Unit-1

Overview:

Data: It is nothing but Raw facts. which is Unorganised but can be made organized into useful information. Eg: 1, ravi, Qis, ... etc

Information: processing the data is called Information

Eg: The roll No 1 is Ravi.
(or)
Ravi Bearing RollNo '01'. Studying in CSE Stream in Qis college.

Database: Database is a well organised collection of data that is stored in a computer system.
(or)
It is a collection of data that describes the activity of one or more related organizations.

Eg: A university database, might contain information about the following Entities such as Students, faculty, courses and classrooms.

Relationship between Entities such as Student Enrollment in Courses, faculty teaching courses and the use of classrooms for Courses.

Database Management System: (DBMS)

DBms Stands for Database management systems, acts as an interface between the database and its End-user programs, Enabling users to extract, update and manage how the information is organized and Optimized

* DBms is a Software designed for Manipulating (Storing, retrieving, updating, deleting) the data.

* Some examples of popular database software or database management systems include Microsoft SQL Server, Oracle Database, MySQL, dBASE, Microsoft Access and Filemaker pro.

Applications of DBMS: Applications where we use database management systems are :-

1. Banking: In banking systems DBMS is used for storing customer information, tracking day-to-day credit and debit transcations and generating Bank Statements, etc.

2. Online shopping: Online websites like Amazon, flipkart Use DBMS to store product information, addressing and preferences credit details and provide us for the list of products based on our Query.

3. Airline: In Airline Reservation system, DBMS is used to keep records of flights Arrival, depature and delay status.

4. Manufacturing: In manufacturing, DBMS is used to keep track of records of all the details about the products like Quantity, bills, purchase, Supply Management, etc.

5. Human Resource Management: In big organizations there are many workers working Human resource management keep records of Each Employee's Salary, tax and work through DBMS

6 Telecom: In telecom there is a need of DBMS to keep track of information regarding Callsmade network usuage, Customer details, etc.

7. Education systems: Db's are used frequently in schools & colleges to Store and retrive

the data regarding Student details, Staff details, Course details, exam details, fee details, etc.

8. Railways: Db's are used to keep records of ticket booking, train departure and arrival status and keep track of trains that are getting late, etc.

9. Library Management System: using DBMS we can maintain all the information related issue dates, Names of books, Author, Aavaliability of the book, etc.

10. Social media: Daily thousands, millions of users Signed up for the social media sites like Fb, twitter, etc All the information of these users are stored and maintained by using databases only.

11. Military: Military keeps the records of millions of soldiers information and millions of files using DBMS keeping the data safe and secure.

Database Users:

* for a small personal database one person defines, constructs and manipulates the database and there is no sharing.

* In Large organizations many people involved in design, use and maintenance of a large database with hundreds of users.

* We have two kinds of database users
  1. Actors on the Scene
  2. Workers behind the Scene.

1. Actors on the scene: These people involves the day-to-day use of a large database.

we have four categories.

(i) Database administrators (DBA): The responsibility of DBA is to administrate the resource of databases

* He is responsible for authorizing access the db.
* He will coordinate and monitor the database and acquire s/w and h/w resources needed.
* He is responsible for maintaining database Securit

(ii) Database designers: These are responsible for identifying the data is to be stored in databas and choosing appropriate structores to represent and store the data.

* These are responsible to communicate database Users to known requirements.

(iii) End users: These are the people who need acces database for Querying, updating, and generating reports.

* The database exists primarily for their use only There are four categories of End users.

(a) Casual End user: These users occasionally access the database, but they may need different information Each time.

Eg: Through Browsers.

(b) Naive (or) parametric End users: These users constar Querying and updating the database.

Eg: 1. Bank account holders continously checking the Balance.

2. Reservation agents continously checking the tickets avaliability.

(c) Sophisticated End user: These users maintain personal db by using ready made programming packages that provide easy-to-use and menu based or graphical based Interface

Eg: User of tax package stores a variety of personal financial data for tax purpose.

(d) Sophisticated End user: These include scientists business analysts who use dbms to implement their own applications to meet their complex requirements.

(ii) Workers Behind the scene: These are the people who are not intrested in the database content itself.

There are Three Categories.

(a) Database System designers and Implementers: These people will design and implement the DBMS modules & interfaces as a s/w package.

* These will implement the modules includes Implementation of catalog, Query processing, Interface processing, buffering data, Controlling Concurrency, recovery & security.

(b) Tool developers: These people will implement the tools that facilitate database modelling & design, data base system design and improved performance.

* Tools are optional packages that are often purchased Separately.

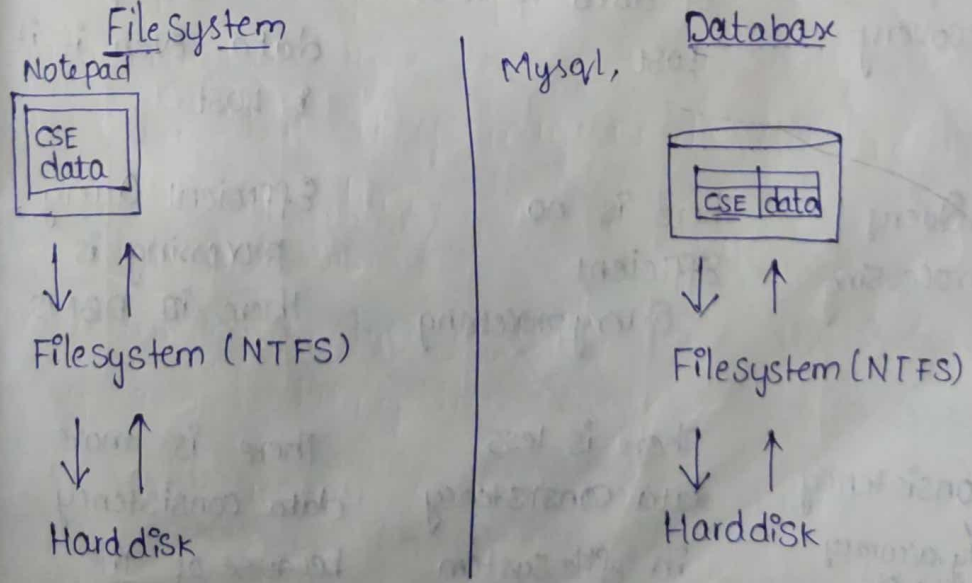
(c) Operators and maintenance personnel:
These are responsible for actual running and maintenance of the h/w & s/w environment for the database system.

→ File system Vs DBMS

Filesystem: (what is file system) → It is used to store the data
How data is stored in hard disk. The way in which we read the data from hard disk or the way in which we write the data to the hard disk (from - to) that is handled by filesystem.

* The file system gets installs in machine when you install OS.
When you install windows OS - NTFS [New Technology File system].
when you install ~~windows~~ Linux OS - EXT
(Extendable file system

* File system acts as an Interface between you & hard disk
* In Both file system & DBms the main functionality is to store the data. then what is difference?



~~Both~~, The major difference is When you store data via filesystem data reaches hard disc as Raw data (i.e, no structure)

* In RDBMS, to store my data in db. Im Giving Instructions through table (Query). i.e, Structure.

* Random access of data is not possible in file system

→ Differences Between File system vs DBMS

Basics	Filesystem	DBMS
(i) Structure	The file system is S/w that Manages and Organizes files in Storage Medium within a Computer	DBMS is S/w for Managing the Database.
(ii) Data Redundancy.	Redundant data Can be present in a file system.	In DBMS, there is no redundant data.
(iii) Backup & Recovery	It doesn't provide Backup and Recovery of data if it is LOST.	It provides backup and Recovery of data. Even if it is LOST
(iv) Query processing	There is no Efficient Query processing	Efficient Query processing is there in DBMS
(v) Consistency ↓ (validity, accuracy & usability of related data. It ensures that each user observes a same view of data, including Changes made by the user's own	There is less data consistency in file system	There is more data consistency because of the process of Normalization

(vi) Complexity	It is less complex.	It has more complexity
(vii) Security constraints	It provides Less security	It provides high security
(viii) Cost	It is less expensive	It has a comparatively higher cost than a file system.
(ix) Data Independence (If data is changed at a level, it does not impact the data view at the higher levels)	There is no data Independence	→ In DBMS data Independence exists
(x) User Access	only one user Can access data at a time	Multiple users Can access data at a time.
(xi) Data Isloation	In file systems the data will be stored in different file formats (.txt, .html, .war) etc, so we have to write application programs to retrive data from Each file	In databases the data is store at one place, so we can access data Easily.
(xii) Atomicity problems	During transcations, there maybe possibility of failing Computer System. In that case the transcation Should not happen. They Should be rolled back. Such type of Mechanism is not there in file System	This will be Efficitively done in DBMs

→ ~~Advantages (same as filesystem vs DBMS~~

→ Demerits of DBMS

* high cost
* Huge size
* complexity
* Requirement of Technical staff

→ Storage Data: [How data is stored in Database?]

A database is collection of tables. A table is just like what it sounds. It is a collection of Rows & columns.

* The raws are the data itself, the columns are the attributes we care about for a table.
* Table is nothing more than a spreadsheet.
* Let's say we wanted to store information about the car's we are selling.

we might have a table that looks like this:

ID	Year	make	model	color	mileage	Iss
1	2011	Toyota	Camry	Red	72000	Yes
2	2015	Nissan	Altima	Black	14000	NO

In a database table, a column is a specific kind of data. Before giving column data we have to specific ~~lahaa~~ data type for that column.

There are three buckets of column types in a database: Text, Numeric & other.

(i) Text column Types: Text columns hold varying lengths of well text. Text can be numbers, characters, punctuation or anything.

The most common text column types are:

* VARCHAR — This is varying length character column. This means that when you define the column you designate the maximum length of the text that can be stored.

