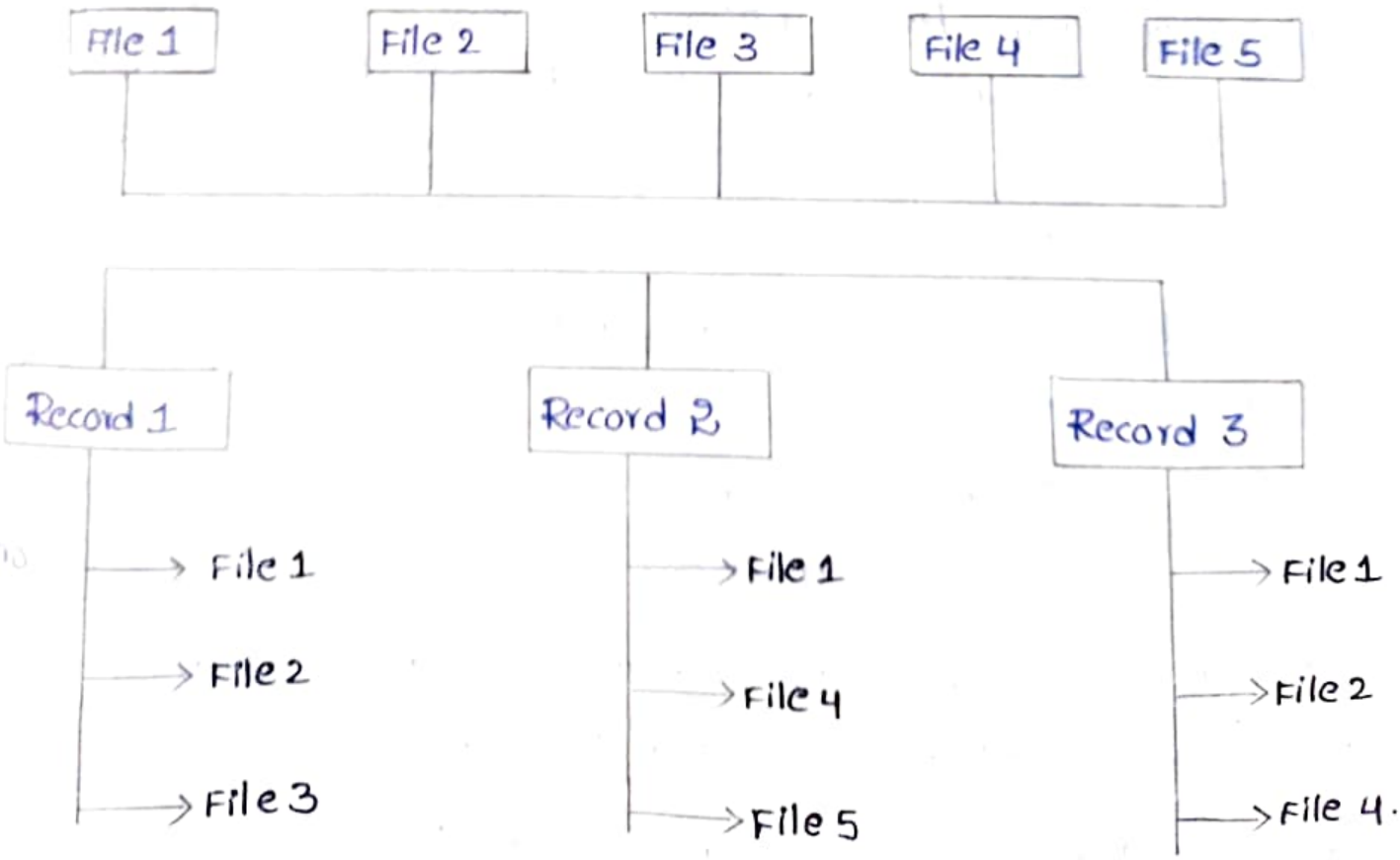
DBMS:

- Data: Data are raw facts the word raw indicates that facts not yet been processed to reveal their meaning.
- End user data: That is raw facts of intersted to the end user.
- Meta data: It is a data about the data.
- Information: It is the result of processing raw data to reveal its meaning.
- Data base: A Data base is a shared integrated computer structure that stores a collection of data.
- DBMS: Data Base Management system it serves as intermediatry between the user and the database. The database structure itself is stored as a collection of files and the only way to acess the data in those files is through DBMS.
- File based system: In the file based system each and every subsyste-m will have its own set of files. File based system is a collect-on of programs that perform service for the end-users such as updating, deleting, inserting.



* Disadvantages of the based system :-

→ Data Redundancy.
→ Data inconsistancy.
→ limited sharing.
→ security problem.
→ Memory wastage.
→ Data isolation.
→ Data Anomiles.

* Data redundancy :- Data redundancy is defined as the storing of the same data in multiple locations also called duplication of data. The result of data redundancy is memory wastage.

* Data inconsistancy :- The result of data redundancy there is a possibili-ty of a particular data will be entered differently in different files which is nothing but inconsistancy of data.

* Data isolation :- The data is available in different files in different formates therefore it is difficult to write a new application progr-am.

* limitted data sharing :- Data are scattered in various files. also different files may have different formate and these files may be stored in different folders may be of different departments so. due to this data isolation. It is difficult to share data among different application.

* security problem :- Data base should be accessible to users in limitted way each.

* Memory wastage :- The data redundancy leeds to memory wastage because we are storing multiple files in multiple locations.

* Data anomalies :- Anomaly is problem it raises when a change doesn't occures at all its required places.

There are three types of data Anomalies.

* updata Anomalies.
* Insertion Anomalies.
* Deletion Anomailes.

* classification of DataBase :-

Data Based on 4 categories the database can be classified into 4 types they are.

① classification of database based on number of users.
② classification of database based on location.
③ classification of database based on structure.
④ classification of database based on usage.

① classification of database Based on number of users :-

Based on users database can be categories into 2 types there are.

① single user database.
② Multi user database.

* single user database :-

Single user database supports only one user at a time These kind of database also called desktop database.

* Multi user data base :-

A Multi user database supports user at a time. Wh the multi user database supports a relatively small number (1-5 -rs) of users or a specific department within an organization. called a work group data base. When the database is used by entire organization and supports many uses across many depar then it is called as enterprise database.

② classification of database Based on location :-

Based on location database can be categoried into types there are.

① Centralized data base.
② Distributed data base.

① Centralized data base :- A data base that supports data located at a single site is called centralized database.

② Distributed data base :- A data base that supports data distributed across serial different sites is called data base.

③ classification of data base Based on usage :-

Based on usage database can be categoried into 2 types there are.

(i) operational database.
(ii) Data ware house.

① operational database :- A database that is designed primarity to support a company's day-to-day operations such kind of databases are called operational database.

② Data ware house :- A data ware house focuses primarity on storing data used to generate information required to make statical or stra-tegic decisions.

④ classification of database Based on structure :-

Based on structure database can be categorized into 2 types there are.

① unstructure data base.
② structure database.

① unstructure data base :-

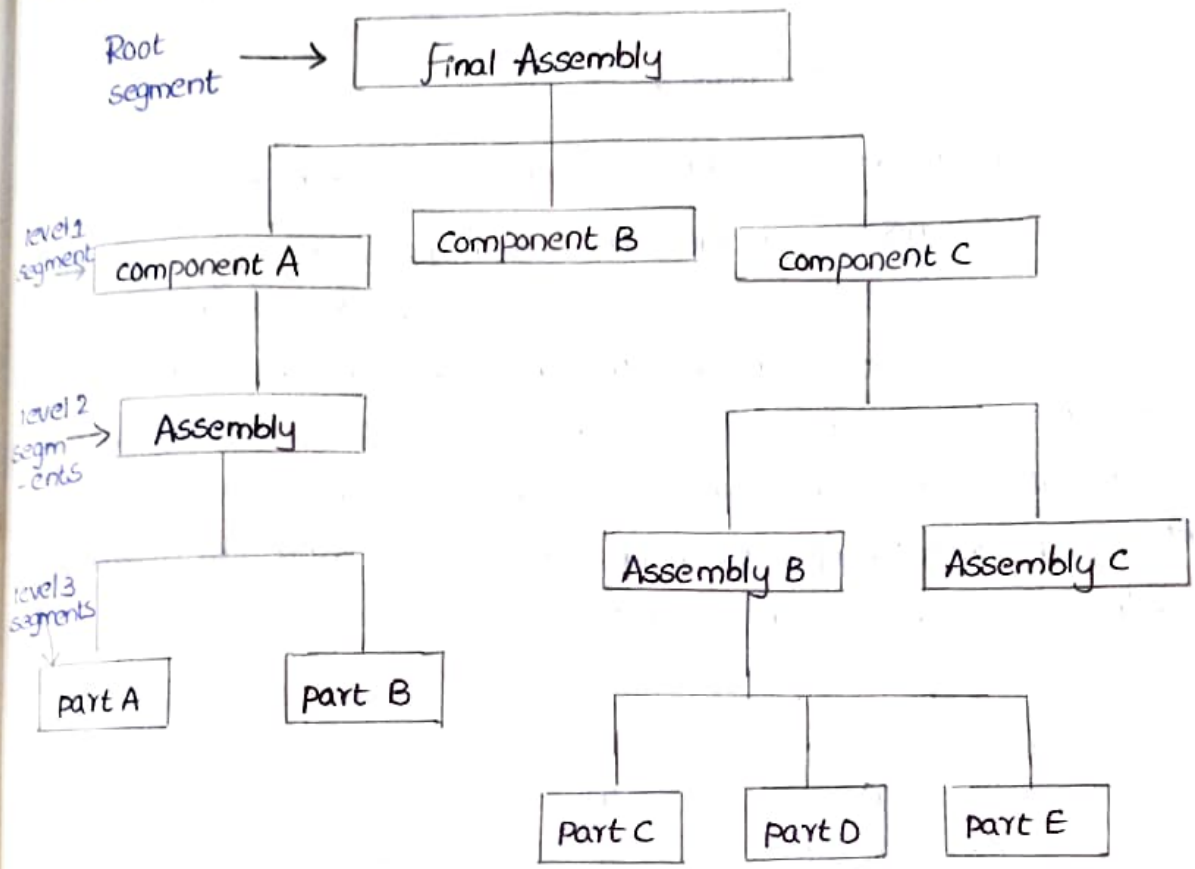
The unstructure database contain raw data.

② structure database :- structure data are result of taking unstructu-red data and formatting data to facilitate such data to generate information.

* What is the hierarchial model ? Explain.

The hierarchial structure contains levels or segments. A segment is the equivalent of a file system's record type.

Within the hierarchy the top layer is perceived as the parent of the segment directly beneath it. The segment below other segments are the childrens of the segments above. In short hierarchial model depicts a set of one-to-many (1:M) relationships between a parent and its children segment.




* Advantages of the hierarchial model:

→ It promotes data sharing.

→ parent child relationship promotes conceptual simplicity.

→ Database security is provided and enforced by DBMS.

→ parent / child relations promotes data integrity.

→ It is efficient with 1:m relationships.

* Disadvantage of hierarchial model:-

→ Complex implementation requires knowledge of physical data storage characterstics.

→ Navigational system yields complex application development manag-ement and use requires knowledge of hierarchical path.

→ change in structure requires changes in all application programs.

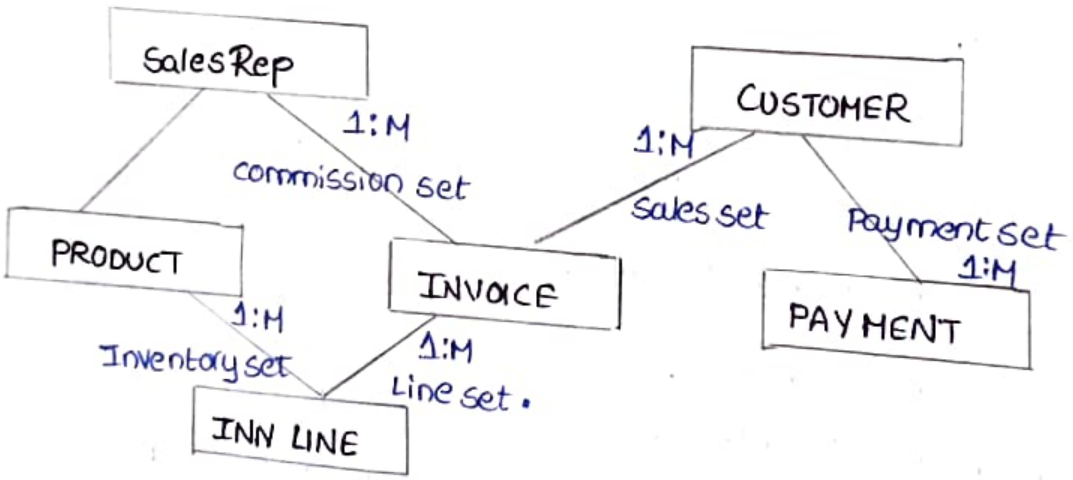
→ There are implementation limitations [No Multi parent or M:N relation -ships].

→ There is no data definition or data manipulation language in the DBMS.

→ There is a lack of standards.

* What is Network model? Explain its advantages and disadvantages?

