented by Double Rectangle.

Attributes : Every Entity can be described using different properties called Attributes such that attributes are used to identity specific Entities.
(a) Simple (Atomic) : Attributes of Entity which cannot be further divided, which are always described uniquely.
Eg. Surname.
(b) Composite : Attributes which can be further divided into Atomic Attributes are called Composite Attributes.
Eg. Address (street, town, postal code).
(c) Multivalued : Attributes which may have more than one value.
Eg. Phone number, E-mail-ID.
(d) Complex : Attributes which are nested, it means its a combination of attributes like multivalued and composite.
(e) Based or Derived : Attributes which are calculated on devired from other attributes.
Eg. Average Salary (It is calculated from salary which is simple).
Age (calculated from DOB)
(f) Single Valued : Attributes which always has a unique value. Unlike multi valued.
Eg. Social Security Number like Adhaar or PAN number.

Concept

A typical database consists 1000s of records, to identify specific record from the bulk, a concept of key is normally used.

Super Key : An attribute or group of attributes whose values uniquely determine a specific entity or a record in Entity Set.

Employee	Super Key	Candidate Key	Primary Key	Foreign Key
1. SSN	SSN	SSN	ID	
2. employee_ID	employee_ID	employee, ID		
3. Name	Phone	Phone		
4. Phone	SSN, Name			
	Phone, Name			

Super Key represents different permutations and combination of attributes to identify a record. Fundamental concept but not used in SQL because many such Super keys are possible

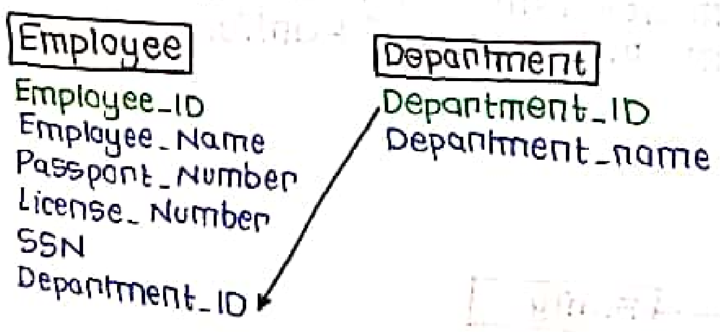
Candidate Key Minimal Superkey, it means candidate key is a Super key in which extraneous info is reduced. It is subset of Superkey. Referring to example, there are 5 Super Keys and only three candidate key such that Name, SSN or Name, Phone is removed due to extraneous info.

Primary Key is Minimal Candidate Key which is practically used in DB application, like in SQL to refer effective selective records. Refering to example there are 3 candidate keys, but only one primary key is selected because the size of this attribute (no. of character and memory requirement) is smallest. It is a subset of Candidate key.

Primary Key must follow some restrictions called constraints –

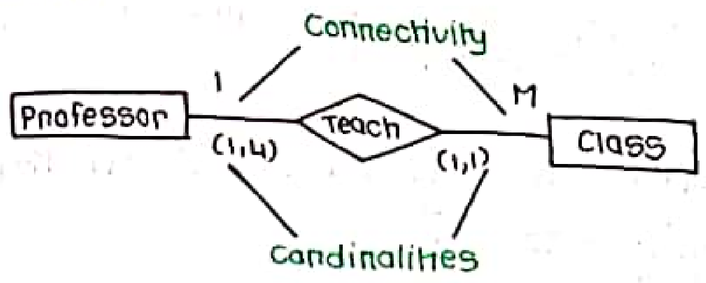
- Primary key must not be null i.e every value of Primary Key attribute must be defined.
- Primary Key must follow a rule that all values are unique and not duplicated in the values of Attribute.

Foreign Key is a type of primary key mainly used to link two tables that is while retrieving a data foriegn key can be used to retrieve a record required along with Primary Key.



Concept of Cardin

cardinality is no. of entities in one entity set which has a relation with corresponding entities with other set indirectly it indicates info or records.



Referring to example, there are two entities and it is relationship of degree 2. one Professor can handle multiple classes. And such relationship is general description. More specific relation is given using Cardinalities i.e. one professor can teach only 4 classes and one class only be taught by one professor. Forms Cardinalities of this. very similar to relationships, cardinality can be 1:M, 1:1, N:1, N:M but such relationships are more specific.