Eg: If you had a column that was VARCHAR(60) that would mean that the column can hold a maximum of 60 characters. In the above table make and model would probably be VARCHAR Columns. Eg: Create table Cardata(make VARCHAR(20), model VARCHAR(20));

* CHAR — This is a character column. AS with VARCHAR, you define the length of the column. The difference between VARCHAR and CHAR is that CHAR is fixed storage length, meaning that the database will pad your text with spaces if you dont fill the Entire field.

Eg: SQL> Create table cardata(color char(10));

* Text — A TEXT column type is used when you don't known how long the text being Stored could / should be. It has no pre-defined length.

(ii) Numeric Column Types: Numeric column types means numeric characters. Such as Numbers

* INTEGER – whole numbers, just like you would image. The ID column in the above Example is an Example of an Integer column

Eg: Create table Carsdata (ID INT(30));

* FLOAT – Floating point decimal can float in the values being stored.

for Eg: float could store 1.2345, 1.2, and 12354.12343535345 in the same column.

(iii) Other column types:

* BOOLEAN – A BOOLEAN column stores the value TRUE OR FALSE.

The "Issold" column above is an Example of a BOOLEAN column.

* BLOB or BYTE ARRAY: BLOB columns are used are used to store things like images.

* It's more common these days to not store the actual image but rather a link to the image but in some cases you might want to store the image itself.

→ Queries:

Structured Query Language

A Query is a request for data or information from a database table

* This data may be generated as results returned by Structured Query Language (SQL)

* Structured Query Language is a standard Database Language which is used to Create, maintain and retrive the relational database.

* SQL is case insensitive. But it is a recommended practice to use key words (like SELECT, UPDATE, CREATE, etc) in capital letters and use user defined things (like tablename, column name, etc) in small letter.

* we can write comments in SQL using -- (double hyphen) at the beginning of any line.

* SQL is the programming Language for relational databases like MySQL, Oracle, Sybase, SQL Server, Postgre, etc. Other non-relational databases (also called NO SQL) databases like MongoDB, DynamoDB, etc do not use SQL.

What is Relational Database?

Relational database means the data is stored as well as retrieved in the forms of relations (tables).

SQL Create table Student;

Student table created.

Insert into student(ROLLNO INT(10), NAME VARCHAR(10), Address VARCHAR(20), phone INT(10));

Student

ROLL NO	Name	Address	phone
1.	RAM	Delhi	96xxxxxxxx
2.	Sujain	Mumbai	86xxxxxxxx
3.	abc	Hyderabad	83xxxxxxxx

Some important terminologies that are used in terms of relation.

Attribute: Attributes are the properties that define a relation

Eg: ROLLNO, NAME, etc.

Tuple: Each row in the relation is known as tuple. The above relation contains 3 tuples.

Data base Languages:

(i) Data Definition Language (DDL)
(ii) Data Manipulation Language (DML)

(i) Data Definition languages (DDL): The Language which is used to define the data in the database is called Data definition Language.

* DDL Statements are CREATE, ALTER, DROP TRUNCATE.

a) Create: It is used to create the database

Syntax: SQL> Create table tablename (attributename datatype (size));

Eg: SQL> create table student (Sid number(6), Sname varchar(15), marks number(6));

SQL> Table created.

b) ALTER: It is used for altering the database

Syntax: SQL> Alter table tablename add columnname datatype(size);

Eg: Adding a new column to the student table

SQL> Alter table Student add address Varchar(15);

c) TRUNCATE: Removes all rows from a table, but the table structure and its columns remains same.

Syntax: SQL> truncate table tablename;

Eg: SQL> truncate table Student;

d) DROP: It is used to drop or delete the table from database.

Syntax: SQL> drop table tablename;

Eg: SQL> drop table Student;

Syntax for dropping a column from the Student table:

SQL Syntax: SQL> Alter table tablename drop column column name;

Eg: SQL> Alter table student drop column address.

(ii) Data Manipulation Languages (DML):

The language which is used to manipulate the data in the database is called data-Manipulation Language.

* DML Statements are INSERT, UPDATE, DELETE, MODIFY.

1. Insert: It is used for inserting a new row into the database table. There are 2 ways of Inserting data into the database

(i) Single Row Insertion
(ii) Multiple Row Insertion

Syntax: Insert into tablename values();

(i) Single Row Insertion:

Syntax: Insert into Student values (561, 'abc', 70);

SQL > 1 row Inserted;

(ii) Multiple rows insertion:

SQL> Insert into Student values (&sid, '&sname', &marks);

SQL> Enter value for sid: 501
SQL> Enter value for Sname: def
SQL> Enter value for marks: 75

SQL> 1 row Inserted

SQL> /

2. UPDATE: This Statement is used to update the data in the database table.

Syntax: SQL> update tablename set column = value where condition;

Eg: Student

Sid	Sname	marks
561	abc	70
562	pqr	65

SQL> UPDATE Student Set Sname = 'xyz' where Sid = 562.

3. DELETE: It is used to delete the one or more rows from a relation.

Syntax: SQL> delete from tablename where Condition;

Eg: Student

Sid	Sname	marks
561	abc	70
562	pqr	65

SQL> delete from Student where Sid = 562.

4. MODIFY: It is used to modify the data in the database table. used to change the datatype of the column

Syntax: SQL> Alter table tablename modify Columnname datatype (size);

Eg: SQL> Alter table Student modify sid number(10);

(iii) Data Control Language (DCL): DCL includes commands such as 'GRANT' and 'REVOKE' which mainly deals with the rights, permissions and other controls of the database system.

a) * GRANT: This command gives users access previleges → (a special right) to the database.

b) * REVOKE:- this command withdraws the user's access previleges given by using the GRANT command.

→ To display the Entire relation the Syntax is SQL> Select * from tablename;

Eg: SQL> Select * from Student;

Sid	Sname	marks
1	a	60
2	b	60

SQL> Select Sid, Sname from Student;

Sid	Sname
1	a
2	b

→ To display all the relations in your database,

Syntax: Select * from tab;

→ To known the description of the table the Command is Syntax: SQL> desc tablename;

Eg: SQL> desc Student;

Attributes	datatypes	Size	Key
Sid	number	10	?
Sname	number	15	-
marks	number	6	-

Rename: It is used for renaming a relation

Syntax: SQL> rename Oldname to newname;

Eg: SQL> rename Student to Std;

* To rename a column the Syntax is

SQL> Alter table tablename rename column Old columnname to newcolumnname;

Eg: SQL> Alter table Std rename column Sid to Sno;

→ Transcation Management:

A transcation is a set of logically related operations.

For Example, you are transferring money from your Bank account to your friend's account, the set of operations would be like this:

Simple transcation Example:

1. Read your account Balance
2. Deduct the amount from your Balance
3. Write the remaining balance to your Account
4. Read your friends Account Balance.
5. Add the amount to his account Balance.
6. Write the new updated balance to his Account

This whole set of operations can be called a transcation. above 6 steps

In DBMS, we write the transaction like this:

Lets say your account is 'A' and your friend's account is 'B', you are transferring 10,000 from A to B, the steps of the transcation are:

1. R(A); // Read operation
2. A = A-10000;
3. W(A); // Write operation
4. R(B);
5. B = B+10,000;
6. W(B);

Transcation failure in Between the Operations:

Now, ~~that we understand~~

We should understand what are the transcation problems associated with it.

→ The main problem that can happen during a transcation is that the transcation can fail before finishing the all the operations in the set. This can happen due to power failure, system crash, etc. This is a serious problem that can leave database in an inconsistent state.

* Assume that transcation fail after the third operation i.e, the amount would be deducted from your account but your friend will not receive it.

To solve this problem, we have the following two operations

(i) Commit: If all the operations in a transcation are completed successfully then commit those changes to the database permanently.

(ii) Roll Back: If any of the operations fails then roll Back all the changes done by previous operation.

Even though these operations can help us avoiding several issues that may arise during transcation but they are not sufficient when two transcations are running concurrently. To handle those problems we need to understand database ACID properties.

ACID properties: To Ensure Integrity of data during a transcation. the database system maintains the following properties.

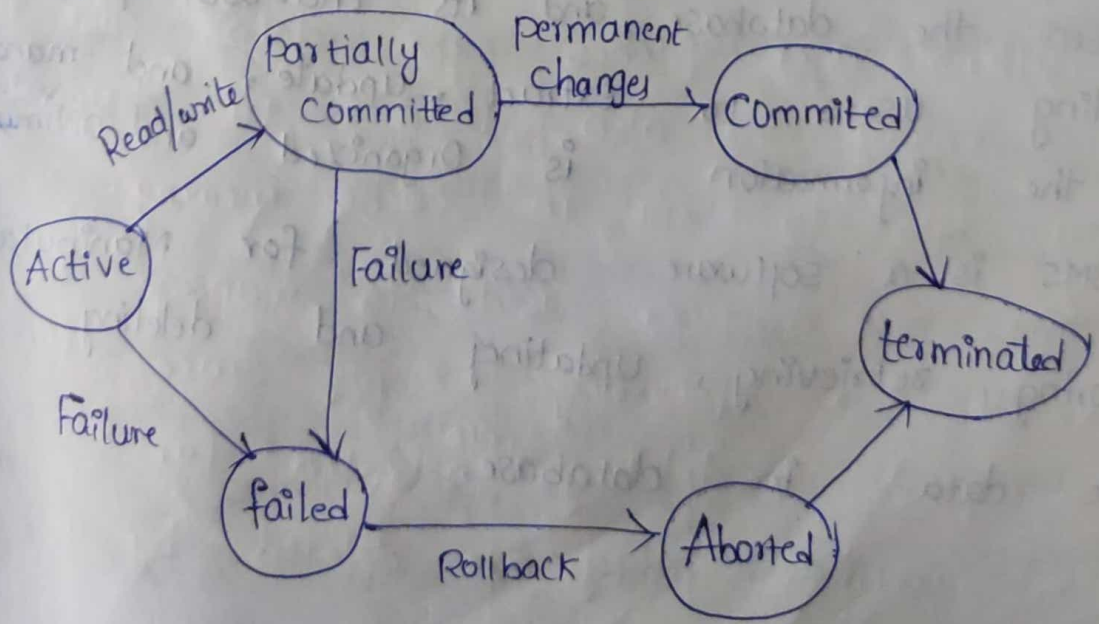
(i) Atomicity: This property Ensures that Either all the operations of a transcation reflect in database or none.