In the network model the user perceives the network database as a collectional of records in 1:M relationships Howe -er unlike the hierarchical model, the network database terminolo a relationalship is called as set. Each set is composed of a least two record types an owner record and a member record. A set represents 1:M relationship between the owner and the member the following fig shows an example.




In the above figure CUSTOMER, SALESREP, INVOICS, INV-LINE, PRODUCT and PAYMENT REPRESENT record types Note that INVOICE is owned by both SALESREP and CUSTOMER Similarly INV-LINE has two owners PRODUCT and INVOICE Further more the network mod -el can also include one owner relation ships such as customers makes. PAYMENT.

* Advantages of network data model:

→ It handles more relationship types such as M:N and multiprint.
→ Data acess is more flexible.
→ Data owner membership pramotes data integrity.
→ There is conformance of /to standards.
→ It includes data definition language (DDL) and data manipulation language (DBL) in DBMS.

* Disadvantages:-

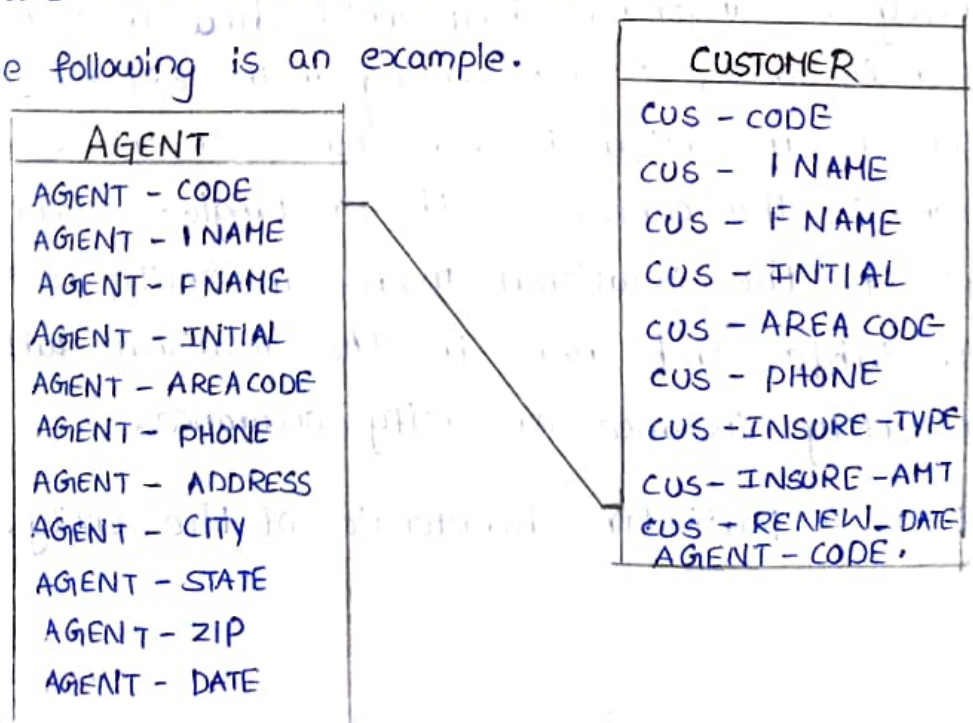
⇒ Navigations system yields complex implementation applicdtion develop -ment and management.
⇒ structure changes require changes in all application programs.

* What is the relational model ? Explain its advantages and Disadvant -ages.

A relational is a matrix composed of intersecting rows and columns each row in a relation is called a tuples. Each column represents an attribute the relational model also describes a precise set of data manipulation constructs based on advanced mathematical concepts.

Today, even microcomputers can run so phisticated relational database software such as oracle, DB2 Microsoft SQL server, My -QL and other maniframe relational software.

The following is an example.



The above relational diagram shows the connecting fields and the relationship type 1:M.

* Advantages :-

→ Structure independence is promoted.
→ Tabular a view substantially improves conceptul simplicity.
→ Ad hod query capability is based on SQL.
→ Powerful RDRMS isolates the end user from physical level details.

* Disadvantages :-

→ The RDMBS requires substantial hardware and system software overhead.
→ Conceptual simplicity gives relatively untrained people the tools to used a good system poorly.
→ It may promote islands of information problems cls individuals and departments can easily develop their own applications.

* What is entity relationship model ? Explain its advantages and disadvantages ?

ER models are normally representing in an entity relationship diagram (ERD) which uses graphical representations to model database components.

The ER model is based on the following components.

Entity :- An entity is anything about which data are to be colle -cted and stored. An entity is representing in the ERD by a rectan -le. also known as an entity box. The name of the entity, a noun is written in the center of the rectangle. usually, wher applying the ERD to the relational model, an entity is mappe to a relational table each row in the relational table is known as an entity instance or entify occurance.

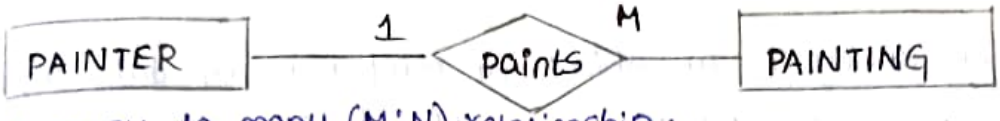
* Attribute :- It is a particular characteristic of the entity.

* Relationship :- Relationships describe associations among data. Most relationships describes associations between two entity.

The following fig shows the different types of relationships

A one-to-many (1:M) relationship.

A PAINTER can paint many PAINTING.




A many to many (M:N) relationship.

An employee can learn many skills.


A one-to-one (1:1) relationalship

An employee manage one STORE.


* Advantages of ER model.

→ Visual modelling yields exceptional conceptual simplicity.

→ Visual representation makes if an effective communication to

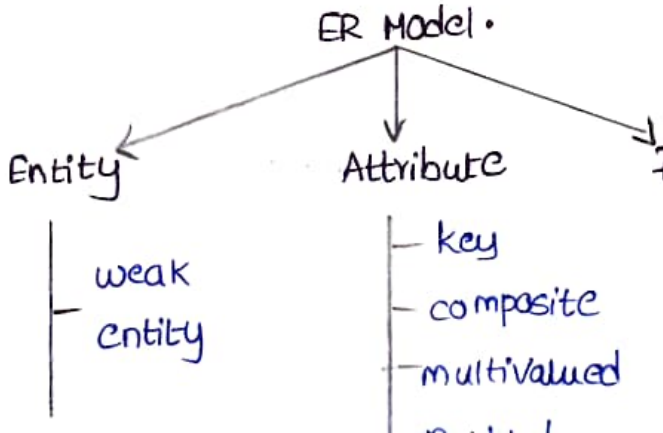
→ It is Integrated with dominant relational model.

* Disadvantages of ER model :

→ There is limited constraint representation.

→ There is limited relationship representation.

→ There is no data manipulation language.



* components of ER Diagram :-
As shown in above diagram, an ER diagram has three main components.

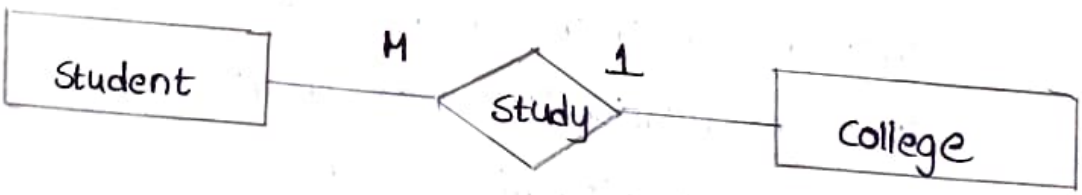
① Entity.
② Attribute.
③ Relationship.

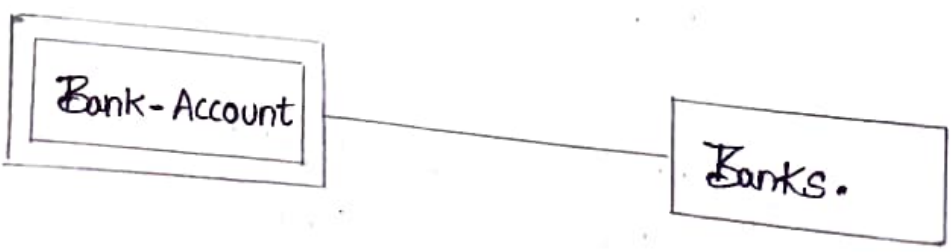
① Entity :- An entity is an object or component of data an entity is represented as rectangular in an ER diagram.

For example :- In the following ER diagram we have two entities student and college and those two entities have many to one relationship as many student study in a single college we will read more about relationships later, for now focus on entities.

→ Entitys are two types they are :-
① strong entity ② weak entity.

* Weak entity :- An entity that cannot be uniquely identified by its own attributies and relies on the relationship with other entit is called weak entity. The weak entity is represented by a double rectangle. For example :- a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.



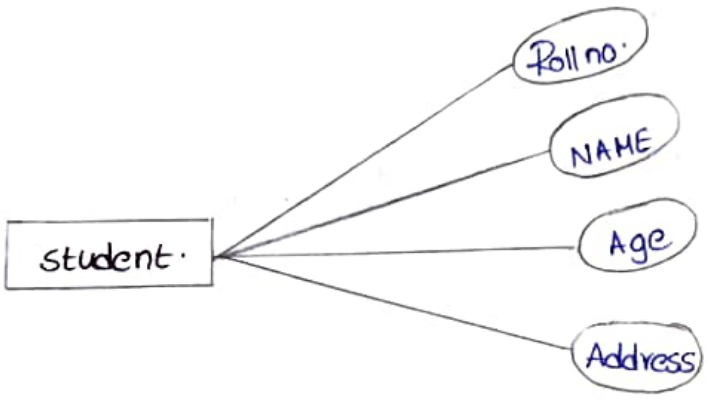



② Attribute :- An Attribute describes the property of an entity an attribute is represented as oval in an ER diagram.

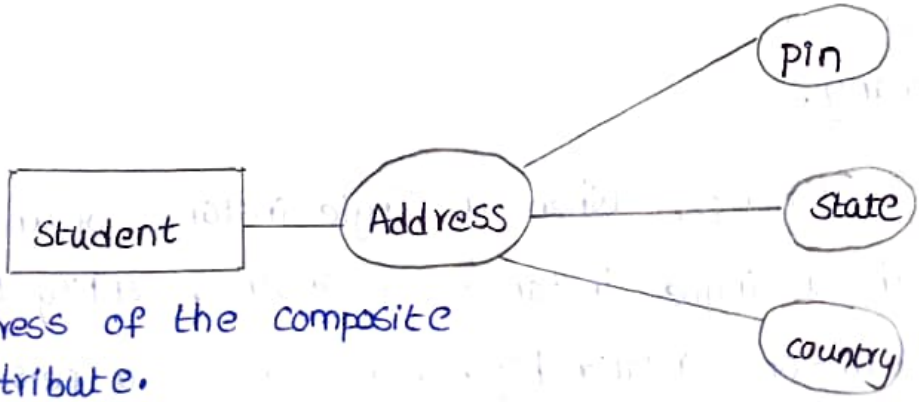
there are four types of attributes.

→ Key attribute.
→ composite attribute.
→ Multivalued attribute.
→ Derived attribute.

① Key attribute :- A key attribute can uniquely identity an enti from an entity set. For example. student roll number can uniquety identify a student from a set of students. Key attr -bute is represented by oval same as other attributes howe the text of key attribute is underlined.




② composition attribute :- An attribute that is a combination of other attributes is known as composition attributes of other -butes ~~is shown~~ for example: In student entity, the study address is a composite attribute as an address is composed other attributes such as pin code, state, country.




Address of the composite attribute.

③ Multivalued attribute :-

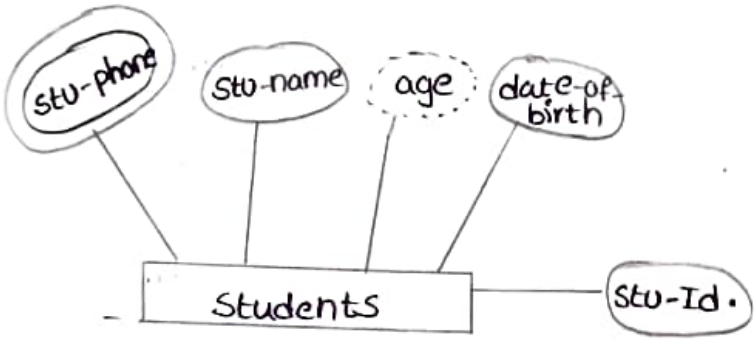
An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in ER diagram.

For example:- A person can have more than one phone numbers so the phone number attribute is multivalued.

④ Derived attribute :- A derived attributes is one whose value is dynamic and derived from another attribute. It is represented by dashed oval in an ER diagram.

For example :- person age is derived attribute as it change over time can be derived from another attribute (Date of birth).

E-R diagram with multivalued and derived attributers.




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

- one to one
- one to many
- many to one
- many to many.

* one to one Relationship :- When a single instance of an entity is associated with a single instance of another entity then it is called one-to-one relationship. For example, a person has

only one passport and a passport is given to one person.


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

For example :- A customers can place many orders but a order cannot be placed by many customers.