Concept of Participation

Participation refers to involvement of different entities in relationship. There are two types -

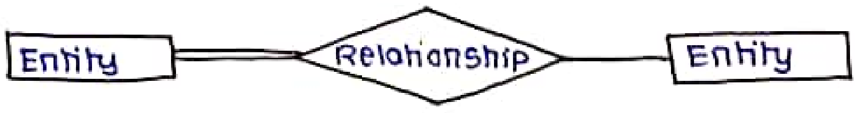
Total Participation : when every entity in entity set is involved in a relationship such participation is called Total which symbolically represented by.

Partial Participation : when not all entities in entity set are involved in a relationship such participation is called which is symbolically

Eg. when there are two entities Employee and Department with employee works in a department, every employee in entity set will belong to some department and it forms Total

Eg. Consider Employee and Department as Entity with Relationship Employee manages Department. In this, only few employee manage specific department and it forms Partial.

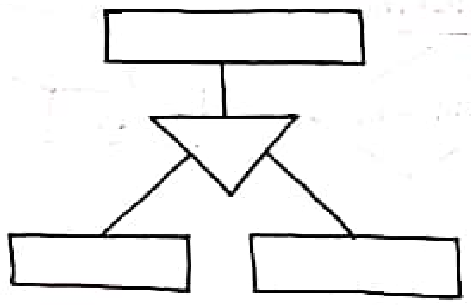
Symbolic Representation:



Concept of Specialization

Specialization is reverse process of Generalization which is also part of EER in which Entities with lots of Attributes is sub-divided into sub-entities with specific Attributes.
It is top to bottom approach in which higher level entity is divided into lower level entity and it also follows a concept of Inheritance.

Entity at the top works like base Entity. Entity at the bottom works like sub-entity. and a combination main entity handles general Attributes and sub-entity handles specific.
Specialization is required when single entity cannot handle small attributes of every record.



Information of Developer in terms of skill set, experience, projects handled is totally different from tester hence entity Employee is split into two entities (sub-entities) so with specific attributes so that retreival of Data

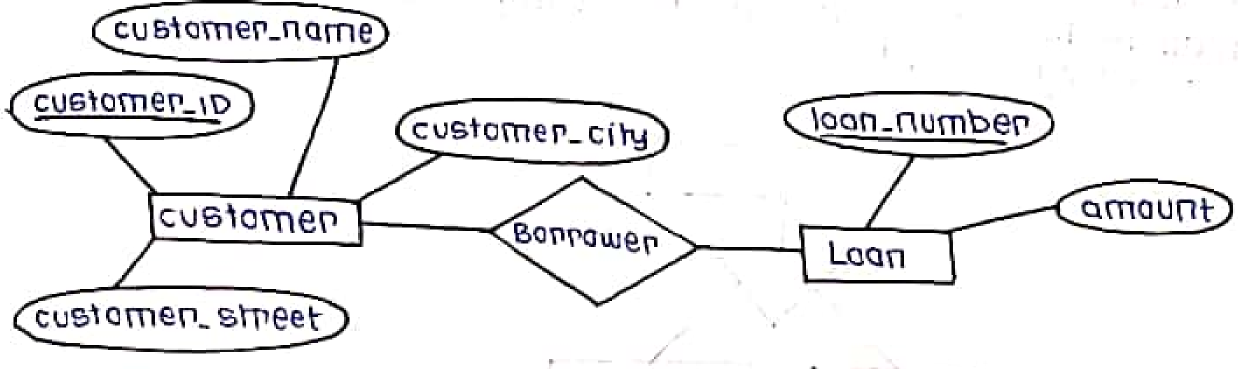
Q. Identify SuperKey, CandidatKey, PrimaryKey and ForeignKey for an Entity Student -

Name
ID
Email
Average_marks
DOB
Age
Address

There are two table which indicates employee works in a department. Employee_ID is primary key for Employee table and Department_ID is primary key of Department Table. Department_ID is used as a Attribute in Employee Table so that Department_ID becomes Foreign key for employee Table but remains Primary key for Department Table.