(ii) Consistency: No transcation should have any adverse Effect on the data residing in the database. If the database was in a consistent state before the Execution of a transaction, it must remain consistent after the Execution too.

(iii) Isolation: For Every pair of transcations, One transcatio Should Start Execution Only when the other finished Execution.

(iv) Durability: Once a transcation Succesfully completed the changes it has made into the database should be permanent Even after the system failure. The Recovery -management component of db's Ensures the durability of transcation.

DBMS transcation States Diagram:



(i) Active state: If a transcation is in Execution then It is said to be in active state. It doesn't matter which step is in execution, until unless the transcation is executing, it remains in active state.

(ii) Failed state: when failure occurs either a s/w failure or h/w failure during transcation then the transcation will goes to failed state from the Active state

(iii) partially committed state: A transcations contains number of read and write operations. Once the whole transcation is successfully executed, the transcation goes into partially committed state.

(iv) Committed state: In this state, the partially committed transcation will stored All the changes permanently in database.

(v) Aborted state: If a transcation fails during execution then it goes into a failed state.

→DBMS Structure:

Database Management system acts as an Interface Between the database and its End-user programs, Enabling users to extract, update, and manage how the information is organized and optimized

* DBMS is a software designed for Manipulation (Storing, retrieving, updating and deleting the data in database.

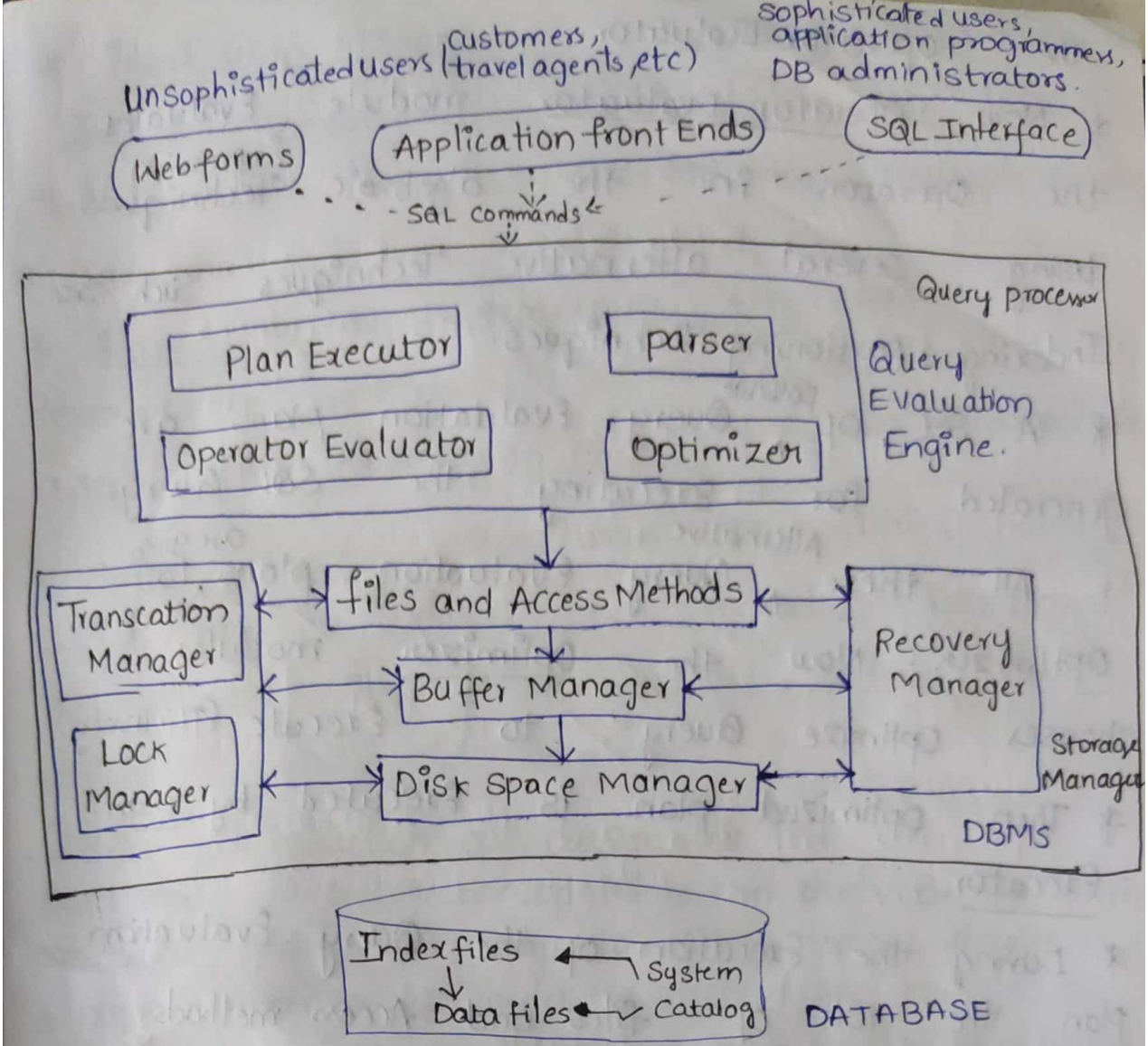


Fig: Structure of DBMS.

* The DBMS accepts SQL Commands generated from a variety of user interfaces, produces Query Evaluation Plans, Executes these plans against the db. and returns the answers.

* Query Evaluation plans Executes by Query Evaluation Engine i.e, parser, Operator Evaluator, Optimize Plan Executor.

* Initially SQL Commands are received by the 'parser' then the parser breaks the Query into tokens and checks the Query is Syntatically and Semantically correct or not. If the Query is correct then it is converted into Algebraic Expression. then this Algebraic Expression is passed as input to

the Operator Evaluator.

* Now Operator Evaluator module Evaluates the Operators in the algebraic Expression using several alternative Techniques such as Indexing, partioning Techniques.

* A Set, of possible Query Evaluation plans are generated for Executing the SQL Query.

* All these Alternative Query Evaluation plans are given to Optimizer. Now the Optimizer modules chooses optimize Query to Execute Efficitvely.

* Then optimized plan is Executed by plan-Executor.

* During the Execution of the Query Evaluation Plan the role of files and Access methods, Buffer Manager, Disk Space Manager. are important.

* Files and Access methods provides the abstraction of filestructure Storing in database and it creates Indexes on the files in order to Access the desired file Very Quickly.

* Buffer Manager fetches the data disc into Main memory. and also used to design what data to kept in cache memory.

* Disk Space manager used to manage the Space on disk providing Empty Space for New request & deleting space allocated for Existing files when Ever they are deleted by user.

* Transcation Manager & Lock manager: used to Impose concurrency control. Because the data in the database can be accessed by multiple user at same time.

* Recovery manager maintains Log used to Restore the system in consistent data state Even after crash occurence in the system or database

* System Catalog also known as Data Dictionary. It is used to store the metadata information. ie, Contains Information tables, Indexes, Views which is stored in database.

→ Brief Introduction of Data models.

Data Model is a collection of high-level data description constructs that hide many low-level storage details.

* Data model means to model the data ie, to give a Shape to the data or to give a figure to the stored data.

There are four types of data models They are

(i) Entity-Relational model
(ii) Relational model (v) Object-oriented Data model.
(iii) Network model
(iv) Hierarichal model

(i) Entity-Relational model: [E-R model]

It is based on the Collection of basic objects called Entities and relationships among these objects. An Entity is a thing or object in the real world

Eg: person, customer

Entities are described in a database by a Set of Attributes. For Customer Entity the attributes are Customer Id, Customer name, Customer ph.no, Customer Address, etc.

Components of E-R diagram:

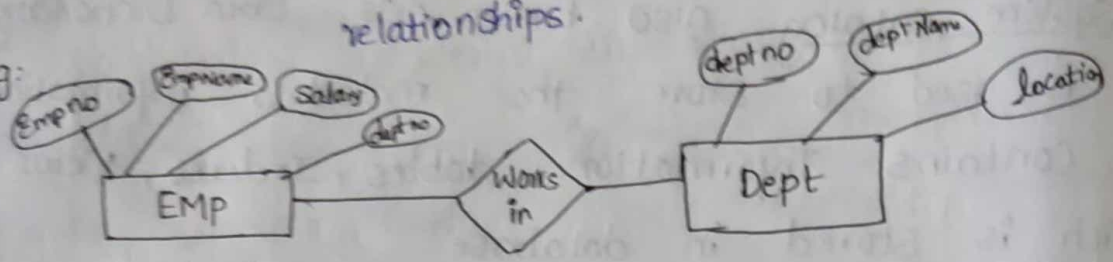
(i) Rectangle — which represents Entity sets

(ii) Diamond ◇ — which represents relationships among Entity sets.

(iii) Eclipse ⬭ — which represents attributes

(iv) Lines —— Which is used to line attributes to Entity sets and Entity sets to relationships.

Eg:



(ii) Relational model:

It represents both data and the relationships among data in the form of tables.

*Each table has multiple columns and each column has unique name.