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

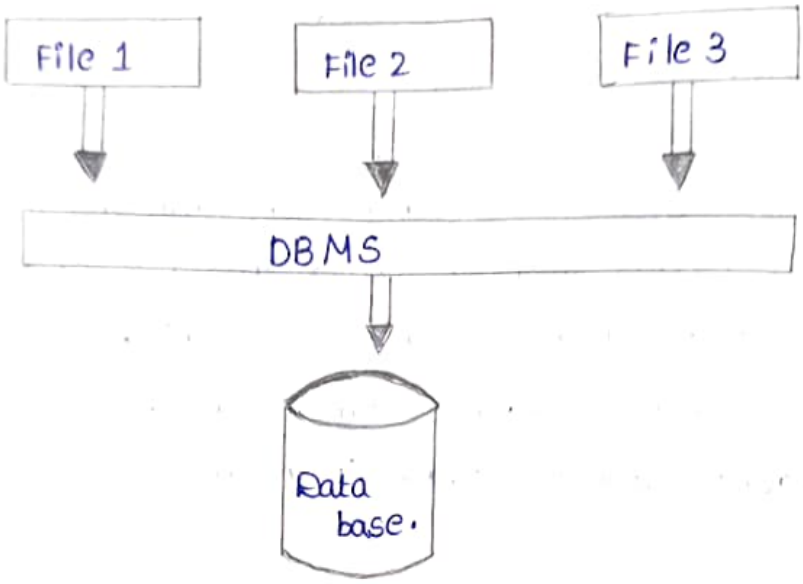
For example :- many students can study in a single college but a student cannot study in many colleges at the same time.


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

For example :- A can be assigned to many projects and a project can be assigned to many students.

* Data base approach:- In order to remove all
based. Approach a new approach was required that must be more effe
-tive known as walabase approach.

The Database is a shared collection of logically related Data, desig
-ned to meet the information needs of an organization. the data base
is a single. large repository of data. which can be used simultane
-ously by many departments and users, Instead of disconnected
files with redundant data, all data items are integrated with a
minimum amount of duplication.




* components of DBMS :-

The data base Mangement system can be divided into four major components, They are :

→ Hardware.
→ Software.
→ Data.
→ procedures.

* Hardware : Hardware consists of a set of physical devices such as computer storage device, input channels, output channels etc.... It is impossible to implement DBMS with out hardware devices.

* Software : A software is set of programs that designed to perform specific task to maintain we need Some software

like O.S, DBMS, etc....

* **Data** :- Data is that resources, for which DBMS was designed. The motive behind the creation of DBMS was to store and utilize data.

* **procedure** :- procedures refer to the instructions and rules that help to design the database and to use the database some procedure are as follows.

→ procedure to install the new DBMS.
→ To log on to the DBMS.
→ To use the DBMS.
→ To make backup copies of database.
→ Three schema of database.

The major purpose of a database system is to provide users with an abstract view of the database. The system hides certain details of how data is stored and created and maintain

The 3 levels are :-

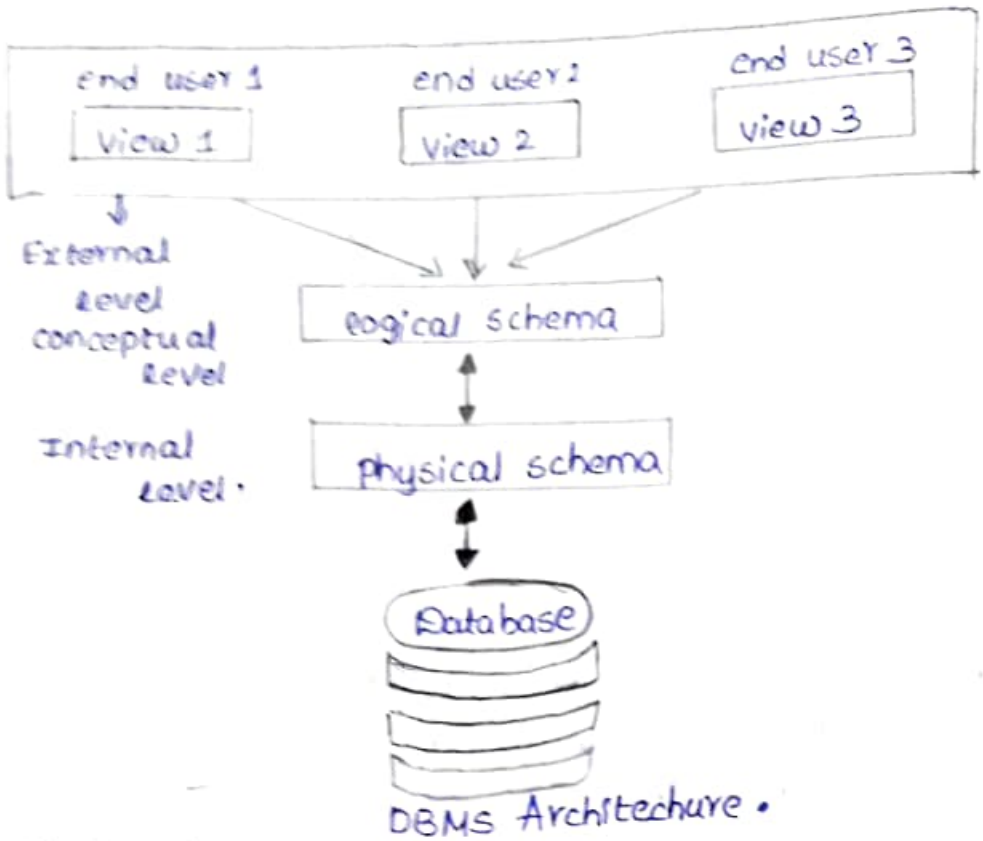
* Internal (or) physical level.
* Logical (or) conceptual level.
* View level (or) External level.

* **Internal level** :- The lowest level of abstraction describe how t data are actually stored. In this level it also describes whi -mat is follows to store the data in database.

* **logical level** :- This level describes what data are stored in data base and what relation ships exist among those dat is used to describes the concept of the data which is in database.

* **View level** :- The system may provide many views for the d manyusers of the database system will not be connected all the information.

* costs and ris




DBMS Architechure.

→ Costs and risks of DBMS :-

① High cost :- Installing a new database system may require investment in hardware and software the DBMS requires more main memory and disk storages. Moreover, DBMS is quite expensive. Therefore a company needs to consider the over head cost of implementing a new database system.

② Training new personnel :- When an organization plans to adopt a database system, it may need to recruit or hire aspecialized data adminstration group, which can cooridinate with different users groups for desigining views, establishing recovery procedures hiring such professionals is expensive.

③ Explicit back up and recovery :- A shared corporate database must be accurate and avaliable at all times. therefore, a system using on a line updating requires explicit backup and recovery procedures.

④ System failure: When a computer system containing the database fails, all user have to wait until the system

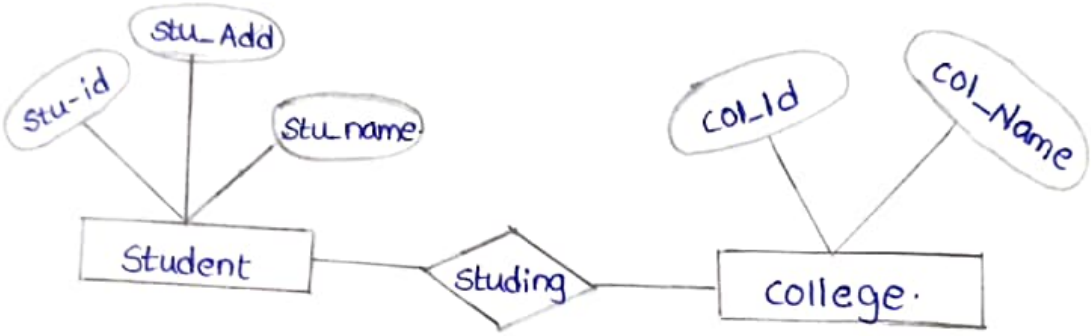
functional again. Moreover, If DBMS or the application program
is, a permanent damage may occur to the database.

UNIT- II

# * ENTITY - RELATIONSHIP MODEL *

* ER MODEL :- An entity-relationship model (ER model) explains the construction of a database with the help of a diagram, called as entity relationship diagram (ER diagram).

Entity relational (ER) model is high-level conceptual data model diagram. It helps you to analyze data requirements systematically and produce a well-designed database. It represents real-world entities and the relationship between them and it is considered as a best practice to complete ER modeling before implementing your database.




* Rectangle :- Represents entity sets.
* Ellipses :- Attributes.
* Diamonds :- Relationship set.
* Lines :- They link attributes to entity sets and entity sets to relationship set.
* Double Ellipses :- Multi valued Attributes.
* Dashed Ellipses :- Derived Attributes.
* Double rectangles :- Weak entity sets.
* Double Lines :- Total participation of an entity in a relationship set.

* ER-Diagram Notations :-

ER-Diagram is a visual representation of data it describes how data is related to each other.

* Rectangles :- This symbol represent entity types.
* Ellipses :- Symbol represent attributes.
* Diamonds :- This symbol represents relationship types.
* Lines :- It links attributes to entity types and entity types with other relationship types.
* primary key :- Attributes are underlined.
* Double Ellipses :- Represent multi-valued attributes.
* Dashed Ellipses :- Derived Attributes.
* Double Rectangle :- Weak entity sets.
* Double Lines :- Total participation of an entity in a relationship set.

 → Entity or strong entity.

 → Weak entity.

 → Attribute.

 → Multivalued Attribute.

 → Relationship.

 → Weak relationship.

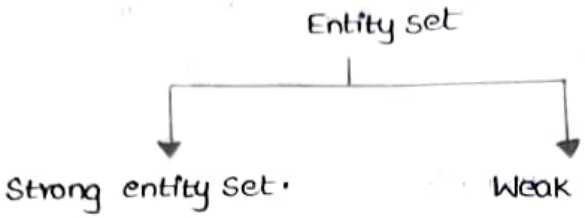
CLASSIFICATION OF ENTITY SETS :-

An entity is an object in the real world with an indepen

-dent existence that can be differentiated from other objects. Entities can be classified based on their strength. It is considered weak if the tables are existence dependent.

* Types of entity sets :-

An entity set may be of the following two types :-




① Strong entity set :- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.

- In other words, a primary key exists for a strong entity set.
- primary key of a strong entity set is represented by underlining it.

* Symbols used

A single rectangle is used for representing a strong entity set.

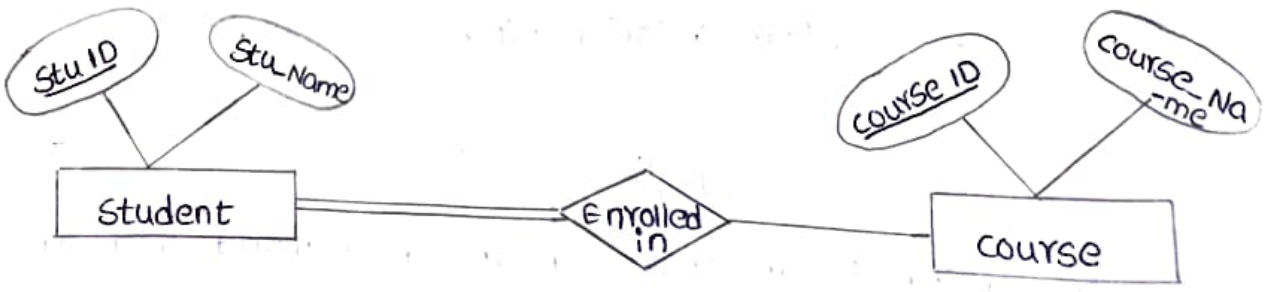
* A diamond symbol is used for representing the relationship that exists between two strong entity sets.

* A single line is used for representing the connection of the strong entity set with the relationship. set.

* A double line is used for representing the total participation of an entity set with the relationship. set.

* Total participation may or not exist in the relationship.

eg :- Consider the following ER diagram.

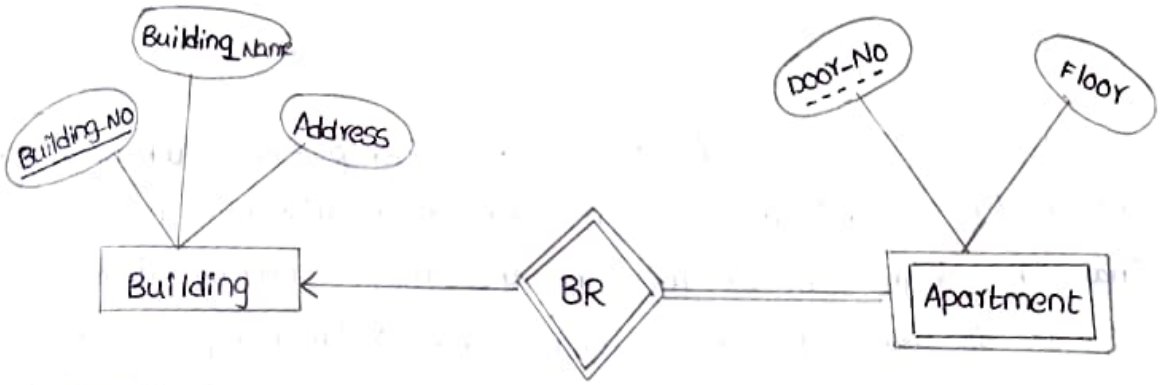


* Weak entity set :- A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities. In other words, a primary key does not exist for a weak entity set. However, it contains a partial key called as a discriminator. It is to identify a group of entities from the entity set and mentioned by underlying with a dashed line.