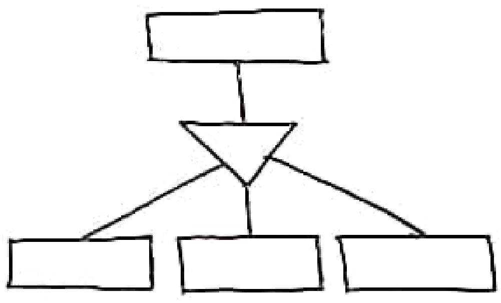
Two tables are linked using Foreign key as Department_ID.

Example:



Concept of Generalization

- It is specially defined in EER such that certain entities with common attributes can be combined into single Entity.
  It is generally Bottom up approach in which lower level entity is combined to form higher level.
  Generalization is indicates Inheritance such lower level entities called sub-entities are combined to form main entity (base class).
  Generalization is normally used when redundancy is visible in ER in the form of repeated attributes for different entity making ER more complex.

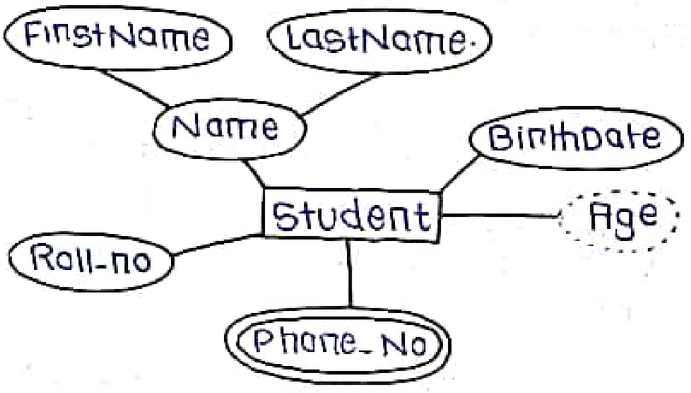


when entity SET is defined with different attributes. It is possible that Specific Entity in Entity SET do not have Attribute value like E-mail or Street name. Such attributes are referred as NULL Attributes and it can be applied at any Attribute.

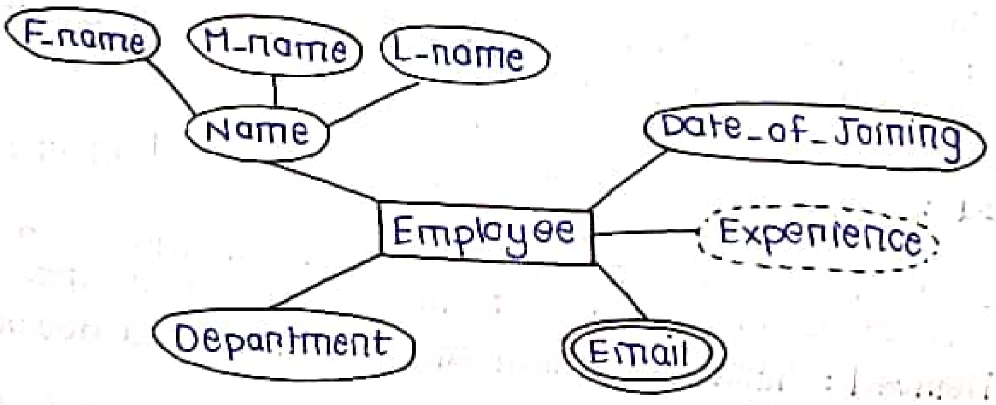
Symbolic Representation :

ER Diagram starts with Entity such that it is represented by Rectangle and every Entity has many Attributes represented by Ellipse.
Composite Attributes are represented by tree structure such that every node is connected to composite Attribute and every node represent Simple.
Multivalued Attributes are represented by Double ellipse.
Derived Attributes are represented by dotted ellipse.

Example :



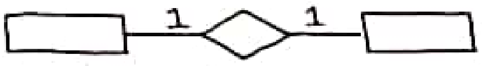
Q. Draw ER diagram to represent employs of organisation using different Attributes
Name
ID
Department
Date of Joining
Email
Experience.



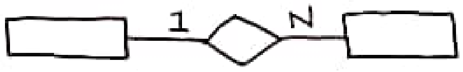
Relationship

Association between two or more Entities which is third requirement of ER. Relationship are always represented by Diamond. Nature of Relationship is written in diamond. Both the entities are connected through Diamond.