Relational model for Emp table

Empno	Ename	Sal	deptno
1	abc	6000	500
2	xyz	6000	400

Relational model for Dept table:

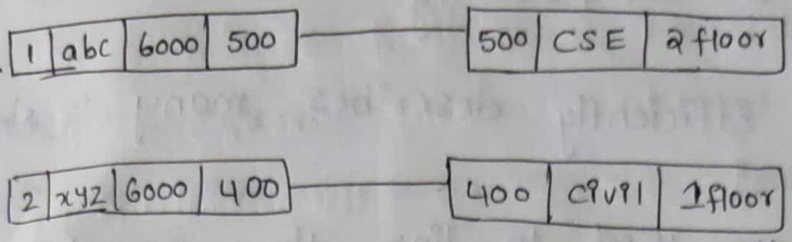
deptno	dname	loc
500	CSE	2floor
400	civil	1 floor

Relational model for works in table:

Empno	deptno
1	500
2	400

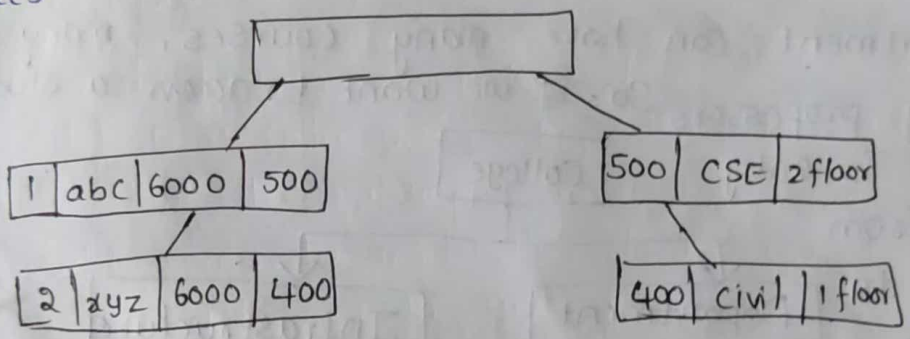
(iii) Network model: Data in the network model are represented by collection of records and relationships among data are connected by links. The records in the database are represented by graphs.

Eg:



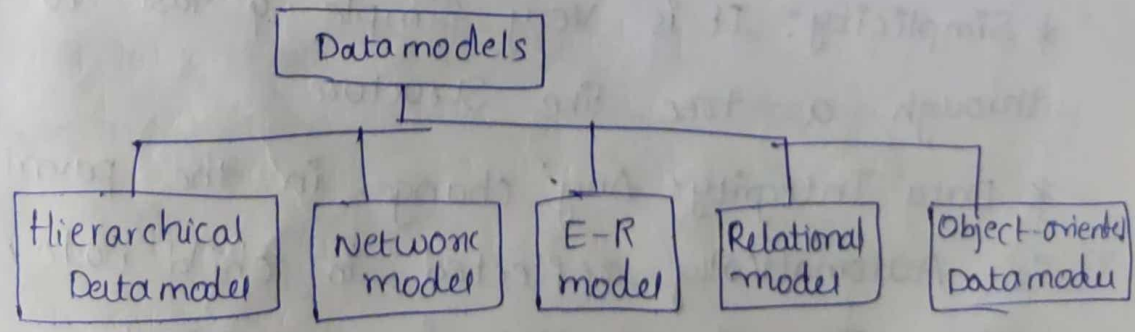
(iv) Hierarchal model: It is same as the N/w model i.e, data in the hierarchal model are represented as a collection of records & relationships among data are connected by links. The records in the database are represented by Trees.

Eg:



→ ~~Three tier Schema Architecture~~ for data independence

→ Among the above da

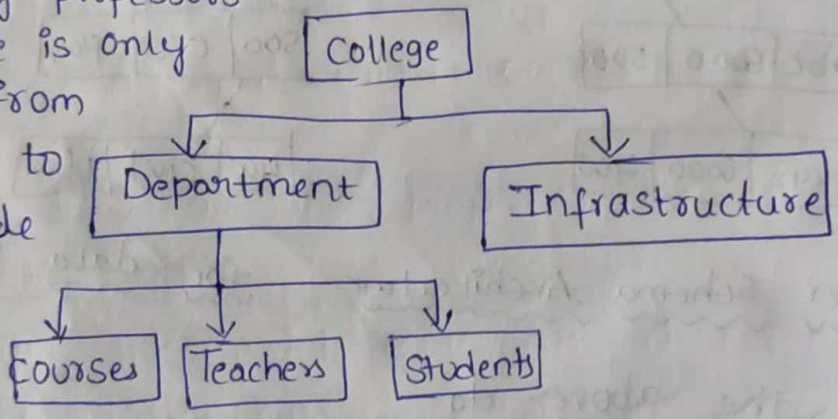


Among the above data models, relational model is the most widely used database model.

(i) Hierarchical DATA Models:

This database model organizes data into a tree like structure, starts from the Root data, and then expands like a tree, adding child nodes to the parent nodes.

* This model Efficiently describes many real-world relationships.
* pointers are used to link the parent node with the child node and also to Navigate b/w the Stored data.
* This model represents One-to-many relationships between the two different types of data, for Eg: one department can have many courses, many students and many professors. So, If we want to access a student record, there is only one path from College node to Student node



Advantages:

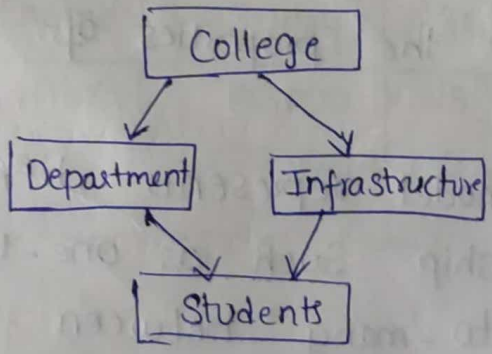
* Simplicity: It is very simple & fast to traverse through a tree like Structure.
* Data Integrity: Any change in the parent node is Automatically reflected in child node, so the Integrity of data is maintained

Disadvantages:

* Searching for data requires the DBms to run through the Entire model from Top to bottom until the required

Information is found. making Queries Very slow.
* If a parent node is deleted then the childnode is Automatically deleted.

(ii) Network Model: It organizes the data using Graph like Structures. It allows a record to have more than one parent.
* It was the most popular model. before the relational model
* This model is Same as the hierarichal model but the only difference is that a record can have more than one parent.
* This model has the ability to manage one to one relationships as well as one-to-many and many-to-many relationships. AS there are more relationship So there can be more than one path to the Same record. This makes data access easier & faster.



Advantages:
* Simple to & Easy to design
* It can handle One to many and many to many relationships
* data access is Easier & faster as there can be more than one path to reach a particular node.
* AS there is a parent-child relationship so data integrity is present. Any change in parent record is reflected in the child record.

Disadvantage:

* All the records are maintained using pointers and hence whole database structure becomes complex
* Insertion, deletion & updating operations of any record require the large no. of pointers adjustments.
* The structural change to the database very difficult

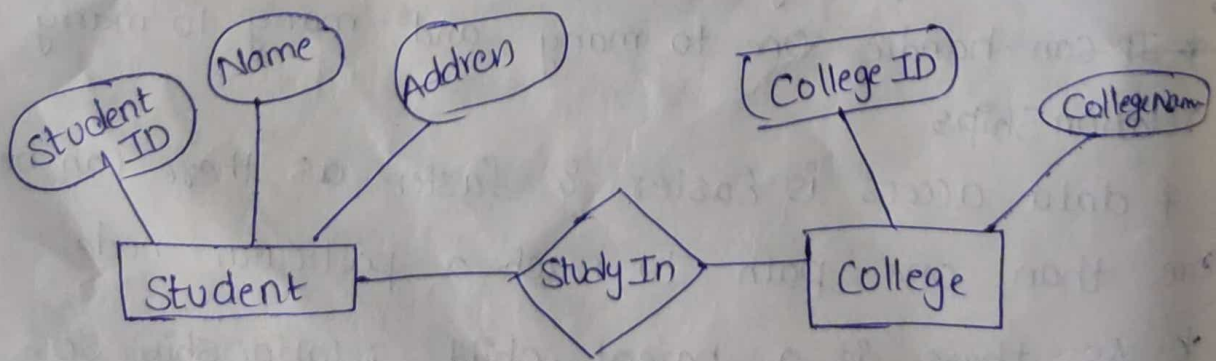
(iii) Entity - Relationship Model (E-R model):

* E-R models (semantic data model) is useful in representing a conceptual (logical) design for the database.
* It is used to pictorially denotes Entities, attributes & relation ships among them.

Entity - These are real world objects represented in Rectangle.

Attributes - These are the properties of Entities repres in eclipse.

Relationship - This model represents different types of logical relationship such as one-to-one, one-t -many and many-to-many between the Entitie in database using diamond symbols.



Advantages

* Simple & Effective Communication tool.

So this model is widely used by db designers.
* It can be easily converted to relational model and also any other data model.

Disadvantages:
* There is no industry standard for developing an ER model so one developer might use notations which are not understood by other developers.