→ symbols used.

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

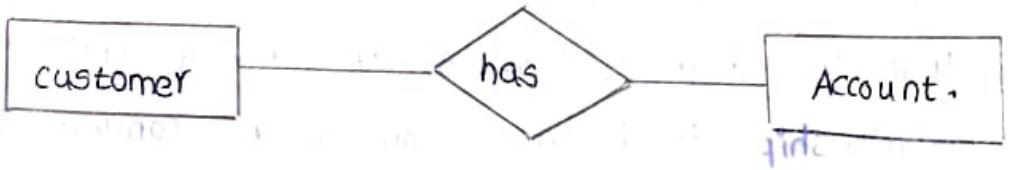
eg:- consider the following ER diagram.




* RELATIONSHIP DEGREE :-

Total number of an entity type and it is connected to relationship is the degree of relationship. The number of entities sets that participate in a relationship set is termed as the degree of that relationship set.

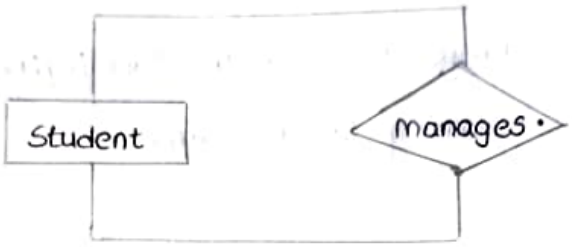
eg:-



Types of Relationship sets.

- Unary Relationship set.
- Binary Relationship set.
- Ternary Relationship set.
- N-ary Relationship set.

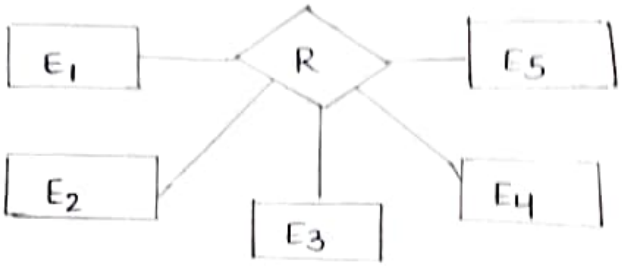
① Unary relationship set :- A unary relationship exists with both the participating entity type are the same. When such a relationship is present, then we called them as degree of relationship is unary relationship.




② Binary relationship set :- A binary relationship exists when exactly two entity type participates. When such a relationship is present, we say that the degree is 2. This is the most common degree of relationship. It is easy to deal with such relationship as these can be easily converted into relational tables.


③ Ternary relationship set :- A Ternary relationship exists when exactly three entity type participates. When such a relationship is present, we say that the degree is 3. As the no. of entities increases in the relationship, it becomes complex to convert them into relational tables.


(4) N-ary relationship set : An N-ary relationship exists when n'numbe-r of entities are participating. so, any number of entities can partici-pate in a relationship. There is no limitation to the maximum no of entities that can participate. But, relations with a higher degree are not common.

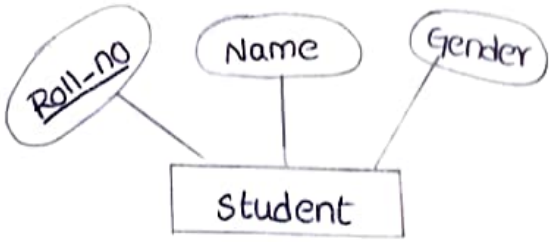



* REDUCING ER DIAGRAM TO TABLES :-

* Rule : 1 For strong entity set with only simple Attributes :-

A strong entity set with only simple attributes will require only one table in relational model. Attributes of the table will be the attributes of the entity set. The primary key of the table will be the key attribute of the entity set.

eg:-



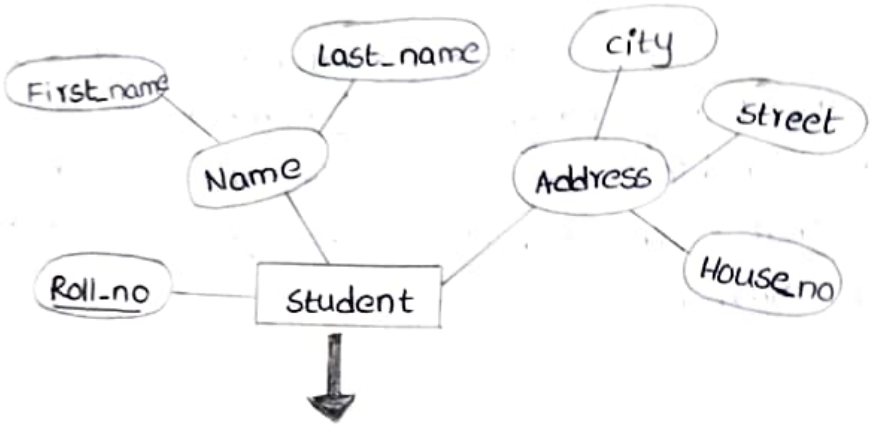

| Roll-no | Name | Gender |
|---|---|---|
| | | |

Schema :- Student (Roll-no, name, Gender).

* Rule-2 :- For strong entity set with composite Attributes :-

A strong entity set with number of composite attributes will require only one table in relational model. while conversion, simple attributes are the composite attributes are taken into account and not the composite attribute itself.

eg :-



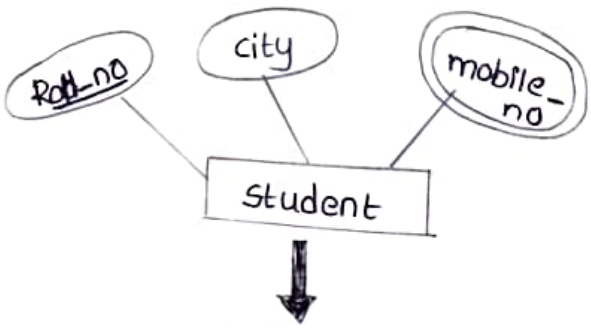

| Roll_no | First_name | Last_name | House_no | Street | city. |
|---|---|---|---|---|---|
| | | | | | |

Schema :- Student (Roll_no, First_name, Last_name, House_no, Street, city)

* Rule-3 :- For strong entity set with multi valued Attributes :-

A strong entity set with any number of multi valued attri-butes will require two tables in relational model. One table will contain all the simple attributes with the primary key. other table will contain the primary key and all the multi valued attributes.

eg :-




| Roll_no | city |
|---|---|
| | |

| Roll_no | Mobile_no |
|---|---|
| | |

* Rule-04: Translating Relationship set into a Table:

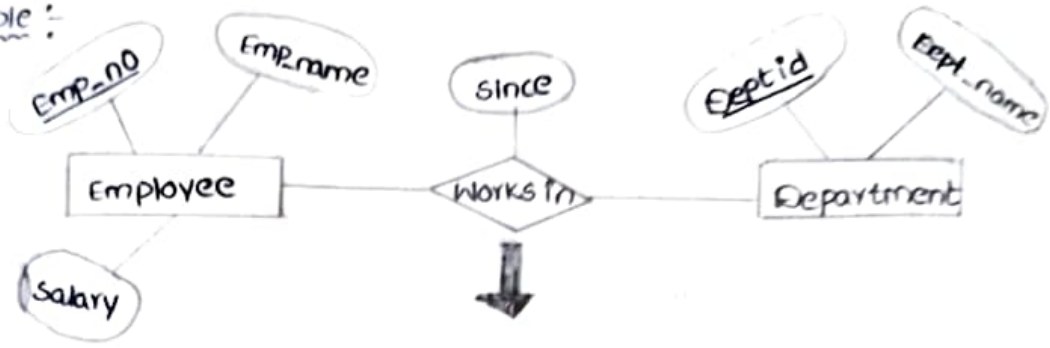
A relationship set will require one table in the relational model.

Attributes of the table are -

- primary key attributes of the participating entity set.
- Its own descriptive attributes if any.

Set of non-descriptive attributes will be the primary key.

Example:




| Emp no | Dept id | since. |
|---|---|---|
| | | |

Schema: Works in (Emp_no, Dept_id, since).

* ENHANCED ENTITY-RELATIONSHIP MODEL (EER MODEL).

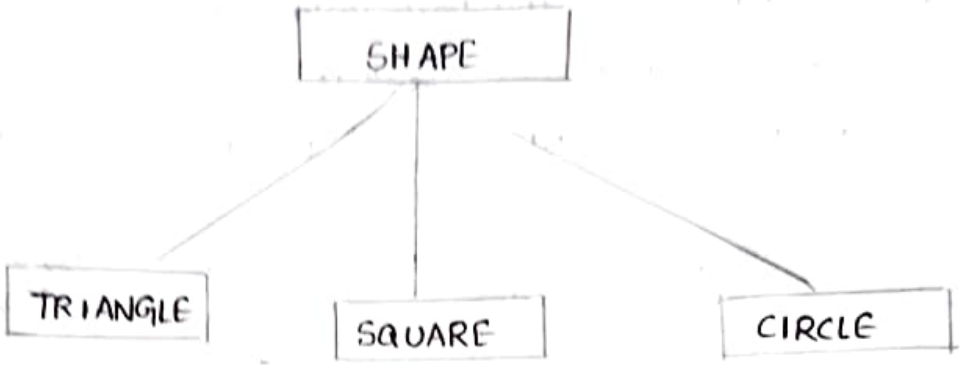
Enhanced entity-Relationship model simply called them as EER, is a high-level data model that incorporates the extensions to the origi ER model. Enhanced ERD are high level models that represent the requirements and complexities of complex database. In addition t ER model concepts EE-R are included. concepts in EE-R diagram They are:-

* Subclasses and super classes.
* Specialization and Generalization.
* Category or union type.
* Aggregation.

* Sub classes and super class :-

Super class is an entity that can be divided into further subtype.

eg:- consider shape super class.

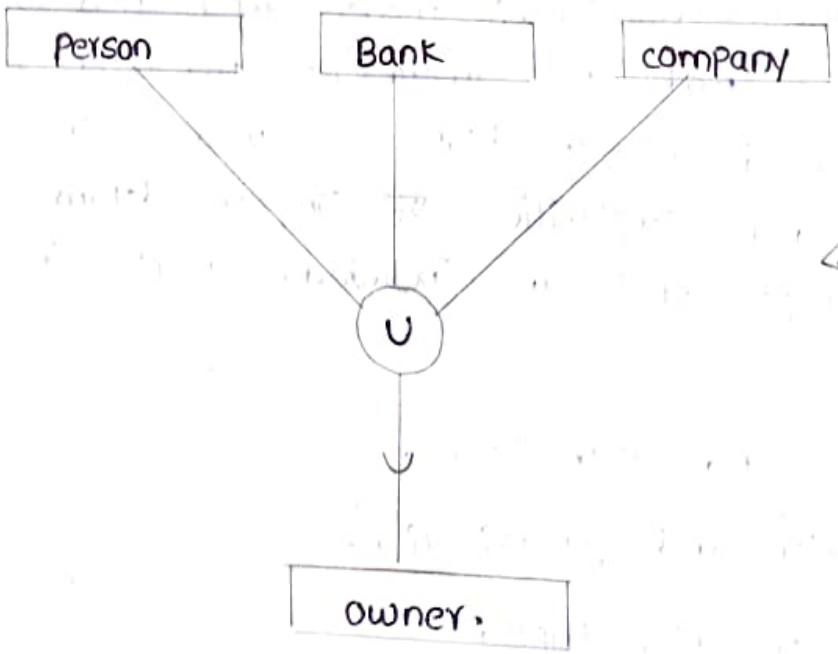



Super class shape has three sub groups. They are Triangle, square and circle. sub classes are the group of entities with some unique attribute sub class inherits the properties and attributes from Super class.

* Generalization
~~Specialization~~ :- It is a process of generating super entity based on common properities of multipble sub entity.

→ In Generalization both subclass and super class shares "is A" relationship.

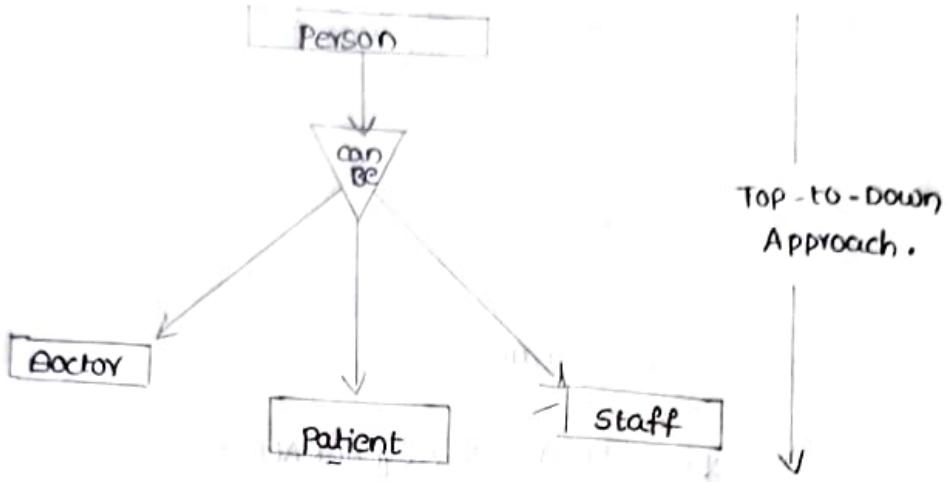
→ Generalization follows bottom to top approach.