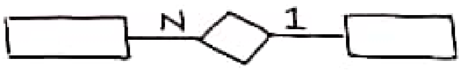
1 : 1 : One instance of entity is associated with another instance of another entity.



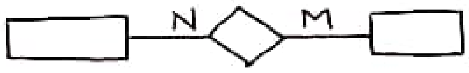
1 : N : One Instance of entity is associated with multiple instances of another entity.



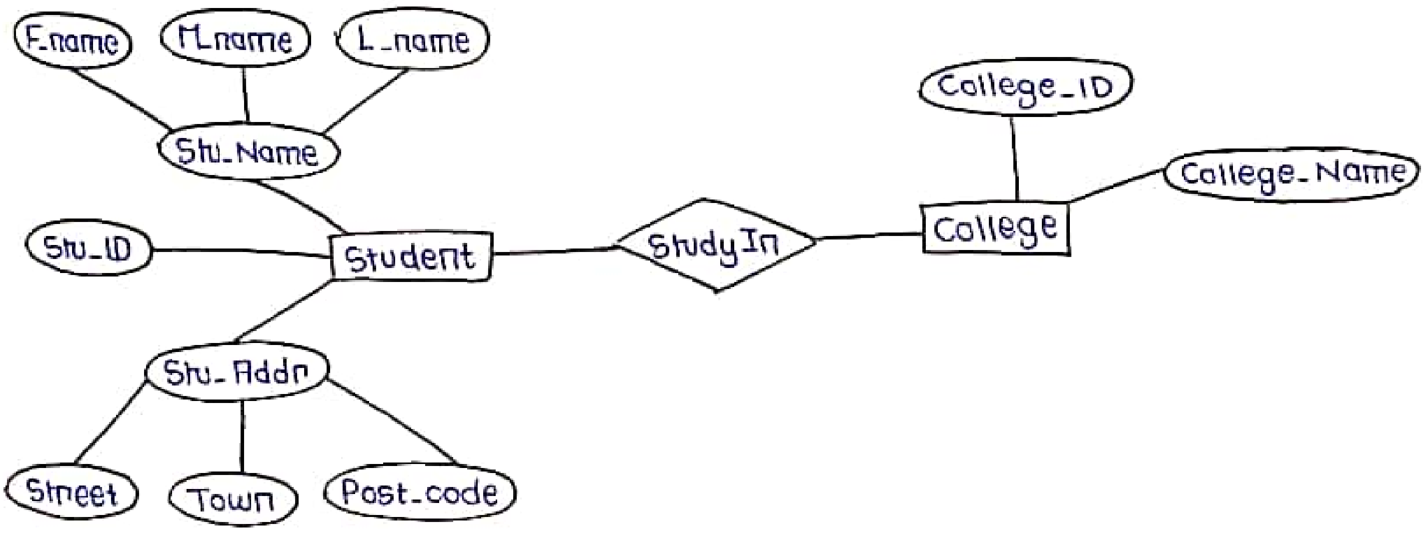
N : 1 : Multiple Instances of entity are associated with single instance of another entity.



N : M : Multiple instances of entity are associated with multiple instances of another entity.



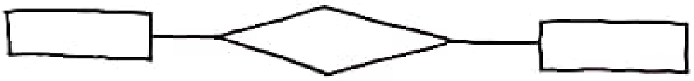
Example :



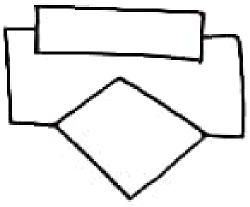
Degree of Relationship

Relationship is one of the basic requirement for drawing ER Diagram such that it can be classified into four types depending on how many entities are linked by it -

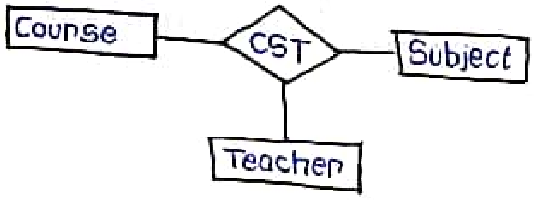
Binary Relationship : In which exactly two entities are linked by a relationship.



Unary Relationship : In which exactly one entity is linked with Relationship



Ternary Relationship : In which exactly three entities are linked by a relationship.



Complex Relationship : In which more than three entities are linked with a relationship.