(iv) Relational model: In this model, data is organised into 2-Dimensional tables and the relationship is maintained by storing a common field. Here tables are also known as relations.
* Each table consists of a collection rows and columns where each row represents records and column represent attribute of an Entity.

eid	ename	Salary	age
540	abc	10,000	25
512	def	20,000	35
514	ghi	30,000	40

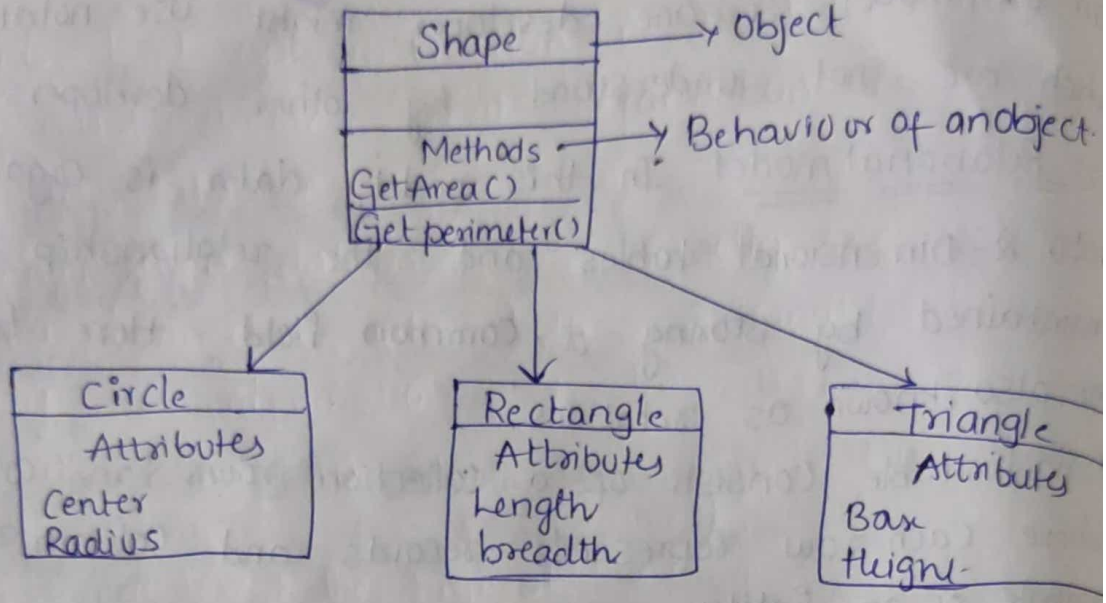
Advantages:
* Simple as compared to N/w model & Hierarchical model
* Scalable as any no. of rows & Columns can be added when ever required
* It supports Both data independence and Structure independence of DBMs.
* Easier to maintain Security

Disadvantages:
* Hardware overheads
* Bad design

But all these disadvantages are minor as compared to the advantages of the relational model.

(v) Object oriented model:

It represents data and the relationships of real world entities in a single structure which is known as object. In this model two are more related objects are connected through links.



* In this model, the key concepts of Object oriented programming Languages such as Inheritance, polymor Overloading, encapsulation and Information hiding can be Easily represented.

Advantages:

* Object oriented DB can handle different types of data such as example, pictures, voice, video, text, numbe and so on...
* Code reusability, Improved reliability & flexibility
* Low maintenance cost

Disadvantages:

* There is no universally defined data model for OODBMS
* doesn't provide adequate Security Mechanisms

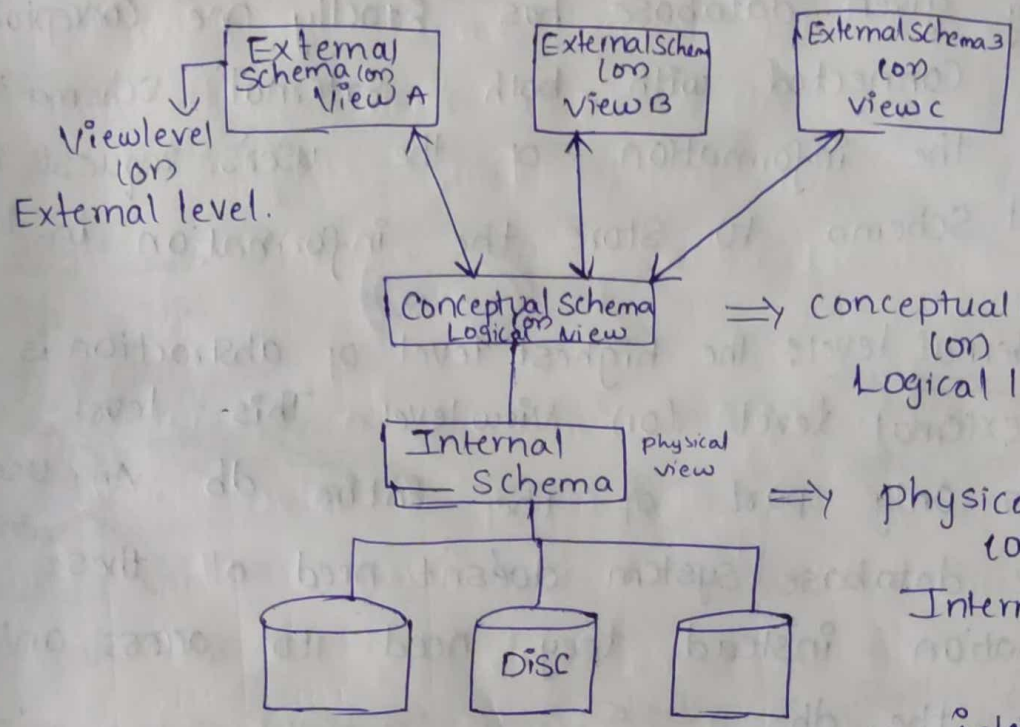
* It makes the system more complex.

→ Three tier schema Architecture for data independence

Data Abstraction

schema: The overall database design is called schema.

Instance: The collection of information stored in the database at a particular moment is called as Instance of the database.



A database system is a collection of interrelated files and a set of programs that allow users to access and modify these files.

* A major purpose of db system is to provide users with an abstract view of data. ie, the system hides certain details of how the data are stored & maintained

In DBMS data can be abstracted in 3 levels they are * physical level (or) Internal level

* Conceptual level (or) Logical level
* External level (or) view level

(i) Physical level: It is the lowest level of abstraction. It describes how the data is actually stored.

(ii) Conceptual level: The next higher level of abstraction is the conceptual level. In this the entire information of the db is displayed.

* This level of abstraction is used by DBA who must decide what information is to be kept in database.

Any given database has exactly one conceptual schema connected with both external schema to display the information of the user's request & Internal schema to store the information in the db.

(iii) External level: The highest level of abstraction is the external level (or) View level. This level describes only part of the entire db Any users of the database system doesn't need all these information instead they need to access only part of the db.

Data Independence: Capacity to change the schema at one level of a db system without having to change the schema at the next higher level.

(i) Logical data Independence: Ability to modify the conceptual schema without changing the external schemas or application programs.

Eg: If we want to add an Attribute (or) New data item than that modification in conceptual schema doesn't Effect in External Schema.

(ii) Physical data Independence: Ability to change the Internal schema without changing the conceptual-Schema.

i.e
* changing file Location, Access path but this doesn't effect the Conceptual (or) Logical schema.

* changes may be needed to improve performance

→ E-R model: [Codd proposed the relational data model in 1970]

Entity: An Entity is a Real-world thing which can be distinctly identified like a person, place or a concept

An Entity Can be of Two types:

(i) Tangible Entity: Tangible Entities are those Entities which exist in the real world physically.

Eg: person, car, etc.

(ii) Intangible Entity: Intangible Entities are those Entities which exist only logically and have no physical existence.

Eg: Bank Account, etc.

Entity Set: An Entity Set is a collection of similar Entities

Eg: Courses, academic staff, Students, flight crews, etc
Job positions,
Set of all customers in Bank, etc.

Attributes: An Entity is represented by set of Attributes. Attributes are the descriptive properties posessed by each member of an Entity set.

for Eg: Employee Entity has attributes such as EmpID, Empname and Salary.

Attribute types:

(i) Simple Attribute: The attributes Cannot be divided into Subparts.

Eg: Empno

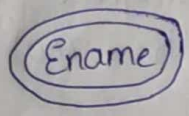
(ii) Composite Attributes: The attributes that can be divided into subparts

Eg: empname can be divided into firstname, Last name.

(iii) Single Valued Attributes: The attributes that have a single value for a particular entity is known as single valued attributes

Eg: Empno Can be only a single value for a particular Employee and it remains same.

(iv) Mutti-Valued Attributes: The attributes that have many values for a particular Entity is known as multi-valued attributes.

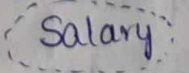


Eg: Ename

(v) Null attribute: A null value is a special Indicator that represents the absence of a value. The value Can be absent because it is unknown, not yet Supplied or non-existent. The DBms treats the Null value as an Actual value, not as a Zero value, a blank or an Empty String.

Eg: The height attribute of a person is unknown then it can be listed as null.

(vi) Derived attributes: The value for this type of attribute can be derived from the values of other related attributes



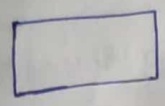
Eg: from Salary attribute we can derive to count the number of Salaries.

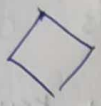
(vii) Key attributes: A key is a minimal Set

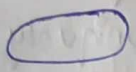
Of attributes whose values uniquely identify an Entity in the Set.

Eg: Eno = No two Employees can have the Same Eno.