* Specialization :- It is a opposite to Generalization. It is a process of seperating multipiable subentity from single super entity.

→ In specialization both subclass and super class shares "isA" relationship.

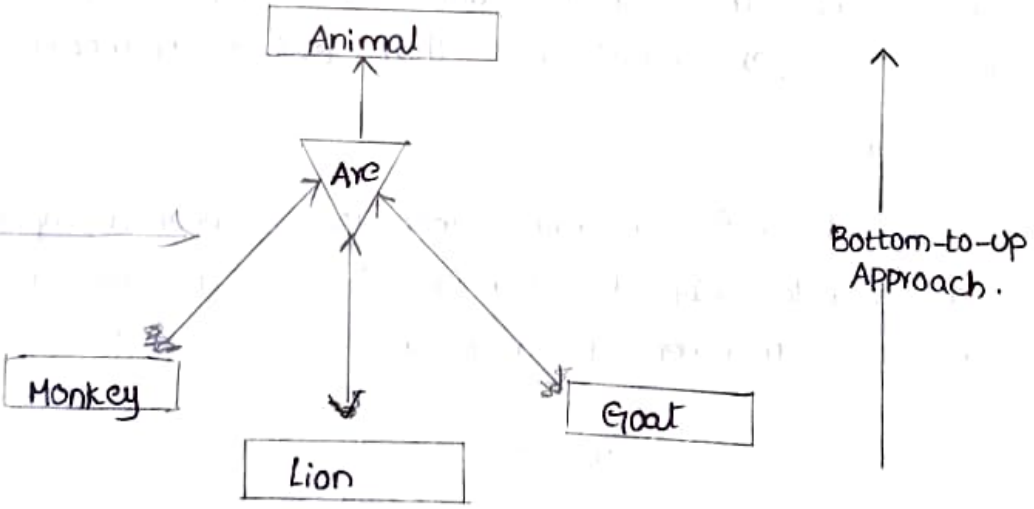
→ Specialization follows top to bottom approach.




* category or union :-

→ category represents a single super class or sub class relationship with more than one super class.

→ It can be a total of partial participation.



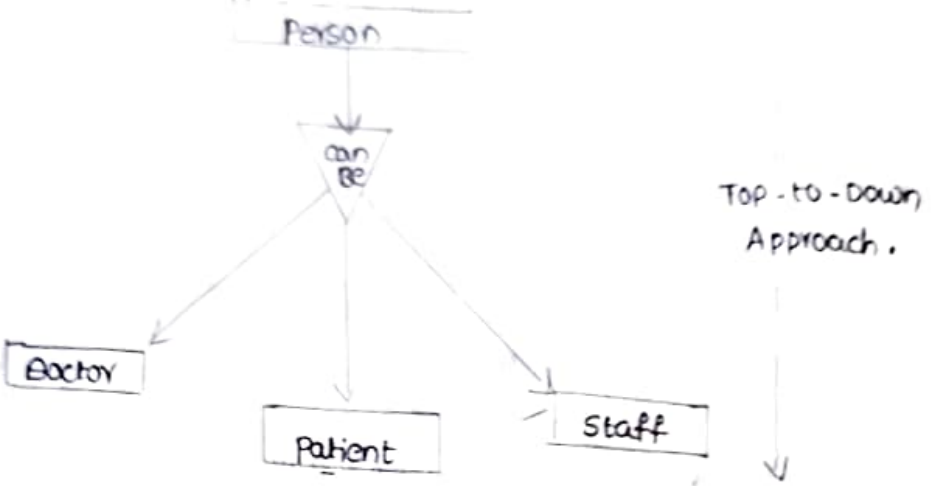

* Aggregation :-

→ Aggregation is a process that represent a relationship between a whole object and its component parts.

* specialization :- It is a opposite to Generalization. It is a process of seperating multipiable subentity from single super entity.

→ In specialization both subclass and super class shares "isA" relationship.

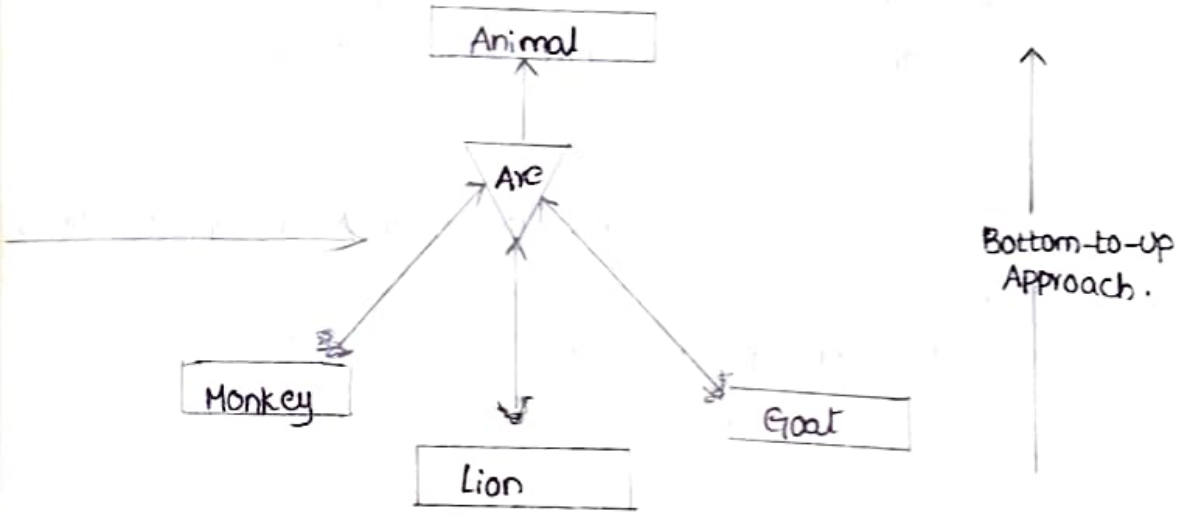
→ Specialization follows top to bottom approach.




* category or union :-

→ category represents a single super class or sub class relationship with more than one super class.

→ It can be a total of partial participation.

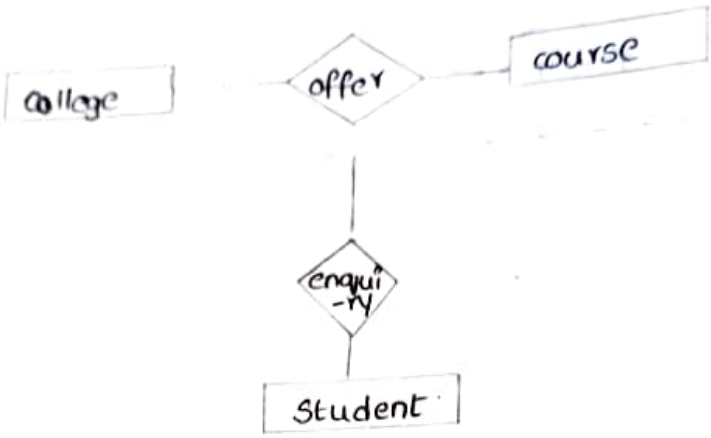



* Aggregation :-

→ Aggregation is a process that represent a relationship between a whole object and its component parts.

→ It abstracts a relationship between obj
-onship as an object.

→ It is a process when two entity is treated as a single entity.




* CONSTRAINTS ON SPECIALIZATION / GENERALIZATION :-

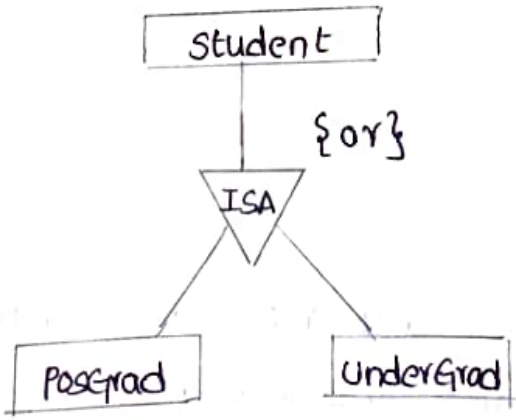
* Membership constraints :-

① condition defined :- Membership of a spedization/ generalization relation
-ship can be defined as a condition in the requirements.

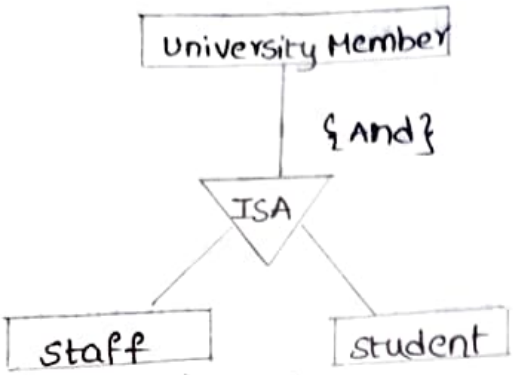
② user defined :- sometimes the designer can define the super class-
sub class relationship. This can be dome to simplify the design model
or represent a complex relationship that exists between entities.

* Disjoint constraints :-

① Disjoint :- The disjoint constraint only applies when a superclass has
more than one subclass. If the subclasses are disjoint, then an entity
occurence can be a member of only one of the subclasses.



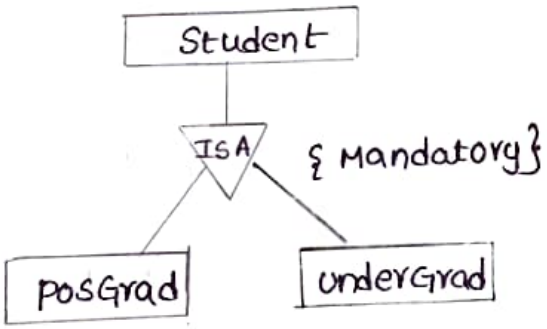
* overlapping:- It applies when an entity occurrence may be a member of more than one subclass, e.g. student and staff some people are both. And is used to represent the overlapping specization/generali-zation relationship in the ER diagram.




* completeness constraints:-

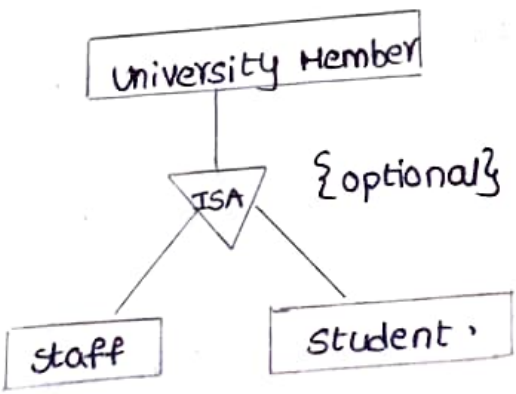
① Total: Each superclass (higher-level-entity) must belong to subclas-es (lower-level entity sets).

eg:- a student must be postgrad or undergrad. To represent comple-eness in the specilalization/ generalization relationship, the keyw-rd Mandatory is used.




② partial: Some super classes may not belong to subclasses (lower-le-entity sets).

eg:- some people at UCT are neither student nor staff. The keyword -onal is used to represent a partial specialization/generalization r-ionship.

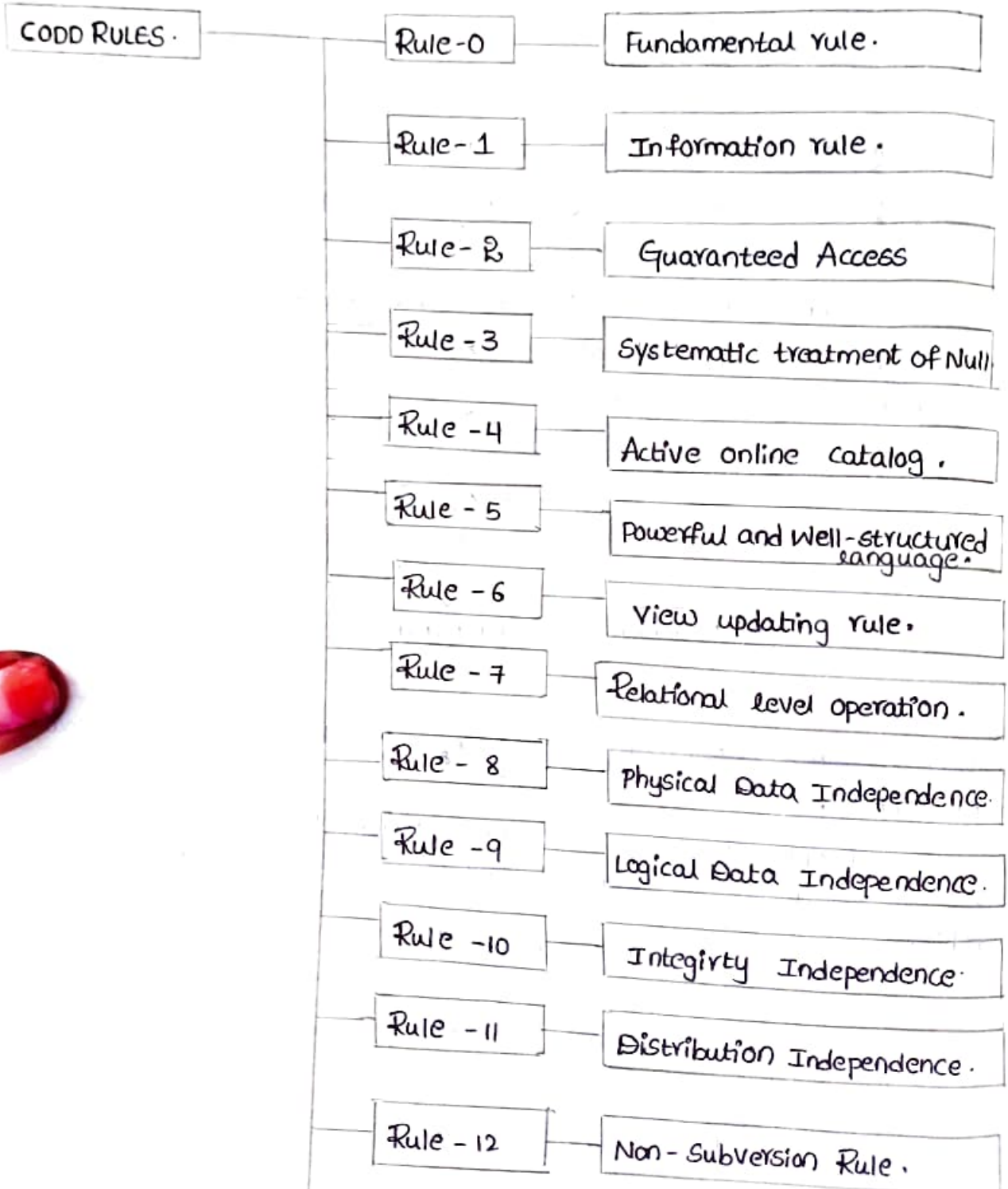


# UNIT-III

## RELATION MODEL.

* **CODD RULES :-** codd proposed 13 rules popularly known as codd's 12 rules to test DBM's's concept against his relational model. Till now there is hardly any commerical product that follows all the 13 codd's rules. even oracle follows only eight and half (8.5) out of 13.

*



* Rule-zero: Fundamental rule :-

This rule states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities.

* Rule-1: Information rule :-

All information to be represented as stored data in cells of tables.

* Rule-2: Guaranteed Access :-

Every single data element is guaranteed to be accessible logically with a combination of table-name, primary-key, and attribute-name (column value).

* Rule-3: Systematic treatment of Null :-

The Null values in a database must be given a systematic and unifo
-m treatment. This is a very important rule because a Null can be interpreted as one the following" data is missing, data is not know or data is not applicable. It should be handled consistenly.

* Rule-4: Active online catalog :-

The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be access by authorized users. users can use the same query language t access the catalog which they can use to access the database -elf. Database dictionary is the structure description of the com -te database and it must be stored online.

* Rule-5: powerful and well-structured language :-

A database can be accessed using a language having linea syntax that supports data definition, data manipulation, and tran -tion management operations.

* Rule-6: View updating Rule :-

All the views of a database, which can theoretically be upd must also be updatable by the system.

* Rule-7: Relational level operation:-

There must be Insert, Delete, Update operations at each level of relations. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

* Rule-8: Physical Data Independence:-

The data stored in a database must be independent of the appli-cations that access the database. Any change in the physical str-ucture of a database must not have any impact on logical data.

* Rule-9: Logical Data Independence:-

The data stored in a database must be independent of the applica-tions that access the database. Any change in the physical structu-re of a database must not have any impact on physical data.

* Rule-10: Integrity Independence:-

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application.

* Rule-11: Distribution Independence:-

A database should work properly regardless of its distribution acro-ss a network. The end-user must not be able to see that the data is distributed over various locations. users should always get the impression that the data is location at one site only.

* Rule-12: Non-subversion Rule:-

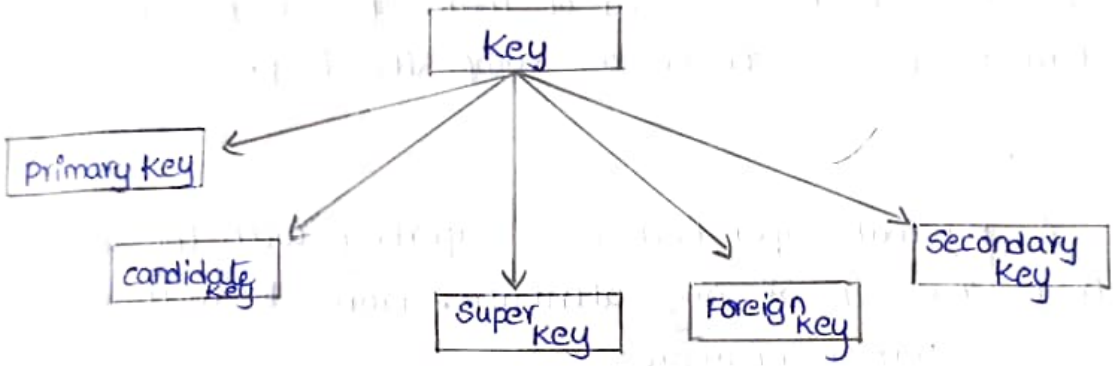
If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change the data.

* CONCEPT OF KEY :-

Key plays an important role in the relational database. It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationship between tables. A key consists of one or more attributes that determines other attributes.

* Types of keys :-

① primary key
② candidate key.
③ super key.
④ Foreign key.
⑤ secondary key.




① Super Key :- Super Key is a set of an at an attribute which can uniquely identify a tuple. Super Key is a superset of a candidate Key. The super key would be EMPLOYEE-ID, (EMPLOYEE_ID, EMPLOYEE-Name) etc.

② candidate Key :- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple. The remaining attributes except for primary key are considered as a candidate key. candidate keys are strong as the primary key.

③ primary Key :- It is the first key which is used to identify one and only one instance of an entity uniquely. An entity can conta multiple key as we saw in PERSON Table. The key which is most suitable from those lists become a primary key.

For each entity, selection of the primary key is based on requirements and developers.

④ Foreign key :- Foreign keys are the column of the table which is used to point to the primary key of another table. Foreign key is used to create relationship between two relations.

* RELATIONAL INTEGRITY :-

* Functional Dependency :- If any attribute dependence on another attribute. The term functional dependency can be defined for "A DETERMINES B". If each value in a column "A" determines only one value in column B.

* Composite key or fully functional Dependency :-

A key may be composed of more than one attribute; such a multi attribute key is known as composite key.

* Minimal super key :-

A Minimal super key is a superkey with the additional property that removal of any attribute from K will cause K not to be a superkey anymore.

{Eno}: is Minimal super key. { A super key which have only one attribute is Minimal super key.}

{voter_id}: is Minimal Super key.

{Eno, Ename}: is Not a Minimal super key { Removal of Ename from {Eno, Ename} = {Eno} is also a super key} hence {Eno, Ename} is not Minimal super key.

* Candidate key :- If a relation schema has more than one key (Minimal Super key) then each of them is called as candidate key.

One of the candidate keys is arbitrarily designated to be the each primary key, and the others are called secondary keys.

A key formed by combining at least two or more columns is called composite key.

* **primary key** :- set of attributes of a relation which uniquely identi-fies a tuple in a relation.

**NOTE** :- A Relation (table) can have many superkeys, and also many minimal superkeys.

If a Relation has more than on minimal superkeys each can be called as candidate keys.

one of the candidate keys is arbitrarily designated to be the prima-ry key, and the others are called secondary keys.

Primary key doesn't allow duplicates and Null values.

* **Foreign key** :- Referential integrity constraint is used to specify inter-dependencies between relations. This constraint specifies a column or list of columns as a foreign key of the referencing table.

A foreign key means the values of a column in one table must also appear in a column in another table. The foreign key in the child table will generaly reference a primary key in the parent table. The referencing table is called the child table & referenced table is called the parent table.

* **secondary key** :-

The secondary key is defined as a key that is used to for data retrieval purpose.

The key which not has be selected as primary key from the pool of candidate key.

# UNIT-3

## * THE RELATIONAL DATABASE MODEL *

* **Relational set operators :-** These operators are applied to the rows -tion and give the result as a relation but do not change the actual relation.

They are 8 types of operators. These operators are called as "SQL comands".

* **UNION :-** It is combines all of the rows in one table with all of the rows in another table except duplicate tuples.

Ex :-

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | D |

→

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | D |

- These two tables are have the same number of columns and same data types.
- The attribute characteristics should be same for two tables. Then the union command is work.

* **INTERSECTION :-** It is combine only rows that appear in both tables (common data).

* **SELECT :-** It is a command to show all the rows in a table.

**NOTE :-** This is also called as "restrict command".

* **PRODUCT :-** It multiplies each row in other table.

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |

| Sub | marks |
|---|---|
| C.S | 70 |
| Mat | 40 |

| I.D | Name | Sub | Marks. |
|---|---|---|---|
| 1. | A | C.S | 70 |
| 1. | A | Mat | 40 |
| 2. | B | C.S | 70 |
| 2. | B | Mat | 40 |
| 3. | C | C.S | 70 |
| 3. | C | Mat | 40 |

* DIVIDE :- These operator gives output in a single column. The two tables are must have a common column. It gives output in a single column which is common in two (both) tables.

| code | Marks |
|---|---|
| A | 5 |
| A | 9 |
| B | 5 |
| C | 9 |
| B | 4 |

| code |
|---|
| B |

| Marks. |
|---|
| 5 |

* JOIN :- It is used to combine columns from two or more tables.

* Types of joins :-

① Inte Inner Join.

② Outer Join

* Inner Join :- It is a simple Join. It is joins data which is matc -hed.

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | D |

| I.D | Address |
|---|---|
| 1. | NRPH |
| 2. | AKP |
| 3. | MVP |

| ID | Name | I.D | Address |
|---|---|---|---|
| 1. | A | 1 | NRPH |
| 2. | B | 2 | AKP |
| 3. | C | 3 | MVP. |

a) Nature join :- It is a type of inner join. It Joins from two tables. which columns and data types are same.

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |
| 4. | D |

| ID | Address |
|---|---|
| 1. | NRPH |
| 2. | AKP |
| 3. | MVP |

| ID | Name | Address |
|---|---|---|
| 1. | A | NRPH |
| 2. | B | AKP |
| 3. | C | MVP. |

* Outer Join :- Types of outer join

1) Right outer join
2) Left outer join
3) Full outer join.

* Right outer join :- It gives the data which is matched from two tables. The data which is not matched with right table then it displays null.

| ID | Name |
|---|---|
| 1. | NRPM |
| 2. | AKP |
| 3. | MVP |

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 4. | C |

| ID | Name | ID | Name |
|---|---|---|---|
| 1. | NRPM | 1 | A |
| 2. | AKP | 2 | B |
| Null | Null | 4 | C |

* Left outer Join :- It gives the data which is matched from two tables. The data which is not matched with left table then it displays Null.

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |

| I.D | Address |
|---|---|
| 1. | NRPM |
| 2. | MVP |
| 4. | AKP |

| I.D | Name | I.D | Address |
|---|---|---|---|
| 1. | A | 1 | NRPM |
| 2. | B | 2 | MVP |
| 3. | C | Null | Null |

* Full outer join :- It writes the data which is matched from two tables, the remaining rows of both left and right tables becomes null.

| ID | Name |
|---|---|
| 1. | A |
| 2. | B |
| 3. | C |

| I.D | Address |
|---|---|
| 1. | NRPM |
| 2. | MVP |
| 4. | AKP |

| ID | Name | I.D | Address |
|---|---|---|---|
| 1. | A | 1 | NRPM |
| 2. | B | 2 | MVP |
| 3. | C | Null | Null |
| Null | Null | 4 | AKP |

* DIFFERENCE :- It gives the values which are present in first set but not in second set.

* PROJECT ($\Pi$) :-

It project the columns which are satisfy the given predict.

* **Relational calculus** :- It is a non-procedural language because it doesn't follow any procedures to produce output.
* By using Relational calculus we are not providing any sol'n for a problem,
* By using this we are conveying user problem.
* Relational calculus tells us what to do but never tells how to do.
* It provides a notation which helps to formulating the desired relation.
* Relational calculus is close to normal language.

There are two types in Relational calculus.

① Tuple RC ② Domain R.C

## ① Tuple Relational calculus :-

* In TRC we are using tuples to generate a notation.
* The result can be one tuple / more than one tuple.
* In TRC we are using two symbols to generate a notation those are ∃ (existancy) - related to, ∀ for all (universal quantifier).