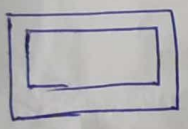
E-R diagram Notations

 Rectangle → represents an Entity

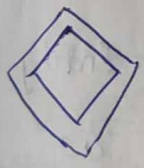
 Diamond → represents relationship sets

 eclipse → represents Attributes

 Double eclipse → represents multi-valued attributes

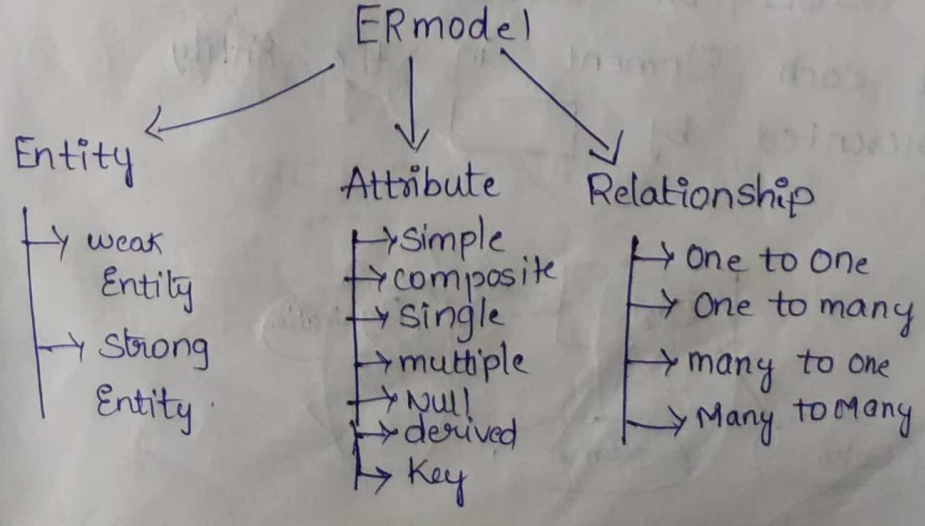
 Double Rectangle → Weak Entity set

 Line → represents Association

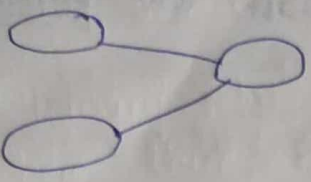
 Double diamond → Identifying relationship set

 double line → represents total participation

 Triangle → 'is a' relationship



key Attributes —



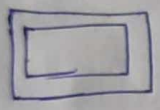
– composite Attribute

– derived attribute

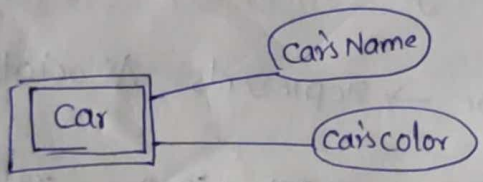
– Multivalued attribute

Weak Entity Set: Weak Entity set in DBMS is an Entity set that does not have a primary key. ie, there exists no attribute that can uniquely represent each Entity in the Entity Set

* Weak Entity is represented as Double Rectangle

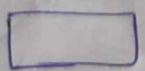


* Eg:

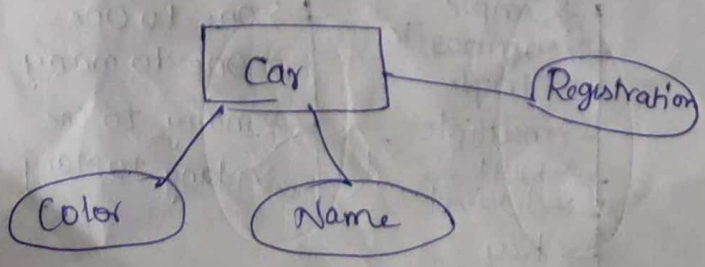


The attributes Car's Name, color is a weak Entity Set Since two cars can have the Same color as well as Same Name.

Strong Entity Set: Strong Entity set in DBMS is the Entity Set in which there exists a primary key. i.e, there Exists an attribute that can uniquely represent each Element in the Entity

* represented by 

- Eg:



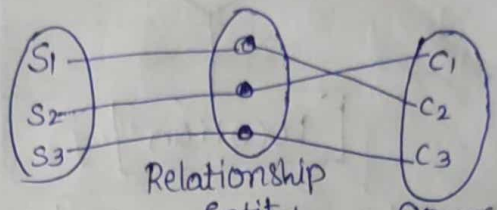
→ Relationship & Relationship sets

A Relationship is defined as an Association among several Entities.

Eg:

Relationship sets: A relationship set is a set of relationshi of same type.

Eg:



Student Entityset — Relationship Entity Set — Course Entity set

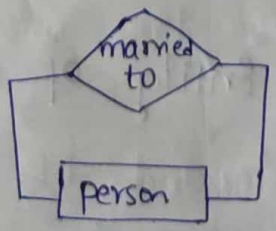
Degree of a Relationship set: The No. of Entity sets that participate in a relationship sets is termed as the degree of that relationship set

Degree of relationship Set	=	Number of Entitysets participating in a relationship set

Types of Relationship sets: on the basis of degree of a Relationship set, a Relationship set can be classified into the following types.

(i) Unary Relationship set: Where only one Entity set participa in a relationship set

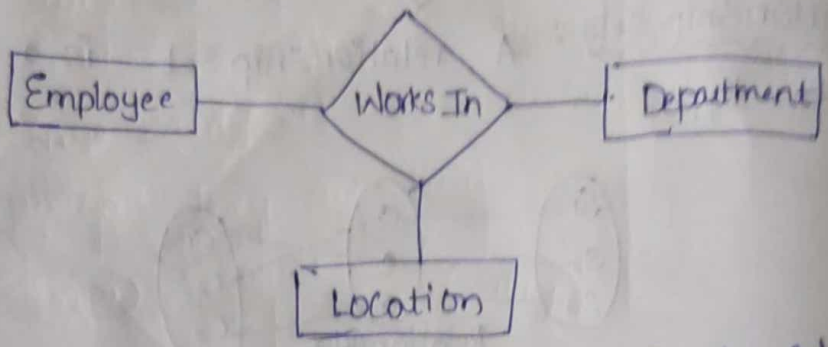
Eg: One person is married to only one person



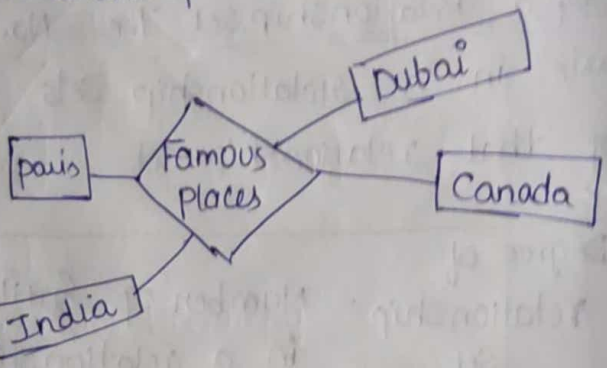
(ii) Binary Relationship Set: Where two Entity sets participales in a relationship sets.

(iii) Ternary Relationship set: Where three Entity sets participate in a relationship set.

Eg:



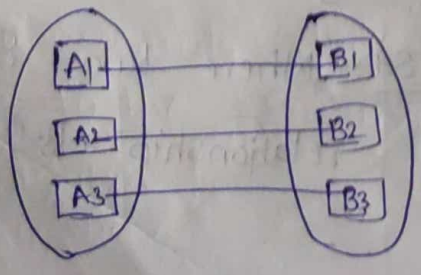
(iv) N-ary Relationship set: Where 'n' Entity sets participate in a relationship set.



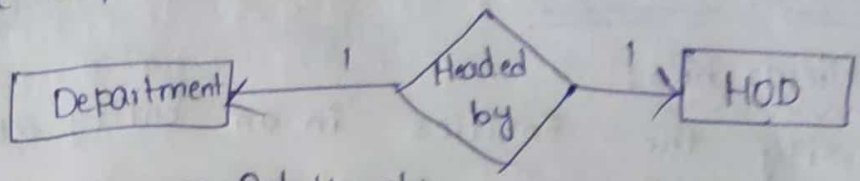
→ Mapping Cardinalites: It is expressed as the no. of Entities to which another Entity can be associated via a Relationship set.

* for Binary Relationship Set there are entity set A and B then the mapping cardinality can be one of the following

* One-to-one :- One Entity of A is associated with one Entity of B.



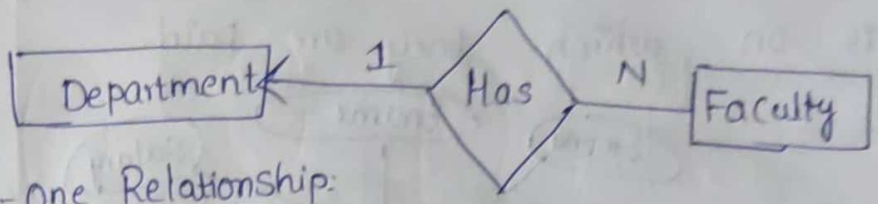
Eg: One department has one HOD



(ii) One-to-many Relationship

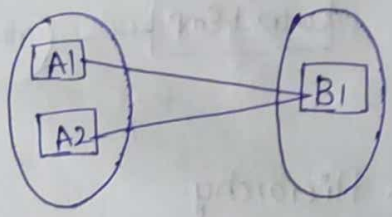
An Entity Set A is associated with any number of Entities in B with a possibility of zero. and Entity in B is associated with at most One Entity in A.

Eg:

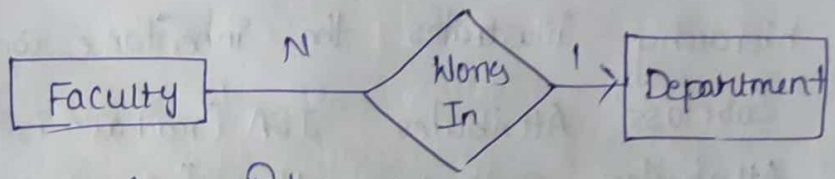


(iii) Many-to-One Relationship:

An Entity Set A is associated with atmost one Entity in 'B'. and an Entity Set in B Can be associated with any number of Entities in A with a possibility of zero.

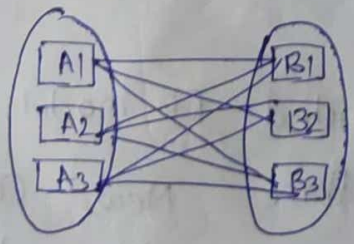


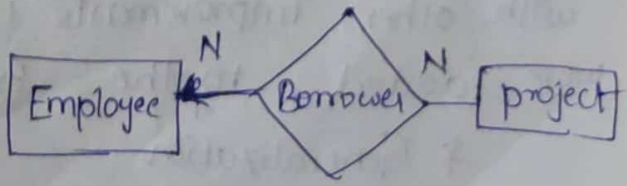
Eg:



(iv) Many-to-Many Relationship: Many Entities of A are associated with many Entities of B.

* An Entity in A is associated with many Entities of 'B'. and an Entity in B is associated with many Entities of A.





→ Features of E-R Model:

Class Hierarchies:

To classify the Entities in an Entity set into Subclass Entity. is known as class Hierarchies

Eg: we might want to classify Employees Entity set into Subclass Entities such as hourly-Emp Entityset and Contract-Emp Entity set to distinguish the basis on which they are paid.

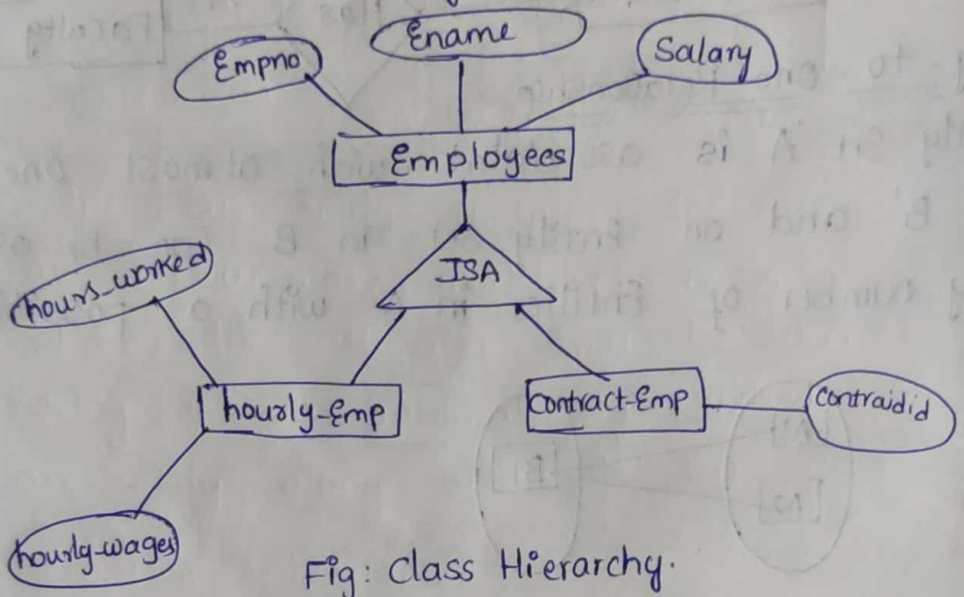


Fig: Class Hierarchy.

The class hierarchy illustrates the inheritance concept where the Subclass Attributes ISA (read as: is a) Super class Attributes Indicating the "is a" relationship

* Therefore the attributes defined for a hourly-Emp Entity set are the attributes of hourly-Emps plus(+) attributes of Employees (Because Subclass can have Super class properties. Like wise Same for Contract-Emp Entity too.

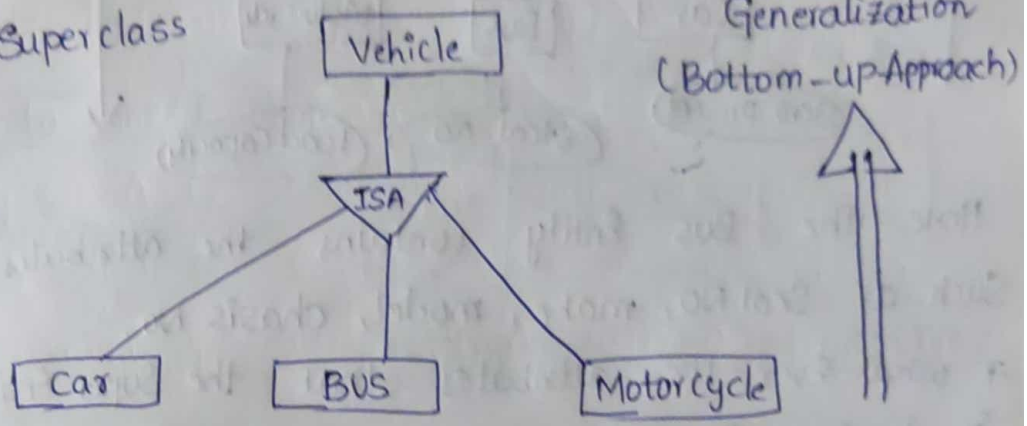
* As a part of the Extended ER model, along with other improvements, three New Concepts were added to the Existing ER model

  * Generalization
  * Specialization
  * Aggregation

(i) Generalization: Process of Extracting common properties from a Set of Entities and create a generalized Entity from it.

* It is a "Bottom-up-Approach". In which two or more Entities can be combined to form a higher level Entity if they have some attributes in common.
* ie, Sub classes are combined to form Super class.

Eg:

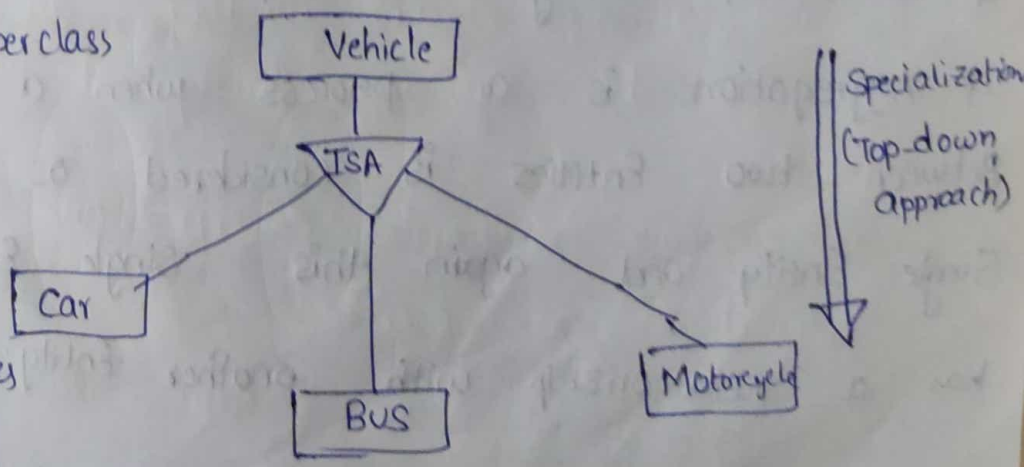


In the above Example 3 Sub Entities Car, Bus, Motorcycle. Now these 3 Entities can be generalized into One higher-level Entity (or Super class) Named as Vehicle.

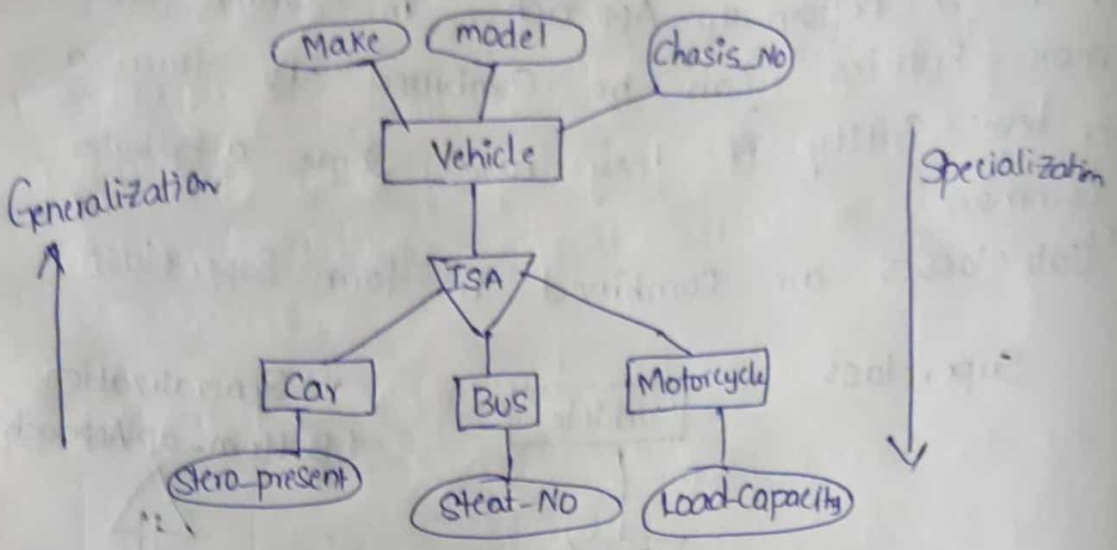
(ii) Specialization:

* It is opposite to Generalization
* An Entity is broken down into Sub Entities based on their characteristics
* It is a Top-down approach where higher level Entity is Specialized into two or more Lower Level Entities

Eg:



* Attribute Inheritance: It allows Lower Level Entities to inherit the attributes of higher level Entities



Here the Bus Entity contains the attributes Such as Seat NO, make, model, chasis_No.

* what ever the attributes that the Super Entity is having it will automatically applied to the Subclass Entity along with its own ~~Entity~~ attribute.

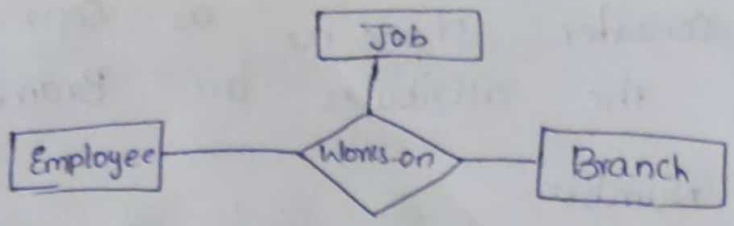
* participation Inheritance:

If an higher level Entity participates in an relation then that relation will be applicable to Lower level Entity along with the Individual relationships ~~of~~ If ~~they~~ it have.