Notation Syntax :- $\{ T / P(T) \}$

where, T = resultant tuple

P(T) = condition which helps to

| ID | N |
|---|---|
| 1 | A |
| 2 | B |

$t_1$

$\{ T.Name \mid t_1 \text{ AND } t_1.Id = 1 \}$

$\{ R \mid \exists T \forall (t_1.Id = 1 \text{ AND } R.Name = T.Name) \}$

## ② Domain Relational calculus :-

* In DRC we are generating a notation with the help of domain.
* In DRC we are using two symbols ∀, ∃.
* In DRC we are using logical operations like AND (&&), OR (||), NOT (!)

| logical operator | Bitwise operator |
|---|---|
| && | & |
| \|\| | \| |
| ! | ∧ |

2 | 4
2 | 2 - 0
1 - 0

100

2 | 5
2 | 2 - 1
1 - 0

101

100 - False
101 - T
100

* DRC Notation syntax :-

$\{(< a_1, a_2, a_3, a_4 >) | (a_1, a_2, a_3)\}$

1st part $< a_1, a_2, a_3, a_4 >$ - set of resultant attributes.

2nd part $(a_1, a_2, a_3)$ - condition to get result.

$\{ < id, name, age > \in std \wedge id = 1 \}$.

* FUNCTIONAL Dependency :-

* If an attribute depends on another attribute for its existance in a relation is called Functional dependency.
* Arrow ($\rightarrow$) is used to represent the F.D.
* The attribute which present at right side of the arrow is called Dependent.
* The attribute which exists as left side of the arrow is called Determinent.

$$X \longrightarrow Y.$$

* In the above expression the dependent 'y' is depends on determinant 'x'.

→ There are FIVE types in F.D they are.

i) Trival F.D ii) Non-Trival F.D. iii) Multivalued F.D

iv) Fully F.D v) Transitive F.D.

i) Trival F.D :-

If dependent is a subset of determinant such types of dependency are called T.F.D.

| Father | Mother | child |
|---|---|---|

$\{f, m\} \rightarrow (c)$

$\{f, m\} \rightarrow (m)$

ii) Non-Trival F.D :- The dependent is strictly not a subset of determinant it is called N-TF.D.

iii) Multivalued F.D :- More than one unrelated dependence are depends on same determinant.

| Father | Mother | child |
|---|---|---|

| Father | Emp |
|---|---|

$F \rightarrow E$

$F \rightarrow E \quad F \rightarrow \{C, E\}$

-te.

* Transitive :- If an attribute depends on one attribute with the help of another attribute.

$$A \rightarrow B = A \rightarrow C$$
$$B \rightarrow C$$

In the above expression transitive dependency is exists because 'c' is depends on A with the help of B.

* Normalisation :-

It is a process of deleting redundant data from a relation.

There are 7 types.

1. First Normal form
2. second Normal form
3. Third Normal form
4. BCNF (3.5 NF) (Boyce codd)
5. Fourth Normal Form
6. Fifth Normal Form
7. Domain key Normal form (DKNF).

(1) First Normal form:-

FNF must satisfy following 2 conditions.

condition-1:- In a relation no two attributes contain same name but we can create with same data type.

condition-2:- Data in a attribute must be same it must be not unique.

* In FNF we are dealing with Multivalued attributes.

* Every attribute in a relation must conetain atomic value after completion of FNF.

| Id | Name | phn |
|---|---|---|
| 1 | A | **** **** |
| 2 | B | **** **** |

| Id | Name |
|---|---|
| 1 | A |
| 2 | B |

| Id | Phone |
|---|---|
| 1 | **** |
| 1 | **** |
| 2 | ***** |
| 2 | **** * |

| Id | Name | phone |
|---|---|---|
| 1 | A | 99999 |
| 1 | A | 88888 |
| 2 | B | 77777 |
| 2 | B | 66666 |

(normalised).

(ii) Second Normal Form:-

In this we are dealo dealing with Non-prime attributes.

Non-prime attribute:- Every attribute in a relation should not be primary key.

* It is a part of candidate key.
* Every non-prime attribute must be dependent on prime attribu-te.

| S-ID | Sub | F-Name |
|---|---|---|
| 1 | C | A |
| 2 | J | B |
| 3 | C | A |

| S-ID | Sub |
|---|---|
| 1 | C |
| 2 | J |
| 3 | C |

S-ID → sub

| S-ID | F-Name |
|---|---|
| 1 | A |
| 2 | B |
| 3 | A |

S-ID → F-Name.

(iii) Third Normal Form:-

In this we are eliminating the transitive dependency.

when it satisfies SNF then only TNF satisfies.

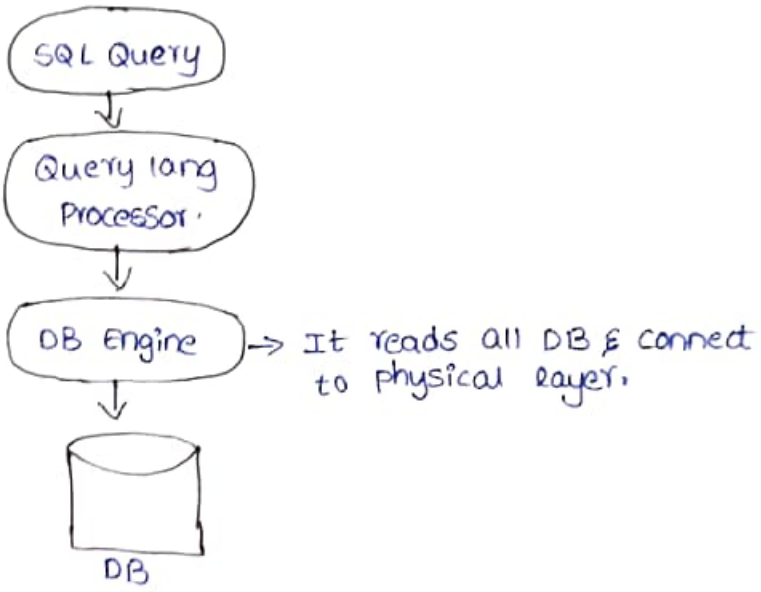
(iv) Boyce codd Normal form| 3.5 NF:-

BCNF is also called as 3.5NF because it is the advanced version of 3NF.

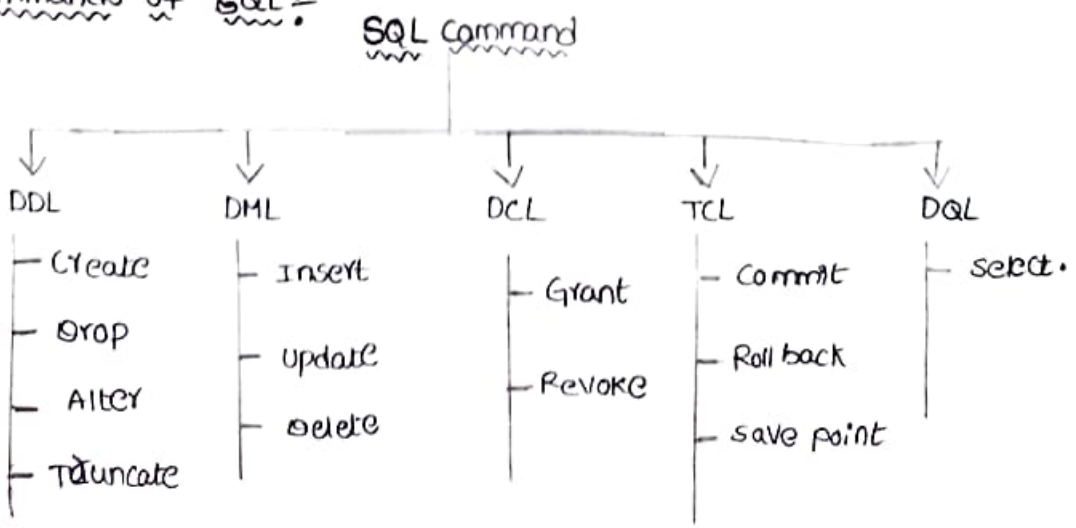
# UNIT - IV

## STRUCTURED QUERY LANGUAGE

* History of SQL standard :-
* SQL is a relational data base management system which is used to access the data. It is not a case-sensitive.
* Father of SQL - Dr. Edger Frank ted, codd.
* SQL is developed in IBM corporation.
* He completed PHD (1964).
* In 1970 he announced officially 'RDB'.
* He proposed 12 codd rules in 1974.
* In 1976 IBM corporation developed codel rules.
* Working title for SQL by IBM is "SystemIR"
* In 1980 first SQL module released into market.
* SQL is mother of all RDBMS.




* Commands of SQL :-



① DDL :-

* These commands are used to define a structure.
  - → create
  - → alter
  - → truncate
  - → drop.

* Create :- used for creation of new structure.

Syntax :- create table table name (Attribute name1 Datatype (size), Attribute name n Datatype (size));

eg :- create table stu (ID number (5), Name varchar2 (10));

Output :-

stu

| ID | Name |
|---|---|
| | |

* alter :- Alternate the defined structure is known as alter.
  - → Alter table add
  - → Alter table modify
  - → Alter table drop

} 3 types.

① Alter table add :- It is used to add a new attribute name in an existing table.

Syntax :- Alter table tablename add (attribute names datatype (size), attribute name datatype (size));

eg :- Alter table stu add (address varchar2 (20), phno number (10));

Output :-

stu

| ID | Name | address | phno |
|---|---|---|---|
| | | | |

② Alter table modify :-

It is used to alter the behaviour of an attribute.

Syntax :- Alter table table name modify (old attribute name datatype (new size), ---);

eg :- Alter table stu modify (address varchar2 (30));

Output :-

| ID | Name | address | phno. |
|---|---|---|---|
| | | | |

③ Alter table drop :- To delete a column in a table.

syntax :- Alter table table-name drop column_name 1, ...;

eg :- Alter table stu drop Phone_no;

* Truncate :- It is used to delete the data inside the table.

syntax :- Truncate table table name;

eg :- Truncate table stu;

* Drop :- It is used to delete a structure in a database.

Syntax :- drop table table name;

eg :- drop table stu;

* DML commands :- (Data Manipulation language) :- It is used to ma -nipulate the data stored in the table.

→ insert into
→ updata
→ delete

} 3 types.

① Insert into :- It is used to insert new records in a table.

1st Syntax :- insert into table name values (value1, value 2).

eg :- insert into stu values (1, "ABC", "Tuni", 9876543210).

| ID | Name | address | phno. |
|---|---|---|---|
| 1 | ABC | Tuni | 9876147610 |

2nd syntax :- insert into table name (attribute1, attribute 2---) values (value1, value 2);

eg :- insert into stu (ID, Name, ~~attribut~~address, phno) values (2, "DEF", vizag, 11111111);

| ID | Name | address | Phno. |
|---|---|---|---|
| 1 | DEF | Vizag | 11111111 |

3rd syntax :- insert into table name values (& attribute name1, & attri -bute name 2 - --);

eg:- Insert into stu values (&ID, &name, &add, &phone);

| ID | Name | address | phno |
|---|---|---|---|
| 1 | GHI | vizag | 9686761482 |
| 4 | XYZ | Tuni | 9296827642 |

* Update :- It is used to update the content of a table.

* Syntax :- update table-name set column name = value where (condition);

eg :- update stu set address id=4 where id=1;

| ID | name | address | phno |
|---|---|---|---|
| 4 | GHI | vizag | 7894... |

* Delete :- It is used to delete all the records or selected records in a table.

1 Syntax :- delete from tablename;

eg :- delete from stu;

2nd Syntax :- delete from table name where condition;

eg :- delete from stu where address = "tuni";

* DCL (Data control language) :-

The commands used to grant and take back authority from any database user.

① Grant ② Revoke.

① Grant :- Grant command is used to privilage to user.

Syntax :- Grant privilage name on table-name to user name;

eg :- Grant priviliage name on stu to u1, u2, u3;

② Revoke :- Revoke command is used to get back the privilage from the user.

Syntax :- Revoke privilage name on table-name from user name;

eg :- Revoke privilage name on stu from u1, u2, u3;

* TCL (Transfer control language) :-

Manage their changes made by dml statements.

① commit ② Roll back ③ save point.

① commit :- It is used to save the changes perminantly.

Syntax :- commit ;

② Roll back :- It is used to roll back to the previous modifications.

Syntax :- Roll back ;

③ save point :- It is used to save the changes temporarily.

Syntax :- save point name ;

* DQL commands (Data Query language)

It is used to fetch the data from the table based on condition.

① select :- It is also known as restrict commond.

Select command is used to display all the data from the table which satisfy the given condition.

Syntax :- select * from table_name where condition ;

eg :- select * from stu where ID=2 ;

* Selection operation :-

The select operation is used for selecting a subset of the tuples according to a given selection condition. selection operator (?) is a unary operator in relational algebra that performs a selection opera-tion.

$\sigma p(r)$

$\sigma$ is the predicate

r stands for relation which is the name of the table.

p is prepositional logic.

Syntax :-

$\sigma$ < selection_condition > (R).

* examples :-

- Select tuples from a relation "Books" where subject is "database".

  $\sigma$ subject = "database" (Books).

* Some imporant points :-

- We may use logical operators like $\wedge, \vee, !$ and relational operators like $=, \neq, >, <, <=, >=$ with the selection condition.
- Selection operator only selects the required tuples according to the selection condition.
- It does not display the selected tuples.
- To display the selected tuples, projection operator is used.
- Selection operator always selects the entire tuple.
- It cannot select a section or part of a tuple.

* PROJECTION OPERATION :-

Projection operator ($\Pi$) is a unary operator in relational algebra that performs a projection operation. It displays the columns of a relation or table based on the specified attributes.