Aggregation: It is used when we need to express a relationship among relationships

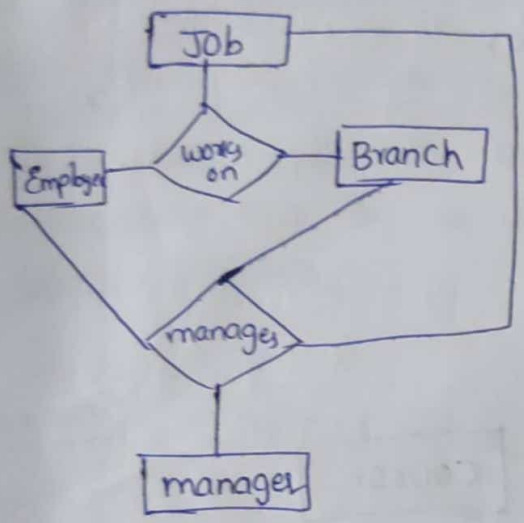
* Aggregation is a process when a relationship Between two Entities is considered as a Single Entity and again this Single Entity has a relationship with another Entity.

Eg:



An Employee Work on particular job at a particular Branch.

* Suppose we want to assign a manager for jobs performed by an Employee at a Branch. need a Separate Manager Entity Set.



This problem can be solved by aggregation

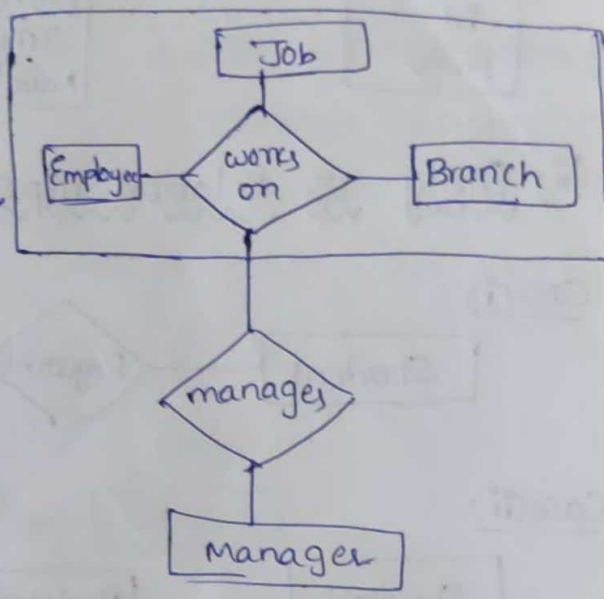


fig: It Showing overlapping & redundant Relationship.

⇒ Conceptual design with E-R model:-

(i) Entity set vs attribute:

While identifying the attributes of an Entity set, it is sometimes not clear, whether a property Should be modelled as an attribute or as an Entity set

* Consider a Student Entity with attributes Such as Name, id, phone no.

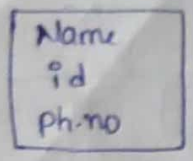
Here the attribute phoneno can be Considered as Entity.

* If you consider phone no. as Separate Entity then the attributes are 'Brand' ImEI No, Number.

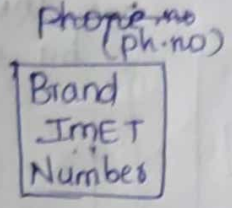
* If you use 'ph.no' as attribute then It allows one value. But when ph.no is Considered as Entity we will add multiple Phone numbers.

Case(i)

Student



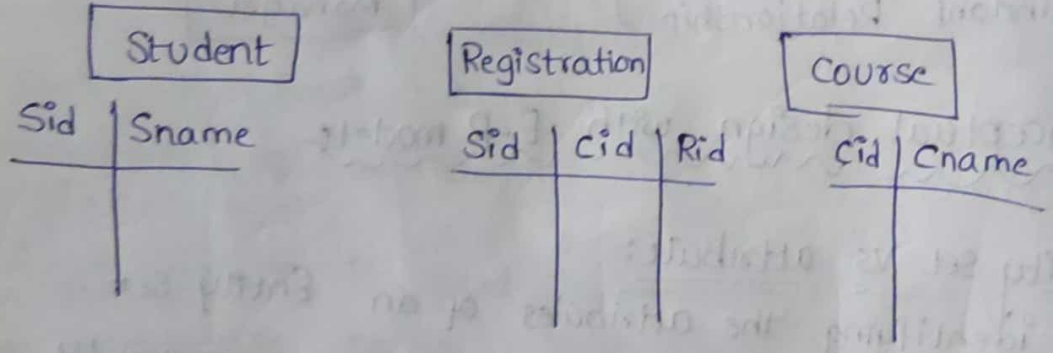
Case(ii)



(ii) Entity Vs Relationship set

Case(i)

Case(ii)



It is not always clear whether an object is best Expressed by an Entity Set or a relationship.

* In the above Example Case(i) One Student Can registered in one Course.

* But this model won't work in Situations where one Student wants to register

in more than one course.

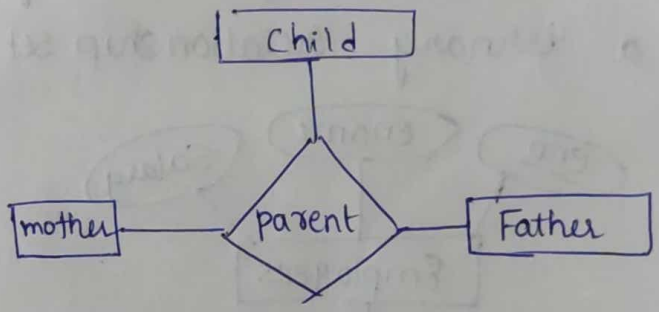
* By using case (ii) one student can registered in more than one course.

*One possible guideline is determining whether to use an Entity set or a relationship set to designate a relationship set, an action that occurs between Entities. This approach can be useful in deciding whether certain attributes may be more appropriately Expressed as relationships.

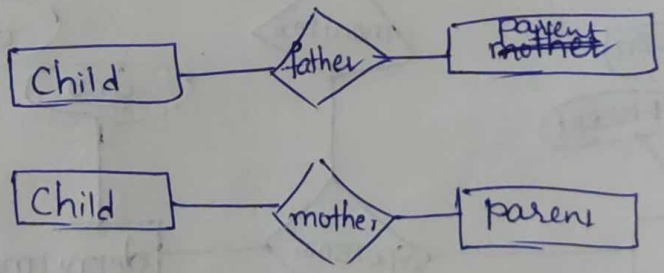
(iii) Binary Vs Ternary Relationships:

It is always possible to replace a non-binary (n-array relationship set by a number of distinct binary Relationship sets

Case(i)



Case(ii)



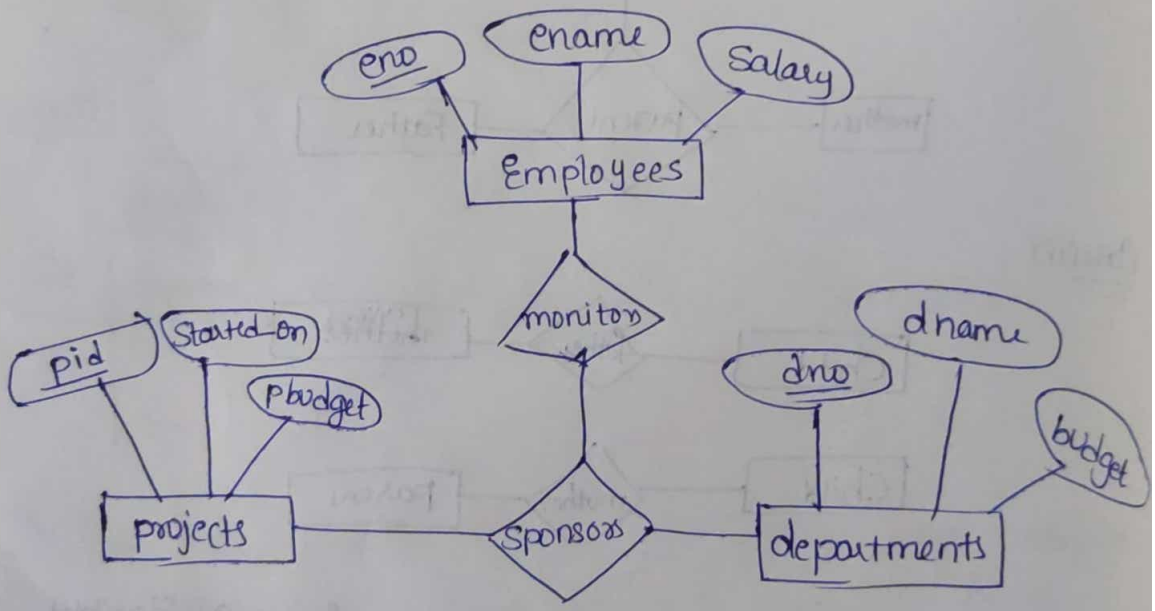
We can Generalize this process in a Straight forward manner to n-array relationship sets.

Thus, conceptually, we can restrict the E-R model to include only Binary Relationship Sets

(iv) Aggregation Versus Ternary Relationships

The choice b/w using aggregation or a ternary relationship is mainly determined by the existence of a relationship that relates a relationship set to an Entity set.

Eg: Consider the constraint that each Sponsorship (of a project by a department) be monitored by atmost one Employee. we can express this constraint interms of the Sponspors relationship set. On the other hand, we can easily express the constraint by drawing an arrow from the aggregated relationship sponspors to the relationship monitors, Thus the presence of such a constraint serves as another reason for using aggregation rather than a ternary relationship set.



Using Ternary Relationship