Syntax :- $\Pi$ < attribute list > (R).

Example :- consider the following student relation.

| ID | Name | subject | Age |
|---|---|---|---|
| 100 | Parnika | Maths | 19 |
| 200 | Rahul | Science | 20 |
| 300 | Srisai | physics | 20 |
| 400 | Sameer | chemistry | 21. |

- Now we need Name and age from the above then Result for Query $\Pi$ Name, Age (student)

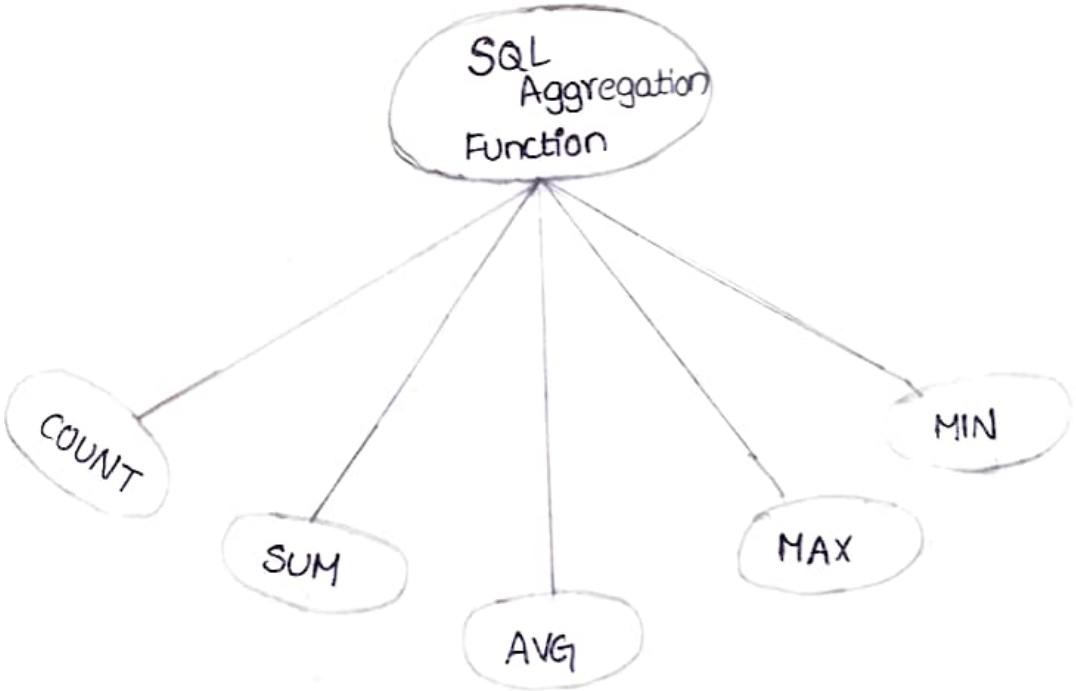
| Name | Age |
|---|---|
| Parnika | 19 |
| Rahul | 20 |
| Srisai | 20 |

- projection operator automatically removes all the duplicates while projecting the output relation.
- If there are no duplicates in the original relation, then the cardinality will remain same otherwise it will surely reduce.
- If attribute list is a super key on relation R, then we will always get the same number of tuples in the output relation.
- There is only one difference between projection operator of relationalgebra and SELECT operation of SQL.
- projection operator does not allow duplicates while SELECT operation allows duplicates.

## * Aggregate functions :-

It is applied on single column on the table and produce the result as a single value.

* count.
* sum.
* AVG.
* MIN.
* MAX.
* count(*).



* COUNT :- It can be applied on every attribute.

- Count function returns the total number of values in the specified field.
- Works on both numeric and non-numeric data types.

eg :- SELECT COUNT ('movie_id') FROM 'Movierentals'
WHERE 'movie_id' = 2;

Output :- COUNT ('movie_id')
3.

* DISTINCT :-

- The Distinct keyword that allows us to omit duplicates from our results. This is achieved by grouping similar values together.
- To appreciate the concept of Distinct, lets execute a simple query.

Example :- SELECT 'movie_id' FROM 'movierentals';

output :- movie_id
1
2
2
2
3

* MIN FUNCTION :-

The MIN Function returns the Smallest Value in the Specified table field. Use My SQL's MIN function to get the desired infor-mation.

Example :- Select Min (column name) from (table name).

* MAX FUNCTION :-

MAX Function is the opposite of the MIN function. It returns the largest value from the specified table field. Let's assume we want to get the year that the latest movie in our database was released.

eg :- SELECT MAX ('Year_released') FROM 'movies';

* AVG function :-

AVG Function returns the average of the values in a specified column. Just like the SUM function, it works only on numeric data -types.

eg :- Select AVG (column name) from table name;

* TABLE MODIFICATION COMMANDS :-

The SQL ALTER TABLE command is used to add, delete or modi -fy columns in an existing table. use the ALTER TABLE comm -and to add and drop various constraints on an existing table.

Syntax :- ALTER TABLE table_name ADD column_name datatype;

eg :- ALTER TABLE $T_1$ ADD Name varchar2;

* The basic syntax of an ALTER TABLE command to change the DATA TYPE of a column in a table is as follows.

Syntax :- ALTER TABLE table_name MODIFY COLUMN column_name datatype;

eg :- ALTER TABLE $T_2$ MODIFY COLUMN ID datatype;

* Syntax :- ALTER TABLE table_name MODIFY column_name datatype NOT NULL;

eg :-

* Syntax :- ALTER TABLE table_name.

ADD CONSTRAINT MY UNique constraint UNIQUE (column1, column2..);

eg :-

* Syntax :- ALTER TABLE table_name.

ADD CONSTRAINT Myunique constraint CHECK (CONDITION);

eg :-

- the basic syntax of an ALTER TABLE command to ADD PRIMARY key constraint to a table is as follows.

Syntax: ALTER TABLE table_name
ADD CONSTRAINT MY primary key PRIMARY KEY (column1, column2...);

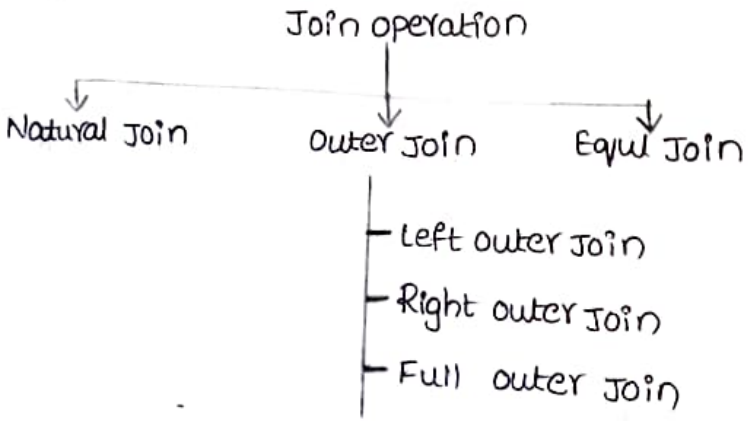
eg:-

* JOIN OPERATION:-

Join is a combination of a cartesian product followed by a selection process. A Join operation pairs two tuples from different relations, if and only if a given join condition is satisfied. SQL Joins clause is used to combine records from two or more tables in a database.

* Syntax:- Select column_name_list
FROM table 1 CROSS Join table 2;

* Types of Join operations:-




* Natural Join:- Natural Join is the set of tuples of all combinations in R and S that are equal on their common attribute names. It is denoted by ⋈.

* Syntax:- Select * From.
table 1 NATURAL JOIN table 2;

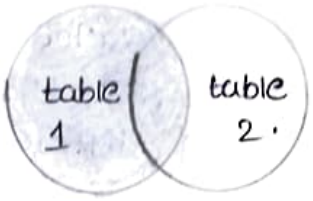
* Query :- $\Pi$ EMP-NAME, SALARY (EMPLOYEE ⋈ SALARY)

SELECT EMP-NAME, SALARY from EMPLOYEE NATURAL JOIN SALARY;

② Outer Join :- Outer join operation is an extension of the join operation and it is used to deal with missing information.

(a) Left outer join
(b) Right outer join
(c) Full outer join

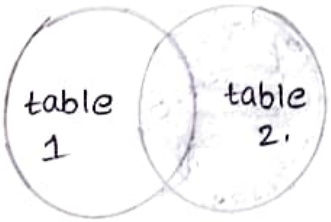
(a) Left outer Join :- Left outer join contains the set of tuples of all combinations in R and S that are equal on their common attribute names. In this tuples in R have no matching in S and it is denoted by ?




Syntax :- Select column-name-list FROM
table 1 LEFT OUTER JOIN table 2
ON table1.column-name = table 2.column-name;

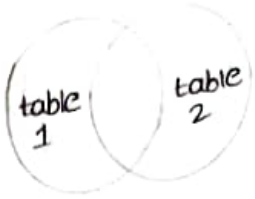
(b) Right outer Join :-

Right outer join contains the set of tuples of all combinations in R and S that are equal on their common attributes names and tuples in S have no matching in R and it is denoted by ?




Syntax :- SELECT column-name list FROM
table 1 RIGHT OUTER JOIN table 2
ON table 1.column-name = table 2.column-name;

* Full outer Join :- Full outer Join is left or right Join except that it contains all rows from both tables. In this Tuples in R have no matching into S with their common attribute name are denoted by ⋈.

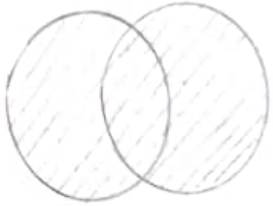



Syntax :- Select column-name-list FROM
table 1 FULL OUTER JOIN table 2
ON table 1. column-name = table 2. column-name.

* SET OPERATIONS :-
The set operations are availed to syndicate information of similar type from one or more than one table. It seems to SQL joins although there is a big difference.

* Types of set operation
1. Union
2. Union All
3. Intersect
4. Minus.

① Union :- union operation is used to combine the result of two or more SQL SELECT queries. union operation eliminates the duplicate rows from its result set.

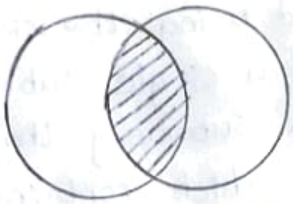


Syntax :- SELECT column-name FROM table 1
UNION
SELECT column_name FROM table 2;

② Union All :- Union All operation is equal to the union operation. It retur -ns the set without removing duplication and sorting the data.

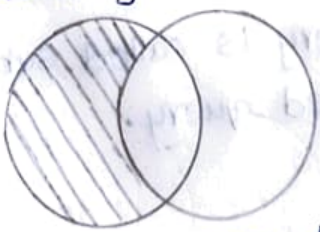
syntax :- select column_name FROM table 1
union All
select column_name FROM table 2;

③ Intersect :- Used to combine two select statements. The intersect operation return the common rows from both the select statements. In this number of number of datatypes and columns must be the same and no duplicates and it arranges the data in ascending order by default.



syntax :- Select column_name From table 1
INTERSECT
SELECT column_name From table 2;

④ Minus :- It combines the result of two select statements. Minus operator is used to display the rows which are present in the first query but absent in the second query. It has no duplicates the data arranged in ascending order by default.



Syntax :- SElect column_name From table 1
MINUS
select column_name From table 2;

***

* **VIEW :** A view is a virtual Representation of a table. It doesn't occupy any memory. Contain all rows of a table or select rows from a table. It can be created from one or many tables which depends on the written SQL query to create a view.

-ch

- A structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data in such a way that a user can see and modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

* **How to create a view ?**

A view can be created using the CREATE VIEW statement, we can create a view from a single table or multiple tables. Simple views which contains a subquery that retrieves from one base table and complex view which contains a subquery that retrieves from multiple base tables.

**Syntax :**

```
CREATE VIEW view_name AS
    SELECT column1, column2....
    FROM table_name
    WHERE condition;
```

* **SUB QUERY :**

→ A query inside the query is called subquery.
→ It is also called Nested query.

* **Some Important rules :**

- subquery can be placed in a number of SQL classes like WHERE clause, FROM clause, HAVING clause.
- use subquery with SELECT, UPDATE, INSERT, DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.
- Subquery is a query within another query.

- Outer query is known as the main query, and the inner query is known as a subquery.
- sub queries are used on the right side of the comparison operator.
- A subquery is enclosed in parentheses.
- In subquery, ORDER BY command cannot be used. But GROUP BY command can be used to perform the same function as ORDER BY command.

syntax:-

```
Select column_name
FROM table_name
WHERE column_name expression operator.
    (SELECT column_name from table_name WHERE ...);
```

* example:-

| ID | NAME | AGE | ADDRESS | SALARY |
|---|---|---|---|---|
| 1 | John | 20 | US | 2000 |
| 2 | David | 26 | Bangkok | 3000 |
| 3 | Alina | 27 | UK | 1500 |
| 4 | Kathrin | 29 | Bangalore | 6500 |
| 5 | Harry | 32 | china | 8500 |
| 6 | Jackson | 45 | US | 4500 |

* The subquery with a SELECT statement will be:

```
SELECT *
    FROM EMPLOYEE
    WHERE IN (SELECT ID FROM EMPLOYEE WHERE SALARY >4500);
```